arXiv:2603.02148v1 [cs.DS] 2 Mar 2026

一致性低秩近似

David P. Woodruff* Samson Zhou†

2026年3月3日

摘要

我们引入并研究一致性低秩近似问题：输入矩阵$\mathbf{A} \in \mathbb{R}^{n\times d}$的行依次到达，目标是提供一系列子空间，使其能够良好地近似每个时刻t已到达的子矩阵$\mathbf{A}^{(t)}$的最优秩-k近似，同时将重调度（即解序列的总体变化）降至最低。首先，我们证明，当目标是使低秩代价与最优代价之间的差不超过加性$\varepsilon\cdot||\mathbf{A}^{(t)}||_F^2$因子时，大约$\mathcal{O}\left(\frac{k}{\varepsilon}\log(nd)\right)$的重调度量是可行的。对于更具挑战性的目标，即实现最优秩-k代价的相对$(1+\varepsilon)$乘法近似，我们证明，在这种情形下，一个简单的上界是$\frac{k^2}{\varepsilon^2}\cdot\text{poly}\log(nd)$重调度量；对于整数有界矩阵，我们进一步将其改进为$\frac{k^{3/2}}{\varepsilon^2}\cdot\text{poly}\log(nd)$；对于在线条件数为多项式的数据流，则改进为$\frac{k}{\varepsilon^2}\cdot\text{poly}\log(nd)$。我们还证明，对于任何能维持最优低秩代价乘法$(1+\varepsilon)$近似的算法，即使预先已知完整输入，也至少需要$\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$的重调度量。最后，我们进行了多项实证评估，以补充理论保证，并展示了这些算法在实践中的有效性。

1 引言

低秩近似是机器学习、数据科学和统计学中经常使用的一项基础技术，用于从大型数据集中识别重要的结构信息。给定一个由n个观测值和d个特征组成的输入矩阵$\mathbf{A}\in\mathbb{R}^{n\times d}$，低秩近似的目标是将$\mathbf{A}$分解为由$k$个相互独立的潜变量（称为因子）构成的组合。我们用矩阵$\mathbf{U}\in\mathbb{R}^{n\times k}$和$\mathbf{V}\in\mathbb{R}^{k\times d}$表示这些因子，使得乘积$\mathbf{UV}$应当是原始数据集$\mathbf{A}$的一个“准确”表示。形式上，我们希望最小化量$\min_{\mathbf{U}\in\mathbb{R}^{n\times k},\mathbf{V}\in\mathbb{R}^{k\times d}}\mathcal{L}(\mathbf{UV}-\mathbf{A})$，其中$\mathcal{L}$表示根据特定性质预先选定的任意损失函数。尽管存在多种标准度量，例如子空间对齐角或峰值信噪比（PSNR），但本文聚焦于Frobenius损失。这或许是最常见的损失函数，因为它与最小二乘回归存在联系。因此，该问题等价于寻找合适的因子矩阵$\mathbf{V}\in\mathbb{R}^{k\times d}$，使其具有正交归一行，从而最小化量$||\mathbf{A}-\mathbf{AV}^\top\mathbf{V}||_F^2$；在这种情况下，最优的左因子矩阵为$\mathbf{U}=\mathbf{AV}^\top$。

秩参数k通常根据所选底层模型的复杂程度来确定，该模型用于拟合数据。识别并随后利用这些因子，往往可以减少观测中相关特征的数量，从而同时提高可解释性

*卡内基梅隆大学。 dpwoodru@gmail.com
†德克萨斯农工大学。 samsonzhou@gmail.com

并降低维度。具体而言，我们可以使用因子 U 和 V 对原始数据集 A 进行近似表示，因此表示中只需使用 $(n+d)k$ 个参数，而非 A 中原有的 nd 个条目。此后，给定一个向量 $\mathbf{x} \in \mathbb{R}^d$，我们可以在 $(n+d)k$时间内计算矩阵-向量乘积 UVx，作为 Ax 的近似；而计算 Ax 则需要nd 时间。凭借这些优势及其他优点，低秩近似成为最常用的工具之一，应用领域包括推荐系统、数学建模、预测分析、降维、计算机视觉和自然语言处理 [CGK+17、CW17、CMM17、MW17a、MW17b、BW18、BHM+21、VVWZ23、MS24、SVWZ24、KMS26]。

事实上，在许多应用中，基于低秩近似的方法始终优于其他数据摘要技术，包括聚类。一个著名的例子是 Netflix奖，这是2006年发起的一项公开竞赛，旨在设计出最准确的算法，仅利用历史评分数据预测用户电影评分，而不使用任何有关用户或电影的额外信息。获胜方案 [BKV07、BKV08、Kor09、BKV10] 主要依靠低秩矩阵近似来估计缺失评分，其准确度显著高于包括 Netflix 自身基于线性回归的系统在内的竞争方法。

数据流。 流式计算模型已成为分析大型动态数据集的一种常用框架，例如由商业交易产生的数据库日志、金融市场数据、物联网 (IoT) 设备数据、科学观测、社交网络通信或虚拟流量测量。在行到达模型中，每次流更新都会提供一个额外的观测值，即最终输入 $\mathbf{A}_t \in \mathbb{R}^d$ 的一个额外矩阵行 $\mathbf{A} \in \mathbb{R}^{n \times d}$，用于低秩近似。通常，在不确定未来输入的情况下，必须制定下游决策和策略。也就是说，输入可能以数据流的形式逐步揭示，而从业者可能需要在时刻 $t \in [n]$仅根据数据集 $\mathbf{A}^{(t)} = \mathbf{a}_1 \circ \ldots \circ \mathbf{a}_t$作出不可撤销的选择；该数据集由前 t 个数据点观测组成。一个典型例子是在线缓存/分页算法 [FKL+91、Ira96], ，这些算法必须在每个时间步决定在缓存中保留还是逐出项目。这一问题可推广为 k服务器问题，其中惩罚由度量空间 [MMS90、BBMN15、BCR23]刻画。

一致性。尽管理论上，在在线算法的背景下，不可撤销的决策很有意思，但在许多实际场景中，这种限制可能过于严格。 [LV17] 指出，在早期的缓存和分页置换场景中，接收在线请求、将任务分配给不同机器的负载均衡器，可以在必要时将先前的一些任务重新分配给其他机器，以提升整体性能。此外，算法能够根据更新后的信息调整先前决策，这使我们得以更深入地理解底层问题的结构，超越在线算法的不可能性障碍。另一方面，这类调整会对依赖算法所产生的决策／策略的应用造成后续影响。因此，人们定义了一致性和重调度这两个相关概念，用以量化算法输出解随时间变化的累计次数。针对许多问题，低回溯在线算法已得到广泛研究 [GK14、GKS14、LOP+15、GGK16、MSVW16、GL20、BBLS23], 而一致性聚类和设施选址近年来也受到广泛关注 [LV17、CHP+19、FLNS21、LHG+23]。在特征工程的典型场景中，低秩近似算法用于选择特定特征或特征的线性组合，并以此训练模型。因此，一致性代价高昂意味着需要对整个大型机器学习系统进行代价高昂的重新训练 [LV17], ，原因是传递给学习器的特征集合不同

传递给学习器的特征。

基于一致性的概念，我们对一致性低秩近似问题进行形式化，以刻画在线算法的需求：算法需要适应不断演变的数据，同时保持输出稳定，便于下游使用。如上所述，低秩近似广泛用于特征工程：生成的因子定义了下游模型的训练所在特征子空间。在动态设置中，即使只有细微更新，高变化量也可能导致这些特征突然改变，从而触发大规模模型频繁且代价高昂的重新训练 [LV17]。一致且重调度量低的 LRA 算法能够随时间保持特征稳定，降低重新训练成本并提高可靠性。

稳定性在许多实际领域都至关重要。在生物特征识别中，指纹或虹膜纹理的一致低维表示对于维持可靠的身份识别至关重要 [JRP04]。在图像处理中，目标检测、手写识别和人脸识别等任务依赖于派生特征；随着新数据到来，这些特征不应出现不可预测的波动 [NA12]。在数据压缩和信号处理中，稳定的降维表示能够在控制噪声的同时保留关键结构 [WF02，Pro07]。在文本挖掘和信息检索中，词频-逆文档频率向量或嵌入等数值特征必须保持连贯，才能维持搜索和分类的质量 [AA15，SMR08]。即使在面向基础模型的大规模数据整理中也是如此——聚类作为一种受约束的低秩近似形式，被用于去重数据——高变化量会导致代表集不稳定，并使模型反复重新训练 [LHG+23]。

在这些场景中，背后的原则是一致的：下游系统不仅依赖近似的质量，也依赖特征表示随时间推移的稳定性 。通过明确考虑这一需求，一致性低秩近似能够提供随变化平稳演进、同时保持强精度保证的解决方案，为动态的真实世界数据处理流程带来理论洞见和切实的实践收益。

1.1 我们的贡献

本文首次开展对一致性低秩近似的研究。

形式化模型。 给定精度参数 $\varepsilon > 0$，我们的目标是在任何时刻为低秩近似提供一个 $(1+\varepsilon)$-近似。我们假设输入是一个矩阵 $\mathbf{A} \in \mathbb{R}^{n\times d}$，其各行$\mathbf{a}_1, \ldots, \mathbf{a}_n$ 依次到达，因此在每个时刻 t，算法只能访问 $\mathbf{A}^{(t)}$，即 $\mathbf{A}$ 的前t 行。也就是说，算法的目标首先是在每个时刻 $t \in [n]$输出一个由 k 个因子构成的集合 $\mathbf{V}^{(t)} \in \mathbb{R}^{k\times d}$ ，使得

$$\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top \mathbf{V}^{(t)}\|_F^2 \le (1+\varepsilon)\cdot \mathsf{OPT}_t,$$

其中，OPT_t 表示时间 t 处最优低秩近似的代价，$\mathsf{OPT}_t = \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}\mathbf{V}^\top\mathbf{V}\|_F^2$。换言之，我们希望算法返回的因子 $\mathbf{V}^{(t)}$ 所产生的低秩代价能够准确地逼近最优低秩近似。其次，我们希望因子序列 $\mathbf{V}^{(1)}$、……、$\mathbf{V}^{(n)}$ 随时间的变化尽可能小。具体而言，算法的目标是最小化 $\sum_{t=2}^{n}$ 重调度$(\mathbf{V}^{(t)},\mathbf{V}^{(t-1)})$，其中，重调度$(\mathbf{R},\mathbf{T}) = \|\mathbf{P_R} - \mathbf{P_T}\|_F^2$表示两个子空间的正交投影矩阵 $\mathbf{P_R}$、$\mathbf{P_T}$ 之间的平方子空间距离。

我们对因子 $\mathbf{R}$，$\mathbf{T}$ 所采用的成本函数 Recourse$(\mathbf{R},\mathbf{T})$ 作如下说明。乍看之下，一种自然的成本函数定义或许是 $\mathbf{R}$ 与

$\mathbf{T}$ 之间不同向量的数量，因为从某种意义上说，这能反映 $\mathbf{R}$ 与 $\mathbf{T}$ 之间的变化。然而，值得注意的是，即便存在唯一的秩为 k 的子空间 $\mathbf{V}$ ，使低秩近似代价函数$\|\mathbf{A} - \mathbf{A}\mathbf{V}^\top\mathbf{V}\|_F^2$达到最小，$\mathbf{V}$ 仍可能有许多种表示方式，因为子空间内的基向量可以任意旋转。因此，对基向量的选择敏感的成本函数可能并不合适，因为基向量发生较大变化，也未必会导致最终投影 $\mathbf{A}\mathbf{V}^\top\mathbf{V}$发生任何变化。这意味着，合理的成本函数应当反映子空间 $\mathbf{R}$ 与 $\mathbf{T}$ 所张成空间之间的差异。例如，$\mathbf{R}$ 的正交子空间中属于 $\mathbf{T}$ 的部分的维数，就是一个合适的量。不过需要指出的是，该量仍会惩罚仅仅是 $\mathbf{R}$ 的微小扰动的子空间 $\mathbf{T}$ 。例如，若 $\mathbf{R}$ 是标准基向量 $(0,\ 1)$，而 $\mathbf{T}$ 是向量 $(\varepsilon,\ \sqrt{1-\varepsilon^2})$，其中 ε 任意小。更稳健的量是维数的连续类比，即 $\mathbf{T}$ 投影到 $\mathbf{R}$ 之外部分的平方质量；这个量恰好对应于我们的成本函数 Recourse。因此，我们认为，所选择的一致性成本函数相当自然。

我们注意到，还可以进一步假设输入矩阵 $\mathbf{A}$ 的整数项的绝对值均不超过某个参数 M。我们指出，这一假设在数值线性代数中很常见，因为通常很难在算法的输入中以任意精度表示实数。相反，对于有理数输入，经过适当缩放后，输入矩阵的每个条目都可以写成整数。因此，这一标准假设可以在不失一般性的前提下，对编码矩阵每个条目所用的比特数进行建模。

理论结果。 我们首先指出，最优低秩近似可能在每一步都发生彻底变化，也就是说，时刻 $t-1$ 的最优子空间 $\mathbf{V}^{(t-1)}$ 投影到最优子空间 $\mathbf{V}^{(t)}$后，其维数仍可能为 k 。因此，为达到最优性，可能需要 $\Omega(nk)$ 的重调度；也就是说，每当 n 行中的一行到达后，都要重新计算最佳的 k 因子。尽管如此，积极的一面是，我们首先证明，如果目标只是在任何时刻使低秩近似代价达到加性 $\varepsilon\cdot\|\mathbf{A}^{(t)}\|_F^2$ 误差，则可以实现次线性重调度。

定理 1.1。 假设$\mathbf{A}\ \in \mathbb{Z}^{n\times d}$ 是一个秩为 $r > k$ 、各条目绝对值不超过 M 的整数矩阵，并令$\mathbf{A}^{(t)}$ 表示$\mathbf{A}$的前 t 行，其中 $t \in [n]$。存在一种算法，能够始终以 $\varepsilon\cdot\ \|\mathbf{A}^{(t)}\|_F^2$实现对最优低秩近似$\mathbf{A}$代价的加性近似，并实现 $O\left(\frac{k}{\varepsilon}\log(ndM)\right)$的重调度。

我们指出，与 定理 1.1 对应的算法使用 $\frac{kd}{\varepsilon}\cdot\mathrm{polylog}(ndM)$ 比特的空间，且 $d\cdot\mathrm{poly}\left(k,\frac{1}{\varepsilon},\log(ndM)\right)$ 的摊销更新时间。由于平方弗罗贝尼乌斯范数是最优低秩近似代价的上界，实现相对于最优代价的加性 $\varepsilon\cdot\ \|\mathbf{A}^{(t)}\|_F^2$误差，远比实现相对 $(1+\varepsilon)$ 倍乘性误差容易，尤其是当顶端奇异向量对应较大的奇异值时。事实上，如果数据流的在线条件数至多为$\mathrm{poly}(n)$，我们甚至可以实现与 k 成线性的重调度：

定理 1.2。 对于在线条件数为 $\mathrm{poly}(n)$ 的数据流，存在一种算法，可实现低秩近似的 $(1+\varepsilon)$ 近似，且重调度次数为 $O\left(\frac{k}{\varepsilon^2}\log^3 n\right)$。

我们指出，对于在线条件数为$\mathrm{poly}(n)$的数据流，我们实际上在引理 2.3中证明了一个更强的结果。具体而言，我们证明，在对应于单个元素变化、行修改、k 行插入和行删除的秩一扰动之后，最优秩-子空间的变化幅度仅为一个常数。

因此，利用将“有效”流长度缩短的标准技术，该结果可推出定理 1.2；事实上，它还立即给出一种算法，能够在任意一系列此类更新下维护最优秩-k 近似，并且在长度为 n 次操作的数据流上，总重调度次数仅为 $O(n)$。因此，我们的方法能够处理插入-删除流中出现的显式分布偏移；进一步使其能够处理滑动窗口模型等情形下的隐式删除 [DGIM02, CMS13, BWZ21, BGL+18, BDM+20, WZ21, BEL+20, EMMZ22, JWZ22, BLMZ23, WY23, WZZ23, CJY+25, BGW+26, NPW+26]，是未来研究的一个自然方向。

然而，对于条目由 $\mathrm{poly}(n)$ 界定的标准矩阵，在线条件数有界于 $\mathrm{poly}(n)$ 的假设可能并不成立。例如，已知存在维数为 $n \times d$、但最优低秩代价小至 $\exp(-\Omega(k))$ 的整数矩阵。为此，我们首先注意到，简单应用 [BDM+20] 的一个结果，即可在始终保持 $(1+\varepsilon)$ 乘法近似的同时，实现 $\frac{k^2}{\varepsilon^2} \cdot \mathrm{polylog}(ndM)$ 的重调度。事实上，对于常数 ε，通过一种称为在线岭杠杆分数采样的过程，大约可以采样 $k \cdot \mathrm{polylog}(ndM)$ 行，从而始终保持低秩近似。接下来，对于二次重调度，只需在每次采样新行时，重新计算采样子矩阵的前几个右 k 奇异向量。一个自然的问题是，每一步是否都必须进行 $\Omega(k)$ 的重调度，也就是说，重新计算前几个右 k 奇异向量是否不可避免。我们证明事实并非如此，并且总体重调度量可以做到次二次。

定理 1.3. 设$\mathbf{A} \in \mathbb{Z}^{n \times d}$ 是一个整数矩阵，其条目的绝对值不超过 M。存在一种算法，始终能以$(1+\varepsilon)$近似最优低秩近似$\mathbf{A}$ 的代价，并实现 $\frac{k^{3/2}}{\varepsilon^2} \cdot \mathrm{polylog}(ndM)$的重调度量。

我们再次指出，对应于定理 1.3 的算法使用 $\frac{kd}{\varepsilon} \cdot \mathrm{polylog}(ndM)$ 比特的空间，且具有 $d \cdot \mathrm{poly}\left(k, \frac{1}{\varepsilon}, \log(ndM)\right)$ 的摊销更新时间。最后，我们证明，即使完整输入事先已知，任何用于低秩近似的乘法$(1+\varepsilon)$近似算法也必然需要 $\Omega\left(\frac{k}{\varepsilon} \log \frac{n}{k}\right)$ 的重调度量。

定理 1.4. 对于任意参数 $\varepsilon > \log_n^{-n}$，存在一个行序列$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$，使得任何始终以$(1+\varepsilon)$近似最优低秩近似代价的算法，都必须具有$\Omega\left(\frac{k}{\varepsilon} \log \frac{n}{k}\right)$的一致性代价。

实证评估。 我们在 第 5 节 中通过多项实证评估补充了理论结果。我们的结果表明，尽管形式化保证对算法输出的低秩解的近似代价进行了最坏情况分析，但实际表现甚至可能（好得）多。重要的是，我们的结果表明，诸如 Frequent Directions [GLPW16] 等在线低秩近似算法无法实现良好的重调度，这促使我们研究专为一致性低秩近似设计的算法。

论文结构。我们在第 2 节介绍线性回溯算法，并在第 5 节进行实证评估。我们在第 3 节给出关于整数值矩阵的结果。所有证明均放在完整附录中，具体而言，下界的证明见第 4 节。读者还可参阅第 1.4 节，了解本文使用的标准符号和其他预备知识。

1.2 相关工作

1.2.1 相关工作：一致性

我们注意到，近期有大量工作研究各种问题的一致性。[LV17], 率先提出了一致性聚类问题，并为仅插入流（即增量设置）中的 k-聚类给出了一个重调度量为 $k^2 \cdot \text{polylog}(n)$ 的算法。随后，[FLNS21], 将这一重调度上界改进至 $k \cdot \text{polylog}(n)$，而 [GKLX21]则提出了一个能够抵御离群点的版本。近期，[CJWZ25]还将近似保证改进至 $(1+\varepsilon)$。近期还有一系列工作研究动态设置下的 k-聚类 [LHG+23, BCG+24, FS25], ，在这种设置中，点可以被插入或删除。[CLMP22]研究的并非 k-聚类，而是相关聚类：边被标记为正或负，目标是形成任意数量的簇，以最小化簇内负标记边的数量以及不同簇之间正标记边的数量。对于聚类以外的问题，也有一系列工作聚焦于次模最大化 [JKLV19, DFL+24a, DFL+24b]。

一致性聚类。另一种方法或许是借鉴一致性聚类领域的思路。在这种设定中，$\mathbb{R}^d$ 中的点会逐个到达，目标是在最小化总重配置次数的同时，对 (k,z)-聚类代价实现常数因子近似。此处，时刻 t 产生的重调度次数，是时刻 $t-1$ 与时刻 t所选聚类中心之间对称差的大小。唯一能在k 上实现次二次重调度次数的算法是 [FLNS21], 提出的算法。该算法通过考察每个现有点周围半径逐渐增大的几何球，来选取对未来可能出现的点不那么敏感的中心，从而在每个时刻构建稳健的簇。不幸的是，这种技术利用了 k-聚类目标中隐含的几何性质，而对于低秩近似，相应的类比并不显而易见。

1.2.2 相关工作：稻草人方法

本节将简要介绍一些密切相关模型中的现有技术，并直观说明它们为何不足以适用于我们的场景。

Frequent Directions 和在线岭杠杆分数抽样。最自然的方法是应用流式处理文献中现有的低秩近似算法。最相关的两项工作分别是 [GLPW16] 提出的确定性 Frequent Directions 方法，以及由 [CMM17, BDM+20]推广的（在线）岭杠杆分数抽样程序。这两种方法始终维护少量能够捕捉矩阵“重要”方向的行。因此，为了在每个时刻报告近似最优的秩-k 近似，这些算法只需返回各自所存储矩阵的奇异值分解中的前 k 个右奇异向量。然而，很容易设想这样一种情形：第 k大和第（$k+1$）大奇异向量反复交替，从而每一步都会产生重调度。例如，假设 $k=1$并且对所有整数 $t>0$始终有 $2t$，前两个奇异向量分别为 $(2t, 0)$ 和 $(0, 2t-\varepsilon)$；但对所有整数 $t \geq 0$始终有 $2t+1$，前两个奇异向量分别为 $(2t+1-\varepsilon, 0)$ 和 $(0, 2t+1)$。那么对所有 $2t$， $k=1$ 的最佳秩-k 解将是标准基向量 e_1 ；而对所有 $2t+1$，最佳秩-k 解将是标准基向量 e_2。这些算法

将产生重调度 n，而即使是一个从不改变初始向量 $\mathbf{e}_2$ 的算法，也只会产生重调度 0。因此，这些算法似乎遭遇了灾难性失败；换言之，即使对于简单输入，它们甚至无法提供关于重调度的 poly(n)-乘法近似。

可以看出，我们的目标仅仅是给出总重配置次数的上界，而不是对重调度实现乘法近似。事实上，为此目的，在线岭杠杆抽样技术能带来一定收益。具体而言，[BDM+20] 证明了在整个数据流上，至多有 $\frac{k}{\varepsilon^2}\cdot\operatorname{polylog}\left(n,d,\frac{1}{\varepsilon}\right)$ 行会总共被抽样进入草图矩阵，而且草图矩阵能够准确刻画向任意维数为 k的子空间投影的残差。因此，为实现$\mathcal{O}\left(k^2\right)$ 次重调度，只需简单地重新计算前k 个奇异向量，即每当在线岭杠杆抽样过程抽到一个新行时。

奇异值分解。注意，前述例子还表明，更一般地，仅仅输出奇异值分解（SVD）的前k 个右奇异向量是不够的。不过，人们或许希望，每当在线岭杠杆分数抽样将一行抽入草图矩阵时，只需替换 SVD 中的一个方向即可。遗憾的是，前一步的近似最优解似乎可能因单行的到来而需要替换全部 k 个因子。例如，假设因子$\mathbf{V}^{(1)}$由标准基向量$\mathbf{e}_{k+1}$，……，$\mathbf{e}_{2k}$ 构成，与由标准基向量$\mathbf{e}_1$，……，$\mathbf{e}_k$ 构成的因子$\mathbf{V}^{(2)}$ 达到相同的损失。若下一行恰好在坐标1，……，k上非零，则前 k 空间可能从$\mathbf{V}^{(1)}$ 完全变为$\mathbf{V}^{(2)}$。虽然这种最坏情况输入不可避免，我们证明它只会发生很少次。更成问题的是，这些因子中只有一个发生剧烈变化，而另外 $k-1$ 个因子只发生微小变化，但我们仍然输出 k 个全新的因子。我们的算法通过对相应奇异值进行分类讨论，谨慎选择要替换的因子，从而避免了这一问题。

1.3 技术概述

本节将介绍我们主要结果的直观解释，概述我们的算法、各种挑战，以及自然想到的其他方法及其行不通的原因。

热身：加性误差算法。作为简单的热身，我们首先介绍一种算法，该算法在所有时刻 $t\in[n]$ 的加性误差至多为 $\varepsilon\cdot\|\mathrm{A}^{(t)}\|_F^2$，同时仅产生 O $\left(\frac{k}{\varepsilon}\log(ndM)\right)$ 的重调度，对应于定理 1.1。这个算法相当简单。我们在所有时刻 $t\in[n]$ 维护矩阵 $\mathrm{A}^{(t)}$ 的平方弗罗贝尼乌斯范数。只要平方弗罗贝尼乌斯范数自上次更换因子集合以来的增长未达到 $(1+\varepsilon)$ 倍，我们就继续维护同一组因子。当时刻 t 的平方弗罗贝尼乌斯范数自上次更换因子集合以来增长了 $(1+\varepsilon)$ 倍时，我们只需对 $\mathrm{A}^{(t)}$ 进行奇异值分解，并将 $\mathrm{V}^{(t)}$ 设为 $\mathrm{A}^{(t)}$ 的前 k 个右奇异向量。由于我们更换了整组因子，这一过程产生的重配置成本至多为 k。

正确性源于以下观察：每次我们重置$\mathbf{V}^{(t)}$时，都会在时间 t找到最优解。现在，在 t之后的接下来几次中，我们的解最多只会因由传入行构成的子矩阵的平方弗罗贝尼乌斯范数而退化；由于要求每当平方弗罗贝尼乌斯范数增加一个（$1+\varepsilon$）因子时，我们都重新计算$\mathbf{V}^{(t)}$，该范数小于整个矩阵平方弗罗贝尼乌斯范数的 ε比例。因此，弗罗贝尼乌斯范数

最多总共增加一个乘法（$1+\varepsilon$）因子 $\mathcal{O}\left(\frac{1}{\varepsilon}\log(ndM)\right)$ 次，并且每次我们都会产生重配置成本 k，因此最终的重调度至多为所需的 $\mathcal{O}\left(\frac{k}{\varepsilon}\log(ndM)\right)$。

用于相对误差算法的数据流缩减。现在，我们讨论如何在始终保持对最优低秩近似代价的相对（$1+\varepsilon$）乘法近似的同时，实现 $\frac{k^{3/2}}{\varepsilon^2}\cdot\text{polylog}(ndM)$ 的重调度，即定理 1.3。首先，我们利用在线岭杠杆分数采样 [BDM+20]，在数据流 S 中对行 a1、. . .、a_n 进行在线采样，选出 $\frac{k}{\varepsilon}\cdot\text{polylog}(ndM)$ 行，构成数据流 S′。该数据流由 A 中重新加权的行 b1、. . .、b_m 组成，且 $m=\frac{k}{\varepsilon}\cdot\text{polylog}(ndM)$。根据在线岭杠杆分数采样的保证，要使由行 a1、. . .、a_t 构成的矩阵 $\text{A}^{(t)}$ 达到（$1+\varepsilon$）-近似，只需使由截至时间 t 已到达的行 b1、. . .、$\text{b}_{t'}$ 构成的矩阵 $\text{B}^{(t')}$ 达到（$1+\text{O}(\varepsilon)$）-近似。注意，由于 B 是 A 的子矩阵，因此有 $t'\le t$。此外，A 中被采样到 B 的行，其权重最多只增加 $\text{poly}(n)$ 倍，因此我们可以假设条目的大小仍由 n 多项式级地界定。因此，只需改为对矩阵 B 执行一致性低秩近似即可。于是，在接下来的讨论中，我们假设流长度为 $\frac{k}{\varepsilon}\cdot\text{polylog}(ndM)$，而不是 n。

数据流上的相对误差算法。 现在我们描述一种算法，该算法在长度为 $\frac{k}{\varepsilon}\cdot\text{polylog}(ndM)$ 的数据流的所有时刻，均能实现 $(1+\varepsilon)$-乘法相对误差。在数据流中的某个时刻 s，我们使用 $\mathbf{A}^{(s)}$ 的奇异值分解来计算其最大的 k 个右奇异值，并将它们设为我们的因子 $\mathbf{V}^{(s)}$。此后，每当最优低秩近似代价自上次将因子设为 $\mathbf{V}^{(s)}$ 以来增加了一个乘法因子 $(1+\mathcal{O}(\varepsilon))$，我们就可以执行此操作。我们首先讨论如何在最优低秩近似代价每次增加 $(1+\varepsilon)$-乘法因子之间，以所需的重调度次数维持一个 $(1+\varepsilon)$-近似。

为此，我们对 $\mathbf{V}^{(s)}$ 的前 k 个右奇异值中底部 $\sqrt{k}$ 个奇异值的贡献进行分类讨论。也就是说，如果 $\sum_{i=k-\sqrt{k}}^{k}\sigma_i^2(\mathbf{A}^{(s)})$“较小”，那么直观来说，可以替换相应的奇异向量而不会显著增加误差。因此，在这种情况下，每当有一行新到达时，我们都可以用新行替换一个底部 $\sqrt{k}$ 奇异向量。接下来的 $\sqrt{k}$ 次更新中，这一过程将产生 $\sqrt{k}$ 总重调度次数；之后，在时刻 t，我们将因子重置为矩阵 $\mathbf{A}^{(t)}$ 的前 k 个右奇异向量。因此，在 $\sqrt{k}$ 个时间步内，我们产生的重调度次数为 $\mathcal{O}(k)$。

另一方面，如果 $\sum_{i=k-\sqrt{k}}^{k}\sigma_i^2(\mathbf{A}^{(s)})$“很大”，那么最优低秩近似因子就不能过多地从这些方向上被投影掉，否则最优低秩近似代价会显著增加。因此，我们可以在每个时刻简单地选择最优的前右 k 奇异向量集合，因为新子空间与旧子空间之间会有大量重叠。特别地，如果我们选择的阈值是 $\sum_{i=k-\sqrt{k}}^{k}\sigma_i^2(\mathbf{A}^{(s)})$ 作为最优代价的一个 $\mathcal{O}(\varepsilon)$ 因子，那么我们证明，产生 $\sqrt{k}$ 重调度会使低秩近似代价增加 $(1+\varepsilon)$，这违反了如下假设：我们所考虑的时刻中，最优低秩近似代价尚未按 $(1+\varepsilon)$ 乘法因子增长。因此，每个时刻的重调度至多为 $\sqrt{k}$。

总而言之，在最优低秩近似代价增加一个（$1+\varepsilon$）乘法因子的各个时刻之间，每个时刻我们至多需要 $\sqrt{k}$ 次重调度。由于数据流长度为 $\frac{k}{\varepsilon}\cdot\text{polylog}(ndM)$，因此这些时刻的总重调度次数为 $\frac{k^{1.5}}{\varepsilon}\cdot\text{polylog}(ndM)$，

这正是我们所需的界。接下来需要界定在最优低秩近似代价增加一个（$1+\varepsilon$）乘法因子的时刻 s 的总重调度次数，因为此时我们会将解重置为前 k 个右奇异值，并在每个这样的时刻产生 k 次重调度。由于弗罗贝尼乌斯范数至多为 poly(ndM)，自然会得出这样的结论：最优低秩近似代价至多增加 $\mathcal{O}\left(\frac{1}{\varepsilon}\log(ndM)\right)$ 次（每次增加一个（$1+\varepsilon$）乘法因子）。遗憾的是，事实并非如此。

反哈达玛矩阵。问题在于，存在反哈达玛整数矩阵的构造，其维数为 $n\times d$，但最优低秩代价可小至 $\exp(-\mathcal{O}(k))$。因此，最优代价可以加倍 $\mathcal{O}(k)$ 次，从而产生总重配置次数 $\mathcal{O}\left(k^{2.5}\right)$，这大得不可取。相反，我们证明，当最优低秩代价指数级地很小时，矩阵的秩也必然相当小，这意味着我们算法的重调度不可能像满秩情形那样大。具体而言，我们推广由 [CW09] 给出的一个结果，以证明若整数矩阵 $\mathbf{A}\in\mathbb{Z}^{n\times d}$ 的秩为 $r>k$，且条目的绝对值以 M为界，则其最优低秩近似代价至少为 $(ndM^2)^{-\frac{k}{r-k}}$。因此，我们只需在秩小于 $2k$。

幸运的是，当秩至多为 $r<2k$ 时，我们可以对上述情况进行更细致的分析，因为只有 r 个向量张成行空间，因此许多先前操作所需的重调度最多为 $r-k$。特别是，对于 $r<k$，我们只需维护整个行空间。接下来我们证明，如果矩阵的秩 r 介于 $k+2^i$ 和 $k+2^{i+1}$ 之间，那么在最优低秩近似的代价至少为 $(ndM^2)^{-100}$ 之前，最多会经过 $\mathcal{O}\left(\frac{1}{\varepsilon}\frac{k}{2^i}\right)$ 个时期。此外，重新计算顶部特征空间所产生的重调度最多为 $r-k\le 2^{i+1}$，因此矩阵的秩介于 $k+2^i$ 和 $k+2^{i+1}$ 之间时的总重配置次数最多为 $\mathcal{O}\left(\frac{k}{\varepsilon}\log(ndM)\right)$。由此可得，在秩达到 $2k$ 之前所有时刻的总重调度最多为 $\mathcal{O}\left(\frac{k}{\varepsilon}\log^2(ndM)\right)$。

敏锐的读者可能会问，对于我们的算法方法，$k^{3/2}$ 是否是最优的重调度次数。为此，请注意，如果每次更新时我们更改大量因子，那么总重配置次数就会增加。假设大约有 k 次更新，这就是在线投影代价保持核心集的规模。现在，如果每次更新时我们更改 r 个因子，那么总重配置次数至少为 kr。另一方面，如果每次更新时更改的因子较少，那么我们就需要每隔 $\frac{k}{r}$ 步重新计算一次。每次重新计算需要 k 次重调度，因此总重调度次数为 $\frac{k^2}{r}$。因此，kr 和 $\frac{k^2}{r}$ 中的较大值在 $r=\sqrt{k}$ 处达到最小，对应的重调度次数为$kr=\frac{k^2}{r}=k^{3/2}$。

返修下界。用于证明必须进行 $\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$ 重调度、对应定理 1.4 的下界构造非常简单。我们将数据流划分为 $\Theta\left(\frac{1}{\varepsilon}\log\frac{n}{k}\right)$ 个阶段，使得每个阶段之间最优低秩近似代价增加一个乘法 $(1+O(\varepsilon))$ 因子。此外，第 i 个阶段的最优解与第 $(i-1)$ 个阶段的最优解正交，因此根据阶段 i的奇偶性，最优解要么是前 k 个标准基向量，要么是从 $k+1$ 到 $2k$的标准基向量。因此，在任何时刻都保持乘法 $(1+\varepsilon)$-近似，就要求在每个阶段之间承担 $\Omega(k)$ 次重调度，这就得到了所需的 $\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$ 下界。

1.4 预备知识

我们用 $[n]$ 表示集合 $\{1, \ldots, n\}$。我们用 poly(n) 表示关于 n 的固定多项式，该多项式可以通过参数设置中的常数进行调整。我们用 polylog(n) 表示 poly($\log n$)。我们用 $\circ$ 表示行 $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^d$ 的垂直串接，使得 $\mathbf{a}_1 \circ \mathbf{a}_2 = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}$。如果事件 $\mathcal{E}$ 以高概率发生，我们称其发生，其中 $\mathbf{Pr}[\mathcal{E}] \geq 1 - \frac{1}{\text{poly}(n)}$。回顾一下，矩阵 $\mathbf{A} \in \mathbb{R}^{n \times d}$ 的弗罗贝尼乌斯范数定义为 $\|\mathbf{A}\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{d} A_{i,j}^2 \right)^{1/2}$。

矩阵$\mathbf{A} \in \mathbb{R}^{n \times d}$ 秩为 r 时的奇异值分解为$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$，其中$\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$，$\mathbf{V} \in \mathbb{R}^{r \times d}$；$\mathbf{U}$ 的列是正交归一的，$\mathbf{V}$ 的行是正交归一的，而 $\mathbf{\Sigma}$ 是一个对角矩阵，其条目对应于$\mathbf{A}$的奇异值。方阵$\mathbf{Q} \in \mathbb{R}^{n \times n}$的迹记作 Trace(**Q**)，定义为其对角条目之和，即 Trace(**Q**) $= \sum_{i=1}^{n} Q_{i,i}$。等价地，Trace(**Q**) 等于$\mathbf{Q}$的特征值之和（按重数计）。

接下来，我们回顾一个关于我们所采用的低秩近似表述的最优解的重要性质，即使用 Frobenius损失时的情形。

定理 *1.5*（*Eckart-Young-Mirsky* 定理）。 *[EY36, Mir60]* 设 $A \in \mathbb{R}^{n \times d}$ 的秩为 r，其奇异值分解为 $A = U\Sigma V$，其中 $U \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$，$V \in \mathbb{R}^{r \times d}$，奇异值为$\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_d(A)$。令 X 为 A 的前 k 个右奇异向量，即 V 的前 k 行，若存在并列则任意打破。则 A 的一个最优秩 k 近似为 $AX^\top X$，代价为 $\|A - AX^\top X\|_F^2 = \sum_{i=k+1}^{d} \sigma_i^2(A)$。

作为定理 1.5的推论，计算最优秩k 近似的算法过程只需取 k 个$\mathbf{A}$的右奇异向量，参见 RECLUSTER($\mathbf{A}$ k)过程，它位于算法 1中，不过也可以使用更快的近似奇异值分解方法。

推论 1.6。 存在一个算法 *Recluster*($\mathbf{A}$ k)，它输出一个由正交归一行组成的集合 $\mathbf{X}$，从而得到 $\mathbf{A}$ 的最优秩 k 近似。

算法 1 Recluster($\mathbf{A}$ k)，即截断奇异值分解

输入： 矩阵 $\mathbf{A} \in \mathbb{R}^{n \times d}$，秩参数 k
输出： 前 k 个$\mathbf{A}$的右奇异向量
1: 令 r 为$\mathbf{A}$的秩2: 令 $\mathbf{U} \in \mathbb{R}^{n \times r}, \mathbf{\Sigma} \in \mathbb{R}^{r \times r}, \mathbf{V} \in \mathbb{R}^{r \times d}$ 为$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$的奇异值分解3: 返回 $\mathbf{V}$的前 min(r, k) 行

现在，我们来说明采用重调度定义的动机。

引理 1.7. 对于秩为 k 的子空间 $\mathbf{R}$ 和$\mathbf{T}$，设它们对应的正交投影矩阵分别为 $\mathbf{P}$ 和 $\mathbf{Q}$。那么存在常数 C_1, $C_2 > 0$，使得 $C_1(\|\mathbf{P} - \mathbf{PQ}\|_F^2 + \|\mathbf{Q} - \mathbf{QP}\|_F^2) \leq$重调度$(\mathbf{R}, \mathbf{T}) \leq C_2(\|\mathbf{P} - \mathbf{PQ}\|_F^2 + \|\mathbf{Q} - \mathbf{QP}\|_F^2)$.

证明。根据定义，我们有重调度(R, T) $= \|$P $-$ Q$\|_F^2$。根据三角不等式，我们有$\|$P $-$ Q$\|_F \leq \|$P $-$ PQ$\|_F + \|$PQ $-$ Q$\|_F$。注意到 $\|$Q $-$ QP$\|_F^2 = \|$Q $-$ PQ$\|_F^2$，因为左端是(Q $-$ QP)$^\top$(Q $-$ QP)的迹，它等于(Q $-$ PQ)(Q $-$ QP)的迹，

这是因为对于投影矩阵Q和P，有$P^\top$ =P以及$Q^\top = Q$。根据迹的循环性质，(Q − PQ)(Q − QP)的迹因此等于(Q − QP)(Q − PQ)的迹，而由同样的论证，后者是右端的迹。因此可得 ‖Q − QP $\|_F^2 = \|Q - PQ\|_F^2$，即为所求。于是，我们有 $\|P - Q\|_F \le \|P - PQ\|_F + \|QP - Q\|_F$。

接下来，注意到

$$\begin{aligned}\|\mathbf{P}-\mathbf{PQ}\|_F^2 + \|\mathbf{PQ}-\mathbf{Q}\|_F^2 &= \operatorname{Trace}(\mathbf{P}) + \operatorname{Trace}(\mathbf{Q}) - 2\operatorname{Trace}(\mathbf{PQ}) \\ &= 2k - 2\operatorname{Trace}(\mathbf{PQ}),\end{aligned}$$

因为 $\mathbf{P}^2$ =P, $\mathbf{Q}^2 = \mathbf{Q}$ 是对称幂等矩阵，且 Trace(P) =Trace(Q) $= k$。类似地 ,

$$\begin{aligned}2k + 2\|\mathbf{QP}\|_F^2 - 4\operatorname{Trace}(\mathbf{PQ}) &= 2k + 2\operatorname{Trace}(\mathbf{PQP}) - 4\operatorname{Trace}(\mathbf{PQ}) \\ &= 2k - 2\operatorname{Trace}(\mathbf{PQ}),\end{aligned}$$

利用 $\|\mathbf{QP}\|_F^2 = \operatorname{Trace}(\mathbf{PQP}) = \operatorname{Trace}(\mathbf{PQ})$。因此，我们有 $\|\mathbf{P}-\mathbf{PQ}\|_F^2 + \|\mathbf{PQ}-\mathbf{Q}\|_F^2 = 2k + 2\|\mathbf{QP}\|_F^2 - 4\cdot\operatorname{Trace}(\mathbf{P}^\top\mathbf{Q})$。后一个量至多为 $4k - 4\cdot\operatorname{Trace}(\mathbf{P}^\top\mathbf{Q}) = 2(2k - 2\cdot\operatorname{Trace}(\mathbf{P}^\top\mathbf{Q}))$，这正是 $2\|\mathbf{P}-\mathbf{Q}\|_F^2$。

□

因此，只需采用较不自然但或许在数学上更易处理的定义 $\|\mathbf{R} - \mathbf{RT}^\dagger\mathbf{T}\|_F^2 + \|\mathbf{T} - \mathbf{TR}^\dagger\mathbf{R}\|_F^2$，即两个子空间质量的对称差，将其作为重调度的概念。

最后，我们回顾数值线性代数中的以下两个标准结论。

定理 *1.8*（极小极大定理）。设 $A \in \mathbb{R}^{n\times d}$ 是一个奇异值为 $\sigma_1(A) \ge \sigma_2(A) \ge \ldots \ge \sigma_d(A)$ 的矩阵，并且对所有 $j \in [d]$, $\xi_j(A) = \sigma_{d-j+1}(A)$，因此 $\xi_1(A) \le \ldots \le \xi_d(A)$ 是 A 的逆序谱。那么，对于 A 的任意一个维数为 k 的子空间 V，都存在单位向量 x、$y \in V$，使得

$$\|\mathbf{Vx}\|_2^2 \le \sigma_k^2(\mathbf{A}), \qquad \|\mathbf{Vy}\|_2^2 \ge \xi_k^2(\mathbf{A}).$$

定理 *1.9*（柯西交错定理）。 *[Hwa04, Fis05]* 设 $A \in \mathbb{R}^{n\times d}$ 是一个奇异值为 $\sigma_1(A) \ge \sigma_2(A) \ge \ldots \ge \sigma_d(A)$ 的矩阵。设 $v \in \mathbb{R}^d$，且 $B = A \circ v \in \mathbb{R}^{(n+1)\times d}$，其奇异值为 $\sigma_1(B) \ge \sigma_2(B) \ge \ldots \ge \sigma_d(B)$。那么，对所有 $i \in [d]$, $\sigma_i(B) \ge \sigma_i(A)$。

2 具有最优重规划的简单算法

本节简要介绍两种简单算法，其重调度次数为线性于 k。

2.1 加性误差

第一个算法在目标是在任何时刻都维持加性 $\varepsilon\cdot\|\mathbf{A}^{(t)}\|_F^2$误差 $t \in [n]$时，所需的重调度次数大约为 $\mathcal{O}\left(\frac{k}{\varepsilon}\log n\right)$，其中$\mathbf{A}^{(t)}$ 是截至时间 t已到达的矩阵$\mathbf{A}$ 的 t 行。我们只需持续跟踪矩阵$\mathbf{A}^{(t)}$ 的平方弗罗贝尼乌斯范数。每当平方弗罗贝尼乌斯范数增大了$(1+\varepsilon)$倍时，我们就重新计算$\mathbf{A}^{(t)}$ 的奇异值分解，并选择$\mathbf{V}^{(t)}$ 为$\mathbf{A}^{(t)}$的前 k 个右奇异向量。其他时刻则保持相同的因子集合。

核心直觉是，每次重置$\mathbf{V}^{(t)}$时，我们都会找到时刻 t 的最优解。在 t之后的接下来几步中，我们的解会逐渐变差，但它最多只会变差平方

由新到达的行构成的子矩阵的弗罗贝尼乌斯范数。因此，只要该量小于整个矩阵的平方弗罗贝尼乌斯范数的 ε比例分数，我们的正确性保证就成立。另一方面，只要平方弗罗贝尼乌斯范数的增幅未达到倍，这样的保证就$(1+\varepsilon)$必须成立。因此，对于矩阵的平方弗罗贝尼乌斯范数每次以乘法方式增长 k，我们都会产生重调度；这种情况最多发生$\mathcal{O}\left(\frac{1}{\varepsilon}\log(ndM)\right)$次。我们在$(1+\varepsilon)$中完整给出我们的算法：算法 2。

算法 2 低回溯低秩近似的加性误差算法 输入：输入矩阵 A 的行 a1, . . . , a$_n \in \mathbb{R}^{n\times d}$，其整数项的绝对值不超过$M$，误差参数 $\varepsilon > 0$输出：在任意时刻，相对于最优低秩近似代价的加性 $\varepsilon\cdot \|\mathbf{A}^{(t)}\|_F^2$ 误差 1: $C \leftarrow 0$2: 对每一行 a$_t$ 执行 3: 如果 $\|\mathbf{A}^{(t)}\|_F^2 \geq (1+\varepsilon)\cdot C$ 则 4: V $\leftarrow$ Recluster(A$^{(t)}$, k) 5: $C \leftarrow \|\mathbf{A}^{(t)}\|_F^2$6: 结束 if 7: 返回 V 8: 结束 for

---

我们证明算法 2始终具有正确性：

引理 2.1。 设 $\mathbf{A}^{(t)}$ 是 t *A* **的前** 行，且 $\mathbf{V}^{(t)}$ 是算法 *2* 在时刻 t的输出。设OPT_t是在时刻 t的最优低秩近似代价。那么 $\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 \leq \mathsf{OPT}_t + \varepsilon\cdot\|\mathbf{A}^{(t)}\|_F^2$。

证明。 令 s 为首次设置 $\mathbf{V}^{(t-1)}$ 的时刻，因此 $\mathbf{V}^{(t-1)}=\mathbf{V}^{(s)}$ 是 $\mathbf{A}^{(s)}$的前 k 个右奇异向量。因此， $\|\mathbf{A}^{(s)}-\mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2 = \mathsf{OPT}_s$。于是，

$$\begin{aligned}\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 &= \|\mathbf{A}^{(s)}-\mathbf{A}^{(s)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 + \sum_{i=s+1}^{t}\|\mathbf{a}_i-\mathbf{a}_i(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2\\ &= \|\mathbf{A}^{(s)}-\mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2 + \sum_{i=s+1}^{t}\|\mathbf{a}_i-\mathbf{a}_i(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2\\ &= \mathsf{OPT}_s + \sum_{i=s+1}^{t}\|\mathbf{a}_i-\mathbf{a}_i(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_2^2.\end{aligned}$$

注意，由于（$\mathbf{V}^{(t)}$）$^\top\mathbf{V}^{(t)}$ 是一个投影算子，因此$\mathbf{a}_i$ 投影到$\mathbf{V}^{(t)}$的行空间后，其长度不会增加，所以 $\|\mathbf{a}_i-\mathbf{a}_i$（$\mathbf{V}^{(t)}$）$^\top\mathbf{V}^{(t)}\|_2^2 \leq \|\mathbf{a}_i\|_2^2$。因此，

$$\begin{aligned}\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 &= \mathsf{OPT}_s + \sum_{i=s+1}^{t}\|\mathbf{a}_i-\mathbf{a}_i(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_2^2\\ &\leq \mathsf{OPT}_s + \sum_{i=s+1}^{t}\|\mathbf{a}_i\|_2^2\\ &\leq \mathsf{OPT}_s + \varepsilon\cdot\|\mathbf{A}^{(s)}\|_F^2,\end{aligned}$$

其中，最后一个不等式是由算法 2的第 3 行得出的。最后，根据最优低秩近似代价随行数增加而单调变化的性质，我们有$\mathsf{OPT}_s \leq \mathsf{OPT}_t$ 因此，

$$\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top \mathbf{V}^{(t)}\|_F^2 \leq \mathsf{OPT}_t + \varepsilon \cdot \|\mathbf{A}^{(s)}\|_F^2,$$

如所愿。 □

接下来需要界定算法 2的重调度次数：

引理 2.2。 算法 *2* 的重调度次数至多为$\mathcal{O}\left(\frac{k}{\varepsilon}\log(ndM)\right)$。

证明。 由于$\mathbf{A}$ 的每个条目都是绝对值至多为 M 的整数，因此$\mathbf{A}$ 的平方弗罗贝尼乌斯范数至多为 $(nd)M^2$。此外，$\mathbf{A}$ 的每个条目都是整数；当其首次变为非零时，平方弗罗贝尼乌斯范数至少为 1。因此，从$\mathbf{A}$ 首次出现非零条目开始，平方弗罗贝尼乌斯范数最多可以增加 $(1+\varepsilon)$ 的因子 $\log_{(1+\varepsilon)}(nd)M^2 = \mathcal{O}\left(\frac{1}{\varepsilon}\log(ndM)\right)$。每当 t 发生这种情况时，我们都会重新计算$\mathbf{A}^{(t)}$ 的右奇异值，使其成为因子$\mathbf{V}^{(t)}$的集合。因此，这些时刻产生的重调度次数至多为$\mathcal{O}\left(\frac{k}{\varepsilon}\log(ndM)\right)$。在其他所有时刻，我们保留相同的因子选择。因此，所需结论得证。 □

定理 1.1 由引理 2.1 和引理 2.2得出。

2.2 有界在线条件数

接下来，我们证明，在秩一扰动下，最优秩 k 子空间所需的重调度至多为常数。特别地，只需考虑向矩阵添加单个行的情形：

引理 2.3。 设 $\mathbf{A}^{(t-1)} \in \mathbb{R}^{(t-1)\times d}$ 和 $\mathbf{A}^{(t)} \in \mathbb{R}^{t\times d}$，其中 $\mathbf{A}^{(t)}$ 是将行 $\mathbf{A}_t$ 追加到 $\mathbf{A}^{(t-1)}$ 后得到的。设 $\mathbf{V}_{t-1}^*$ 和 $\mathbf{V}_t^*$ 分别为 $\mathbf{A}^{(t-1)}$ 和 $\mathbf{A}^{(t)}$ 的最优秩-k 子空间（即前 k 个右奇异向量张成的空间）。那么 重调度量$(\mathbf{V}_{t-1}^*, \mathbf{V}_t^*) \leq 8$。

证明。两个子空间之间的重调度量定义为其对应的正交投影矩阵之差的平方弗罗贝尼乌斯范数。我们首先考虑由最优秩-k 子空间诱导的协方差矩阵。具体而言，考虑协方差矩阵 $\mathrm{B}_{t-1} = (\mathrm{A}_{(t-1)}^*)^\top \mathrm{A}_{(t-1)}^*$ 和 $\mathrm{B}_t = (\mathrm{A}^{(t)})^\top \mathrm{A}^{(t)}$。于是有关系 $\mathrm{B}_t = \mathrm{B}_{t-1} + \mathrm{A}_t^\top \mathrm{A}$，即 B_t 是通过对 B_{t-1} 进行秩一半正定（PSD）更新得到的。根据 Eckart-Young 定理（参见定理 1.5），子空间 V_t^* 是由 B_t 的前 k 个特征向量张成的；对于 V_{t-1}^* 和 B_{t-1}，同理。

接下来，我们考虑$\mathbf{B}_t$ 和$\mathbf{B}_{t-1}$的特征子空间的交集。首先，我们要证明这两个子空间的交集维数至少为 $k-1$，即 $\dim(\mathbf{V}_{t-1}^* \cap \mathbf{V}_t^*) \geq k-1$。令$\mathbf{S}_a$ 为与$\mathbf{A}_t$正交的子空间，因此$\mathbf{S}_a = \{\mathbf{v} \in \mathbb{R}^d : \mathbf{A}_t\mathbf{v} = 0\}$ 且 $\dim(\mathbf{S}_a) = d-1$。考虑交集$\mathbf{W} = \mathbf{V}_{t-1}^* \cap \mathbf{S}_a$。根据子空间维数性质：

$$\dim(\mathbf{W}) = \dim(\mathbf{V}_{t-1}^*) + \dim(\mathbf{S}_a) - \dim(\mathbf{V}^{t-1} \cap \mathbf{S}_a).$$

由于 $\dim(bV_{t-1}^*) = k$，$\dim(\mathbf{S}_a) = d-1$，且 $\dim(\mathbf{V}_{t-1}^* \cap \mathbf{S}_a) \leq d$，我们有：

$$\dim(\mathbf{W}) \geq k + (d-1) - d = k-1.$$

现在我们分析 $\mathbf{W}$ 中向量的性质。设 $\mathbf{w} \in \mathbf{W}$。由于 $\mathbf{w} \in \mathbf{S}a$，我们有 $\mathbf{A}_t\mathbf{w} = 0$。注意到

$$\mathbf{B}_t\mathbf{w} = (\mathbf{B}_{t-1} + \mathbf{A}_t^\top \mathbf{A}_t)\mathbf{w} = \mathbf{B}_{t-1}\mathbf{w} + \mathbf{A}_t^\top(\mathbf{A}_t\mathbf{w}) = \mathbf{B}_{t-1}\mathbf{w}.$$

因此，$\mathbf{W}$ 是一个同时含于$\mathbf{B}_{t-1}$ 与$\mathbf{B}_t$的子空间。

设 $\lambda_1 \geq \ldots \geq \lambda_d$为 $\mathbf{B}_{t-1}$的特征值，且 $\mu_1 \geq \ldots \geq \mu_d$为 $\mathbf{B}_t$的特征值。由于$\mathbf{B}_t$ 是 $\mathbf{B}_{t-1}$的秩一半正定更新，根据定理 1.9，特征值交错：

$$\mu_1 \geq \lambda_1 \geq \mu_2 \geq \lambda_2 \geq \ldots \geq \mu_k \geq \lambda_k \geq \mu_{k+1} \geq \lambda_{k+1} \ldots.$$

由于 W $\subseteq$ V$^*t-1$，子空间 W 由 B_{t-1} 对应于特征值λ_1，.....，λ_k的特征向量张成。因为对 w $\in W$ 有 B_tw $= \mathrm{B}_{t-1}$w，这些向量同样是 B_t 的特征向量，且具有相同的特征值。由于 $\lambda_k \geq \mu_{k+1}$，与子空间 W 相关联的特征值大于或等于 B_t的第$(k+1)$个特征值。因此，W 必定是 B_t的前$(k+1)$特征子空间的一个子空间。于是，W $\subseteq$ V$^*_{t-1} \cap$(V$^*_t \cup$ {u})，其中 u 是 B_t 对应于特征值 μ_{k+1}的特征向量。由于 dim(W) $\geq k-1$，我们已证明 dim(V$^*_{t-1} \cap$ V$^*_t$) $\geq k-2$。

现在，令 $\mathbf{P}_{t-1}$ 和 $\mathbf{P}_t$ 分别为到 $\mathbf{V}^*_{t-1}$ 和 $\mathbf{V}^*_t$ 的正交投影矩阵。令$\mathbf{W}_{\text{int}} = \mathbf{V}^*_{t-1} \cap \mathbf{V}^*_t$，且 $\mathbf{P}$ shared 为到 $\mathbf{W}_{\text{int}}$ 的投影。若 dim($\mathbf{W}_{\text{int}}$) $= k$，则 $\mathbf{V}^*t-1 = \mathbf{V}^*_t$，因此重调度为 $\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_F^2 = 0$。

否则 ，若 dim($\mathbf{W}_{int}$) $= 1$，我们可以将正交投影矩阵分解为 ：

$$\mathbf{P}_{t-1} = \mathbf{P}_{\text{shared}} + \mathbf{u}_1\mathbf{u}_1^\top, \qquad \mathbf{P}^*t = \mathbf{P}_{\text{shared}} + \mathbf{u}_2\mathbf{u}_2^\top,$$

其中 $\mathbf{u}_1$,$\mathbf{u}_2$ 是 与 $\mathbf{W}_{\text{int}}$ 正交的单位向量。重调度为 $\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_F^2$。因此，我们有 e

$$\mathbf{P}_t - \mathbf{P}_{t-1} = (\mathbf{P}_{\text{shared}} + \mathbf{u}_2\mathbf{u}_2^\top) - (\mathbf{P}_{\text{shared}} + \mathbf{u}_1\mathbf{u}_1^\top) = \mathbf{u}_2\mathbf{u}_2^\top - \mathbf{u}_1\mathbf{u}_1^\top,$$

因此，根据广义三角不等式，

$$\text{Recourse}(\mathbf{P}_t, \mathbf{P}_{t-1}) \leq 2\|\mathbf{u}_1\mathbf{u}_1^\top\|_F^2 + 2\|\mathbf{u}_2\mathbf{u}_2^\top\|_F^2.$$

由于$\mathbf{u}_1$ 与$\mathbf{u}_2$ 均为单位向量，因此我们有 Recourse($\mathbf{P}_t$,$\mathbf{P}_{t-1}$) ≤ 4。使用四个单位向量的同一证明表明，若 dim($\mathbf{W}_{int}$) $= 2$，则 Recourse($\mathbf{P}_t$,$\mathbf{P}_{t-1}$) ≤ 8。 □

最后，我们指出，由于秩一扰动较为简单，任何维护 SVD 的算法只需对 SVD 执行一次秩一更新，所需时间为$\mathcal{O}\left((n+d)k + k^3\right)$，对于维度至多为 $n \times d$ [Bra06] 的矩阵。特别地，请注意，每当矩阵的单个条目被更改、矩阵的任意一行被更改、向矩阵添加新行，或删除矩阵的现有行时，这些都对应于该矩阵的秩一扰动。因此，我们立即得到如下推论：

定理 2.4. 存在一种算法，可在任意秩一更新序列（包括条目修改、行修改、行插入或行删除）下维护矩阵的最优秩-k 近似，并产生总重配置次数 $O(n)$，针对包含 n 次更新的数据流。此外，若每行的维数为 d，则更新时间为 $O\left((n+d)k + k^3\right)$。

根据引理 2.3中的陈述，我们立即得到一个算法，它在行到达模型下处理长度为 n 的数据流时，所需的重调度次数为 $\mathcal{O}(n)$。由于 n 次重调度过多，我们采用以下标准方法来减少矩阵中的有效行数。

定义 2.5 (投影代价保持草图)。**给定矩阵** $\mathbf{A} \in \mathbb{R}^{n\times d}\mathbf{M} \in \mathbb{R}^{m\times d}(1+\varepsilon)$ 是矩阵 $\mathbf{A}$ 的投影代价保持草图，当且仅当对于所有投影矩阵 $\mathbf{P} \in \mathbb{R}^{d\times d}$，

$$(1-\varepsilon)\|\mathbf{A}-\mathbf{AP}\|_F^2 \le \|\mathbf{M}-\mathbf{MP}\|_F^2 \le (1+\varepsilon)\|\mathbf{A}-\mathbf{AP}\|_F^2.$$

直观地说，投影代价保持草图是一种草图矩阵，它能够近似捕获投影掉任意秩为 k 的子空间后的残差质量。下面的定理表明，可以通过在线岭杠杆抽样获得投影代价保持草图。

定理 2.6 （ [BDM+20]中的定理 3.1）：给定精度参数 $\varepsilon > 0$、秩参数$k > 0$以及矩阵 $\mathbf{A} = \mathbf{a}_1 \circ \ldots \circ \mathbf{a}_n \in \mathbb{R}^{n\times d}$，其行以数据流形式依次到达，且条件数为 κ，则存在一种算法，可输出矩阵 $\mathbf{M}$，其中包含 $\mathcal{O}\left(\frac{k}{\varepsilon^2}\log\, n\log^2\kappa\right)$ 个经过重新缩放的 $\mathbf{A}$ 的行，使得

$$(1-\varepsilon)\|\mathbf{A}-\mathbf{A}_{(k)}\|_F^2 \le \|\mathbf{M}-\mathbf{M}_{(k)}\|_F^2 \le (1+\varepsilon)\|\mathbf{A}-\mathbf{A}_{(k)}\|_F^2,$$

从而以高概率，$\mathbf{M}$ 是 $\mathbf{A}$ 的秩为 k 的投影代价保持草图 .

特别地，如果数据流的在线条件数上界为$\mathrm{poly}(n)$，那么定理 2.6 表明，在线岭杠杆抽样能够得到大小为 $\mathcal{O}\left(\frac{k}{\varepsilon^2}\log^3 n\right)$的在线核心集。应用引理 2.3可知，只需始终维护在线核心集的最优秩为 k 的子空间，即可得到定理 1.2。

3 相对误差算法

在本节中，我们给出在数据流的所有时刻实现 $(1+\varepsilon)$-乘法相对误差的算法。设 $t \in [n]$，设 $\mathbf{x}_t \in \{-\Delta, \ldots, \Delta-1, \Delta\}^d$，并设 $\mathbf{X}_t = \mathbf{x}_1 \circ \ldots \circ \mathbf{x}_t$。对每个 $\mathbf{X}_t \in \mathbb{R}^{t\times d}$，我们计算 $\mathbf{X}_t$ 的低秩近似 $\mathbf{U}_t\mathbf{V}_t$，其中 $\mathbf{U}_t \in \mathbb{R}^{t\times k}$ 与 $\mathbf{V}_t \in \mathbb{R}^{k\times d}$ 是 $\mathbf{X}_t$ 的因子。我们滥用记号，将 $\mathbf{V}_t$ 写成 $\mathbb{R}^d$ 中由 k 个点组成的集合 V_t。注意，量$\sum_{t=1}^n |V_t \setminus V_{t-1}|$ 是重调度或一致性代价的一个上界。因此，在本节中，我们互换地将该量称为重调度或一致性代价，因为我们可以对这一更精细的量给出下界。

我们的算法对底部 $\sqrt{k}$ 个奇异值（即顶部 k）的贡献进行分类讨论。若该贡献较小，则可以替换相应的奇异向量，而不会显著增大误差。因此，每当新的一行到达时，我们只需用该新行替换其中一个底部奇异向量。另一方面，若该贡献较大，则意味着最优解不能在这些方向上被投影得过远，否则最优低秩近似代价也会显著增加。因此，我们可以在每个时刻直接选择最优的前 k 个右奇异向量集合，因为新子空间与旧子空间之间将存在大量重叠。我们的完整算法见算法 3。

正确性。为便于讨论，我们将最优低秩近似代价尚未增加超过一个乘法近似因子（$1+\varepsilon$）的这段时间称为一个时期。我们首先证明，在 HEAVY 设为 FALSE 的时期内，我们的算法在所有时刻 t 的正确性。也就是说，我们证明，在一个时期内，当最大的 k 个右奇异值中最小的 $\sqrt{k}$个奇异值不贡献显著质量时，我们的算法在所有时刻 t 都能维持对最优低秩近似代价的（$1+\varepsilon$）乘法−近似。

算法 3 低回溯低秩近似的相对误差算法 输入：输入矩阵 A 的行 a1, . . . , a$_n$ $\in \mathbb{R}^{n\times d}$，其整数项的绝对值不超过$M$输出：在所有时刻对最优低秩近似代价的 $(1+\frac{\varepsilon}{4})$-近似 1: 对每一行 a$_t$ 执行 2: OPT $\leftarrow \sum_{i=k+1}^{d}\sigma_i^2$(A$_t$) 3: HEAVY $\leftarrow$ TRUE，$C\leftarrow 0$, $c\leftarrow 0$4: 如果 OPT $\geq (1+\frac{\varepsilon}{4})\,C$或（$c=\sqrt{k}$且 HEAVY $=$FALSE）或（$c=k$且 HEAVY $=$TRUE），则 5: $C\leftarrow$OPT，$c\leftarrow 0$6: V $\leftarrow$ Recluster(A$^{(t)}$, k) 7: 如果 $\sum_{i=k-\sqrt{k}}^{k}\sigma_i^2$(A$^{(t)}$) $\geq \frac{\varepsilon}{3}\cdot C$则 8: HEAVY $\leftarrow$TRUE 9: 否则 10: HEAVY $\leftarrow$FALSE 11: 结束如果 12: 否则 13: 如果 HEAVY $=$FALSE，则 14: 令 v 为 V 中使 $\|$A$^{(s)}$v$\|_2^2$最小的单位向量，其中 s 是最近一次调用 Recluster 的时刻 15: 将 v 替换为 $\frac{1}{\|\mathbf{a}_t\|_2}\cdot\mathbf{a}_t$ 中的向量 16: $c\leftarrow c+1$17: 否则如果 HEAVY $=$TRUE，则 18: 如果 $\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}\mathbf{V}^\top\mathbf{V}\|_F^2\geq(1+\frac{\varepsilon}{2})$ OPT，则 19: V $\leftarrow$ Recluster(A$^{(t)}$, k) 20: $c\leftarrow c+1$21: 结束如果 22: 结束如果 23: 结束如果 24: 返回 V 25: 结束循环

引理 3.1. 考虑时刻 s，在此期间 c 被重置为 0。假设 HEAVY 在时刻 s 被设置为FALSE，并且对于 $r\leq\sqrt{k}$，c 在接下来的 r 步内未被重置为 0 。令 $\mathbf{V}^{(t)}$ 为 $\mathbf{V}$ 在时刻t的输出。则对所有 $t\in[s,s+r]$，$\mathbf{V}^{(t)}$ 给出了$\mathbf{A}^{(t)}$ 的最优低秩近似代价的一个 $(1+\frac{\varepsilon}{2})$-近似。

证明。考虑 $\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2$。令 $\mathbf{V}^{(t')}$ 为矩阵 $\mathbf{V}^{(s)}$，其中对应于 $\mathbf{A}^{(s)}$ 最小的 $t-s$ 个奇异值的 $t-s$ 个向量被替换为行$\mathbf{a}_{s+1},\ldots,\mathbf{a}_t$。由 $\mathbf{v}$ 的最优性，我们有

$$\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2\leq\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t')})^\top\mathbf{V}^{(t')}\|_F^2.$$

由于对所有 $t\in[s,s+\sqrt{k}]$，$\mathbf{a}_t\in\mathbf{V}^{(t')}$，因此我们有

$$\|\mathbf{A}^{(t)}-\mathbf{A}^{(t)}(\mathbf{V}^{(t')})^\top\mathbf{V}^{(t')}\|_F^2=\|\mathbf{A}^{(s)}-\mathbf{A}^{(s)}(\mathbf{V}^{(t')})^\top\mathbf{V}^{(t')}\|_F^2.$$

换句话说，行 $\mathbf{a}_{s+1}, \ldots, \mathbf{a}_t$ 不会对$\mathbf{V}^{(t')}$ 的低秩近似代价产生贡献，因为它们包含在$\mathbf{V}^{(t')}$的张成空间中。接下来需要给 $\|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(t')})^\top \mathbf{V}^{(t')}\|_F^2$一个上界。我们有

$$C = \sum_{i=k+1}^{d} \sigma_i^2(\mathbf{A}_s) = \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top \mathbf{V}^{(s)}\|_F^2$$

且 $\sum_{i=k-\sqrt{k}}^{k} \sigma_i^2(\mathbf{A}_s) < \frac{\varepsilon}{3} \cdot C$因为 HEAVY 被设为 FALSE。注意，由于 $t \in [s, s+\sqrt{k}]$，因此$\mathbf{A}^{(t')}$包含前 $k - \sqrt{k}$ 个奇异向量$\mathbf{A}^{(s)}$。因此可得

$$\|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(t')})^\top \mathbf{V}^{(t')}\|_F^2 \leq \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top \mathbf{V}^{(s)}\|_F^2 + \sum_{i=k-\sqrt{k}}^{k} \sigma_i^2(\mathbf{A}_s) \leq \left(1 + \frac{\varepsilon}{3}\right) \cdot C.$$

由于$\mathbf{A}^{(t)}$ 的最优低秩近似代价至少与$\mathbf{A}^{(s)}$ 在 $t > s$时的最优低秩近似代价一样大，因此$\mathbf{V}^{(t)}$ 为$\mathbf{A}^{(t)}$在所有 $t \in [s, s+\sqrt{k}]$时的最优低秩近似代价提供了一个 $(1+\frac{\varepsilon}{3})$-近似。

□

接下来，我们展示在 HEAVY 被设为 TRUE 的时期内的各个时刻 t ，算法的正确性。也就是说，我们证明，在一个时期内，当顶部 k 个右奇异值中的底部 $\sqrt{k}$ 个奇异值确实贡献了显著质量时，我们的算法在所有时刻 t 都维持着对最优低秩近似代价的 $(1+\varepsilon)$-乘法近似。

引理 3.2. 考虑一个时间 t ，在此期间HEAVY 被设置为TRUE。令$\mathbf{V}^{(t)}$ 为$\mathbf{V}$ 在时间 t的输出。则 $\mathbf{V}^{(t)}$ 给出了$\mathbf{A}^{(t)}$的最优低秩近似代价的 $(1+\frac{\varepsilon}{2})$-近似。

证明。令 OPT $= \sum_{i=k+1}^{d} \sigma_i^2(\mathbf{A}^{(t)})$。我们有两种情形。要么 $\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}\mathbf{V}^{(t-1)\top}\mathbf{V}^{(t-1)}\|_F^2 < (1+\frac{\varepsilon}{2})\cdot$OPT，要么 $\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}\mathbf{V}^{(t-1)\top}\mathbf{V}^{(t-1)}\|_F^2 \geq (1+\frac{\varepsilon}{2})\cdot$OPT。

在前一种情形中，$\mathbf{V}^{(t-1)}$ 已经是$\mathbf{A}^{(t)}$ 的最优低秩近似代价的 $(1+\frac{\varepsilon}{2})$-近似，并且该算法将$\mathbf{V}^{(t)} = \mathbf{V}^{(t-1)}$设为，从而$\mathbf{V}^{(t)}$ 也是$\mathbf{A}^{(t)}$的最优低秩近似代价的$(1+\frac{\varepsilon}{2})$-近似。

在后一种情况下，算法将 $\mathbf{V}^{(t)}$ 设为 Recluster$(\mathbf{A}^{(t)}, k)$ 的输出，即 $\mathbf{A}^{(t)}$ 的前k 个特征向量；此时 $\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top \mathbf{V}^{(t)}\|_F^2 = \text{OPT}$。因此，在这两种情况下，$\mathbf{V}^{(t)}$都以 $(1+\frac{\varepsilon}{2})$ 倍近似于 $\mathbf{A}^{(t)}$ 的最优低秩近似的代价。

□

由引理 3.1和引理 3.2可知，任何时刻的正确性都成立：

引理 3.3。在任何时刻 $t \in [n]$，算法 3都以 $(1+\frac{\varepsilon}{2})$倍近似于 $\mathbf{A}^{(t)}$的最优低秩近似的代价。

证明。设$\mathbf{V}^{(t)}$ 为时刻$\mathbf{V}$ 的输出 t。我们首先考虑 t 不被重置为零且 c HEAVY 被设为FALSE的时刻。根据引理 3.1，在这些时刻，输出$\mathbf{V}^{(t)}$ 以 $(1+\frac{\varepsilon}{2})$倍近似于$\mathbf{A}^{(t)}$ 的最优低秩近似的代价。

接下来考虑以下时刻： t 此时 c 未重置为零，且 HEAVY 被设为 TRUE。根据引理 3.2，输出 $\mathbf{V}^{(t)}$ 为 $(1+\frac{\varepsilon}{2})$-近似，给出了这些时刻 $\mathbf{A}^{(t)}$ 的最优低秩近似的代价。

最后，我们考虑以下时刻： t 此时 c 重置为零。在这些时刻，算法将 $\mathbf{V}^{(t)}$ 设为 Recluster($\mathbf{A}^{(t)}$,k) 的输出，即 A 的前 k 个特征向量；此时$\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 = \mathrm{OPT}$。因此，算法 3 在所有时刻 $t \in [n]$ 均提供 $\mathbf{A}^{(t)}$ 的最优低秩近似代价的 $(1+\frac{\varepsilon}{2})$-近似。

□

接下来，我们对各时期之间的重调度进行界定，此时前 $\sqrt{k}$ 个奇异值对前 k 个奇异值的显著质量没有贡献。

引理 3.4。考虑一个时刻 t，在此期间 HEAVY 被设为 TRUE。令 $\mathbf{V}^{(t)}$ 为时刻 t时 $\mathbf{V}$的输出。假设 $\mathbf{V}^{(t-1)}$ 不是最优低秩近似代价的 $(1+\frac{\varepsilon}{2})$-近似。那么 重调度($\mathbf{V}^{(t)}$,$\mathbf{V}^{(t-1)}$) $\leq \sqrt{k}$。

证明。令 s 为 HEAVY 被设为 TRUE 的最近时刻，因此 $s \leq t-1$。令 $\mathbf{V}^{(s)}$表示 $\mathbf{A}^{(s)}$ 的前 k 个右奇异向量，并注意到根据将 HEAVY 设为 TRUE 的条件，

$$\sum_{i=k-\sqrt{k}}^{k} \sigma_i^2(\mathbf{V}^{(s)}) \geq \frac{\varepsilon}{3} \cdot \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2.$$

现在，令 r 为首次设置 $\mathbf{V}^{(t-1)}$ 的时刻，因此 $\mathbf{V}^{(t-1)}$ 是前 k 个右奇异向量，它们属于 $\mathbf{A}^{(r)}$。由于 $r \geq s$，根据奇异值交错定理，即 定理 1.9，可得

$$\sigma_i(\mathbf{V}^{(r)}) \geq \sigma_i(\mathbf{V}^{(s)}),$$

对于所有 $i \in [d]$。因此，

$$\sum_{i=k-\sqrt{k}}^{k} \sigma_i^2(\mathbf{V}^{(r)}) \geq \frac{\varepsilon}{3} \cdot \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2,$$

因为根据定义，所有奇异值均为非负数。

反设我们有

$$\mathrm{Recourse}(\mathbf{V}^{(t)}, \mathbf{V}^{(t-1)}) = \mathrm{Recourse}(\mathbf{V}^{(t)}, \mathbf{V}^{(r)}) > \sqrt{k}.$$

那么至少有 $\sqrt{k}$个奇异向量被$\mathbf{V}^{(t-1)}$ 替换了，因此根据极小极大定理，即定理 1.8，

$$\begin{aligned}\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 &> \|\mathbf{A}^{(r)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2 + \sum_{i=k-\sqrt{k}}^{k} \sigma_i^2(\mathbf{V}^{(r)}) \\ &\geq \left(1+\frac{\varepsilon}{3}\right) \cdot \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2.\end{aligned}$$

此外，由于 $\mathbf{V}^{(t)}$ 是 $\mathbf{A}^{(t)}$的前 k 个右奇异向量，因此 $\|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2$是时刻 t的最优低秩近似代价。换句话说，时刻 t 的最优低秩近似代价将大于 $(1+\frac{\varepsilon}{3}) \cdot \|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2$。

另一方面，由于 s和 t处于同一时期，因此

$$\|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(s)}\|_F^2 \leq \|\mathbf{A}^{(t)} - \mathbf{A}^{(t)}(\mathbf{V}^{(t)})^\top\mathbf{V}^{(t)}\|_F^2 \leq \left(1+\frac{\varepsilon}{4}\right)\|\mathbf{A}^{(s)} - \mathbf{A}^{(s)}(\mathbf{V}^{(s)})^\top\mathbf{V}^{(r)}\|_F^2,$$

这就产生了矛盾。因此可得，重调度($\mathbf{V}^{(t)}$, $\mathbf{V}^{(t-1)}$) $\leq \sqrt{k}$。

□

重调度。 首先，我们对以下情况的重调度进行界定：前 $\sqrt{k}$个奇异值在前 k 个右奇异值中不贡献显著质量。

引理 3.5。 假设在时间 s 将 HEAVY 设为 FALSE ，并在时间 s 将 c 重置为 0。如果在接下来的 r 步内没有将 c 重置为 0 ，对于 $r \leq \sqrt{k}$，那么 $\sum_{i=s+1}^{s+r}$ 重调度 $(\mathbf{V}^{(s)}, \mathbf{V}^{(s-1)}) \leq r$。

证明。 设 s 为一个时刻，在该时刻 c 被重置为 0 且 HEAVY 被设为 FALSE。那么在接下来的r 步中，每当收到一行新数据时，算法 3 就会用新行替换 $\mathbf{V}$ 中的一行。因此，重调度次数至多为 r。 □

此处我们指出分析中一个容易被忽略的微妙之处。我们的总体策略是证明，每当最优低秩近似的代价翻倍时，我们应承担 $O\left(k^{1.5}\right)$ 的重调度开销。因此，人们可能会认为，由于矩阵包含由 poly(n) 界定的整数项，最优低秩近似的代价最多只能翻倍 $\mathcal{O}(\log n)$ 次，因为它至多为 poly(n)。

遗憾的是，存在一些反哈达玛整数矩阵，其维数为 $n \times d$，但最优低秩代价却小到 $\exp(-\Omega(k))$。因此，最优代价可以翻倍 $\mathcal{O}(k)$ 次，从而导致总重配置次数达到 $\mathcal{O}\left(k^{2.5}\right)$，这大得令人难以接受。相反，我们证明，当最优低秩代价呈指数级小的时候，矩阵的秩也必然相当小，这意味着我们的算法在重调度方面不会像满秩情形下那样庞大。为此，我们需要一个结构性质（参见 引理 3.6），用于描述最优低秩近似的代价，并将其参数化，以适用于秩为 $r > k$的一般矩阵。这是为了处理最优低秩近似的代价可能相对于 k呈指数级小的情形。因此，当最大 $\sqrt{k}$ 个奇异值中最小的那些 k 不具有显著质量时，我们得到所有时期总重配置次数的如下上界。

引理 3.6。 设$\mathbf{A} \in \mathbb{Z}^{n \times d}$ 是一个秩为 $r > k$ 、且条目绝对值不超过 M的整数矩阵。那么，$\mathbf{A}$ 的最优低秩近似的代价至少为$(ndM^2)^{-\frac{k}{r-k}}$。

证明。 设$\mathbf{A} \in \mathbb{Z}^{n \times d}$ 是一个秩为 $r > k$ 、且条目绝对值不超过 M的整数矩阵。不失一般性，设 $n \geq d$。令 $\sigma_1 \geq \ldots \geq \sigma_d \geq 0$ 为$\mathbf{A}$的奇异值，令 $\lambda_1 \geq \ldots \geq \lambda_d \geq 0$ 为$\mathbf{A}^\top\mathbf{A}$对应的特征值。令 $p(\lambda) = \lambda^{d-r} \prod_{i \in [r]}(\lambda - \lambda_i)$。注意，由于 $\mathbf{A}$ 的条目全为整数，因此 $\mathbf{A}^\top\mathbf{A}$ 的条目也全为整数，故 $p(\lambda)$ 的系数全为整数。特别地，由于 $p(\lambda)$ 中 λ^{d-r} 的系数是$\mathbf{A}^\top\mathbf{A}$的非零特征值之积，因此 $\prod_{i=1}^{r} \lambda_i \geq 1$。由于 $\lambda_i = \sigma_i^2$，且对所有$i \in [d]$都有 $\sigma_i \geq 0$ ，我们也有 $\prod_{i=1}^{r} \sigma_i \geq 1$。

此外，平方弗罗贝尼乌斯范数满足

$$\sum_{i=1}^{d} \lambda_i = \sum_{i=1}^{d} \sigma_i^2 = \|\mathbf{A}\|_F^2 \leq ndM^2.$$

因此，对于所有 $i \in [d]$， $\lambda_i \leq ndM^2$ 。由此，

$$\lambda_{k+1}^{r-k} \geq \prod_{i=k+1}^{r} \lambda_i \geq \frac{1}{(ndM^2)^k} \prod_{i=1}^{r} \lambda_i \geq \frac{1}{(ndM^2)^k}.$$

因此我们有 $\lambda_{k+1} \geq (ndM^2)^{-\frac{k}{r-k}}$。由此可得，最优低秩近似代价为

$$\sqrt{\sum_{i=k+1}^{d} \lambda_i} \geq \lambda_{k+1} \geq (ndM^2)^{-\frac{k}{r-k}}.$$

□

接下来，我们界定当前 k 个奇异值中较小的 $\sqrt{k}$ 个贡献显著质量时，各时期之间的重调度量。

引理 3.7。假设在时刻 t，HEAVY 被设为 TRUE，且在接下来的 r 步内，c 未被重置为 0，其中 $r \leq k$。那么 $\sum_{i=t+1}^{t+r} \text{重调度}(\mathbf{V}^{(t)}, \mathbf{V}^{(t-1)}) \leq r\sqrt{k}$。

证明。根据 引理 3.4，可得 $\text{重调度}(\mathbf{V}^{(t)}, \mathbf{V}^{(t-1)}) \leq \sqrt{k}$ 对所有 $i \in [t+1, t+r]$ 均成立。因此，总重配置次数至多为 $r\sqrt{k}$。□

我们分析因最优低秩近似的代价加倍而重置计数器 c 时的总重调度量。

引理 3.8。设 T 为满足以下条件的时刻集合：c 被设为 0，因为 $\sum_{i=k+1}^{d} \sigma_i^2(\mathbf{A}_t) \geq 2C$。那么 $\sum_{t \in T} \text{重调度}(\mathbf{V}^{(t)}, \mathbf{V}^{(t-1)}) \leq O\left(\frac{k}{\varepsilon} \log^2(ndM)\right)$。

证明。我们定义时刻 $\tau_1 < \tau_2$，并将 T 分解为 τ_1 之前的时刻、τ_1 与 τ_2 之间的时刻，以及 τ_2 之后的时刻。具体而言，令 t_0 为最优低秩近似代价首次非零的时刻，也就是输入矩阵的秩为 $k+1$ 的最早时刻。令 T_0 为时刻 t_0 的最优低秩近似代价。将每个时期 i 定义为最优低秩近似代价至少为 $\left(1+\frac{\varepsilon}{4}\right)^i \cdot T_0$ 且小于 $\left(1+\frac{\varepsilon}{4}\right)^{i+1} \cdot T_0$ 的那些时刻。

令 τ_1 为输入矩阵的秩为 $k+1$ 的最早时刻。注意，在时刻 τ_1 之前，我们可以通过将每个新出现的线性无关行加入低秩子空间，维护矩阵的整个行空间，从而始终保持最优低秩近似代价不变。此过程在时刻 τ_1 之前产生的重调度总量至多为 k。

接下来，令 τ_2 为输入矩阵的秩首次不超过 $2k$ 的时刻。我们分析时刻 τ_1 与 τ_2 之间的重调度。根据引理 3.6，对于条目为整数、其值有界于 M 且秩为 r 的矩阵，其最优低秩近似的代价至少为 $(ndM^2)^{-\frac{k}{r-k}}$。因此，如果矩阵的秩 r 至少为 $k+2^i$ 且小于 $k+2^{i+1}$，则最优低秩近似的代价至少为 $(ndM^2)^{-\frac{k}{2^i}}$。所以，在最优低秩近似的代价至少达到 $(ndM^2)^{-100}$ 之前，最多只能经过 $O\left(\frac{1}{\varepsilon}\frac{k}{2^i}\right)$ 个时期。令 j 为任意一个时期的索引，在该时期内最优低秩近似的代价超过 $(ndM^2)^{-\frac{k}{2^i}}$。由于截至时期 j 已到达的 A 的行的张成空间的总维数为 r，因此，通过重新计算顶部特征空间所产生的重调度至多为 $O(r-k) = O(2^i)$。由于在最优低秩近似的代价至少达到 $(ndM^2)^{-100}$ 之前，最多只能经过 $O\left(\frac{1}{\varepsilon}\frac{k}{2^i}\log(ndM)\right)$ 个时期，因此，对于矩阵秩至少为 $k+2^i$ 且小于 $k+2^{i+1}$ 的时刻，一致性代价至多为 $O\left(\frac{k}{\varepsilon}\log(ndM)\right)$。因此，时刻 τ_1 与 τ_2 之间的一致性总代价为 $\sum_{i=0}^{\log k} O\left(\frac{k}{\varepsilon}\log(ndM)\right) = O\left(\frac{k}{\varepsilon}\log^2(ndM)\right)$。

注意，在时刻 τ_2 之后，最优低秩近似的代价至少为 $(ndM^2)^{-100}$。由于平方弗罗贝尼乌斯范数至多为 ndM^2，因此低秩代价也至多为 ndM^2。所以，在时刻 τ_2 之后最多只能经过 $\mathcal{O}\left(\frac{1}{\varepsilon}\log(ndM)\right)$ 个时期。由于重新计算截至该时刻已到达的 $\mathbf{A}$ 的前缀的顶部特征空间，每个时期产生的重调度为 $\mathcal{O}(k)$。因此，由 T 所表示的时刻集合在 τ_2 之后产生的总重调度为 $\mathcal{O}\left(\frac{k}{\varepsilon}\log(ndM)\right)$。

总而言之，我们可以将 T 分解为 τ_1之前的时刻、 τ_1 与 τ_2之间的时刻，以及 τ_2之后的时刻。在 τ_1 之前的 $t \in T$时刻，总重配置次数至多为$\mathcal{O}(k)$。在 τ_1与 τ_2之间的 $t \in T$时刻，总重配置次数至多为$\mathcal{O}\left(\frac{k}{\varepsilon}\log^2(ndM)\right)$。在 τ_2 之后的$t \in T$时刻，总重配置次数至多为$\mathcal{O}(k\log(ndM))$。因此，$t \in T$时刻的总重配置次数至多为$\mathcal{O}\left(\frac{k}{\varepsilon}\log^2(ndM)\right)$。 □

最后，还需要界定从一个时期过渡到另一个时期时的总重配置次数。具体而言，我们界定 t 时刻的重调度次数；在这些时刻，自上一个时期开始以来，最优低秩近似代价已增加了一个乘法因子为$(1+\varepsilon)$的近似。

引理 3.9. 设 T 为将 c 设置为0的时刻集合。则 $\sum_{t\in T}\mathrm{Recourse}(\mathbf{V}^{(t)},\mathbf{V}^{(t-1)}) \le O\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。

证明。 注意， c 只能因以下三种不同原因之一而被重置：

(1) HEAVY =TRUE且 $c=k$(2)
HEAVY = FALSE且 $c=\sqrt{k}$(3)$-$
$\sum_{i=k+1}^{d}\sigma_i^2(\mathbf{A}_t) \ge 2C$

在所有这三种情形中，算法都会调用Recluster($\mathbf{A}$, k)，产生重调度 k。注意到，对于矩阵$\mathbf{A}$ ，当其有 n 行时，计数器 c 超过 $\sqrt{k}$ 的次数至多为 $\frac{n}{\sqrt{k}}$ 次。因此前两种情形至多发生 $\frac{n}{\sqrt{k}}$ 次，从而前两种情形所贡献的总重配置次数至多为 $n\sqrt{k}$。

仍需考虑最优低秩近似的代价相对于上一次 C 被设置时至少翻倍的那些步骤所产生的总重配置次数。根据引理 3.8，这些时刻产生的重调度至多为 $\mathcal{O}\left(\frac{k}{\varepsilon}\log^2(ndM)\right)$。因此，总重配置次数至多为

$$\mathcal{O}\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right).$$ □

利用 引理 3.5、引理 3.7 和 引理 3.9，我们随后对我们算法的总重配置次数给出上界 .

引理 3.10. 总重配置次数（即 算法 3 在输入矩阵 $\mathbf{A}\in\mathbb{R}^{n\times d}$ 上、整数项绝对值以 M 为界时）为 $\mathcal{O}\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。

证明。 我们首先考虑时刻 t ，其中 c 未被重置为零，且HEAVY 被设为 FALSE 。根据引理 3.5，任意连续集合中 r 个此类步骤上的重调度至多为 r。因此，在 n 行的数据流上，所有将 HEAVY 设为 FALSE 的步骤所产生的总重配置次数至多为 n。

我们接下来考虑那些时刻 t ，其中 c 未被重置为零，且 HEAVY 被设为 TRUE。由引理 3.7，任意一段不间断序列中 r 这些步骤的重调度至多为 $r\sqrt{k}$。因此

在由 n 行构成的数据流上，所有将 HEAVY 设为 TRUE 的步骤所产生的总重配置次数至多为$n\sqrt{k}$。

最后，我们考虑 t 被重置为零的 c 时刻。根据引理 3.9，所有步骤产生的总重配置次数至多为$\mathcal{O}\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。因此，总重配置次数为$\mathcal{O}\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。

□

利用引理 3.3 和引理 3.10，我们可以给出该子程序的形式化保证。

引理 3.11. 给定一个输入矩阵$\mathbf{A}\in\mathbb{R}^{n\times d}$，其整数项的绝对值不超过 M，算法 *3* 可实现对最优低秩近似代价的 $(1+\frac{\varepsilon}{2})$-近似，并且其重调度次数为$\mathcal{O}\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。

证明。正确性由引理 3.3 可知，而重调度次数的上界由引理 3.10可得。

□

算法 4相对误　　差低秩近似算法，具有重调度 $\frac{k^{3/2}}{\varepsilon^2}\cdot$polylog$(ndM$　　—　　$)$

输入： 输入矩阵 $\mathbf{A}\in\mathbb{R}^{n\times d}$的行 $\mathbf{a}_1,\ldots,\mathbf{a}_n$，其整数项的绝对值至多为 M**输出：** 在所有时刻对最优低秩近似的代价的 $(1+\frac{\varepsilon}{4})$-近似

1: **对于** 每一行$\mathbf{a}_t$ **执行**2: 根据在线岭杠杆分数 ▷定理 2.6**对** $\mathbf{a}_t$ 进行采样3: 运行 算法 3 处理由采样行诱导的数据流4: **结束循环**

为了减少输入中的行数，我们再次应用 定理 2.6，结合引理 3.11，得到主要结果 定理 1.3 对于 算法 4。

定理 1.3。 设$\mathbf{A}\in\mathbb{Z}^{n\times d}$是一个整数矩阵，其条目的绝对值不超过 M。存在一种算法，可在所有时刻实现对$\mathbf{A}$ 的最优低秩近似代价的$(1+\varepsilon)$近似，并实现重调度 $\frac{k^{3/2}}{\varepsilon^2}\cdot\operatorname{polylog}(ndM)$。

证明。令 $\varepsilon\in\left(0,\frac{1}{100}\right)$。根据定理 2.6，对于截至时间 t 已采样的$\mathbf{M}$的各行，其最优低秩近似可实现 $(1+\frac{\varepsilon}{10})$-近似，其代价不超过截至时间 t到达的$\mathbf{A}$ 的最优低秩近似代价的。因此，只需证明我们在所有时刻都能为矩阵$\mathbf{M}$ 的最优低秩近似代价提供 $(1+\frac{\varepsilon}{2})$-近似。因此，我们转而考虑由$\mathbf{M}$的各行构成的新数据流 $\in\mathbb{R}^{m\times d}$，其中$m=\frac{k}{\varepsilon^2}\cdot\operatorname{polylog}(ndM)$，且$\mathbf{M}$ 的各项均为绝对值不超过 $M\cdot\operatorname{poly}(n)$的整数。

考虑以 M 为输入的算法 3。根据引理 3.11，正确性显然成立，因此还需重新设定引理 3.11 中的参数，以分析总重配置次数。根据引理 3.10，对于各项绝对值不超过 M 的输入矩阵 A $\in\mathbb{R}^{n\times d}$，总重配置次数为 O $\left(n\sqrt{k}+\frac{k}{\varepsilon}\log^2(ndM)\right)$。因此，对于具有 $\frac{k}{\varepsilon^2}\cdot$ polylog(ndM) 行且各项绝对值不超过 $M\cdot$ poly(n) 的输入矩阵 M，总重配置次数为 $\frac{k^{3/2}}{\varepsilon^2}\cdot$ polylog(ndM)。

□

最后，我们指出，由于 定理 2.6 需要采样 $\mathcal{O}\left(\frac{k}{\varepsilon^2}\log\ n\log^2\kappa\right)$ 行，并且存在用于近似采样概率的输入稀疏算法 [CMM17]，，因此 算法 4可以使用 $\frac{kd}{\varepsilon}\cdot\operatorname{polylog}(ndM)$空间位数，并具有 $d\cdot\operatorname{poly}\left(k,\frac{1}{\varepsilon},\log(ndM)\right)$摊销更新时间。

4 重调度下界

本节将证明我们的重调度下界。核心思路是将数据流简单地划分为 $\Theta\left(\frac{1}{\varepsilon}\log\frac{n}{k}\right)$ 个阶段，使得最优低秩近似代价在每个阶段之间增加一个乘法 $(1+\mathcal{O}(\varepsilon))$ 因子。我们还设计输入矩阵，使得第 i 个阶段的最优解与第 $(i-1)$ 个阶段的最优解正交。因此，根据阶段 i的奇偶性，最优解要么是前 k 个标准基向量，要么是从 $k+1$ 到 $2k$ 的标准基向量；这意味着，为了始终保持 $(1+\varepsilon)$-近似，每个阶段之间都必须承担 $\Omega(k)$ 的重调度，从而得到所需的 $\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$ 下界。

定理 1.4。 对于任意参数 $\varepsilon > \log_n n$，都存在一个行序列 $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$，使得任何始终能对最优低秩近似的代价给出$(1+\varepsilon)$-近似的算法，都必须具有 $\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$ 的一致性代价。

证明。 我们将数据流划分为 $\Theta\left(\frac{1}{\varepsilon}\log\frac{n}{k}\right)$ 个阶段。设 $C>2$ 为我们稍后将确定的某个参数。如果 i 为奇数，则在第 i 阶段，我们加入 $(1+\varepsilon)^i$ 个标准基向量 $\mathbf{e}_1, \ldots, \mathbf{e}_k$。如果 i 为偶数，则在第 i 阶段，我们加入 $(1+C\varepsilon)^i$ 个标准基向量 $\mathbf{e}_{k+1}, \ldots, \mathbf{e}_{2k}$。注意，由于共有 $\Theta\left(\frac{1}{\varepsilon}\log\frac{n}{k}\right)$ 个阶段，且每个阶段插入 $(1+C\varepsilon)^i$ 个 k 行，因此在常数取值正确时，每行插入的副本总数至多为 $\mathcal{O}(n)\,2k$，这里的常数用于确定 $\Theta(\cdot)$ 记号；因此总共至多有 n 行。

我们指出，根据构造，在第 i 阶段之后，如果 i 是奇数，则最优秩-k 近似为标准基向量 $\mathbf{e}_1, \ldots, \mathbf{e}_k$；如果 i 是偶数，则最优秩 -k 近似为标准基向量 $\mathbf{e}_{k+1}, \ldots, \mathbf{e}_{2k}$。特别地，根据 Eckart-Young-Mirsky 定理（即定理 1.5），在第 i 阶段之后，如果 i 是奇数，则对底层矩阵的最优秩-k近似所产生的代价为 $k\cdot\left((1+C\varepsilon)^2+(1+C\varepsilon)^4+\ldots+(1+C\varepsilon)^{i-1}\right)$；如果 i 是偶数，则代价为 $k\cdot\left((1+C\varepsilon)+(1+C\varepsilon)^3+\ldots+(1+C\varepsilon)^{i-1}\right)$。注意，这两个量都至多为 $2k(1+C\varepsilon)^{i-1}$。因此，对于奇数阶段 i 之后的时刻 t，一个秩为-k 因子的矩阵 $\mathbf{M}$ 必须满足

$$\|\mathbf{X}^{(t)}-\mathbf{X}^{(t)}\mathbf{M}^{\dagger}\mathbf{M}\|_F^2 \le 2\varepsilon k(1+C\varepsilon)^{i-1},$$

为了成为最优低秩代价的 $(1+\varepsilon)$-近似。

令 $\mathbf{E}^{(1)}=\mathbf{e}_1\circ\ldots\circ\mathbf{e}_k$，且 $\mathbf{E}^{(2)}=\mathbf{e}_{k+1}\circ\ldots\circ\mathbf{e}_{2k}$。对于任意常数 $C>100$，可知，为了使 $\mathbf{M}$ 成为对 $\mathbf{X}^{(t)}$ 的最优低秩近似代价的 $(1+\varepsilon)$-近似，$\mathbf{M}$ 在 $\mathbf{E}^{(1)}$ 上的平方质量至少为 $k(1-2\varepsilon)$，即对于紧接奇数阶段 i之后的时间 t，$\|\mathbf{M}(\mathbf{E}^{(1)})^{\top}\mathbf{E}^{(1)}\|_F^2 \ge k(1-2\varepsilon)$。类似地，为了使 $\mathbf{M}$ 成为对 $\mathbf{X}^{(t)}$ 的最优低秩近似代价的 $(1+\varepsilon)$-近似，$\mathbf{M}$ 在 $\mathbf{E}^{(2)}$ 上的平方质量至少为$k(1-2\varepsilon)$，对应于紧接奇数阶段 i之后的时间 t。然而，由于 $\mathbf{E}^{(1)}$ 和 $\mathbf{E}^{(2)}$ 不相交，因此可知 $\mathbf{M}$ 在每个阶段之间必须进行 $\Omega(k)$ 次重调度。由于共有 $\Omega\left(\frac{1}{\varepsilon}\log\frac{n}{k}\right)$ 个阶段，因此总重配置次数必须为 $\Omega\left(\frac{k}{\varepsilon}\log\frac{n}{k}\right)$。 □

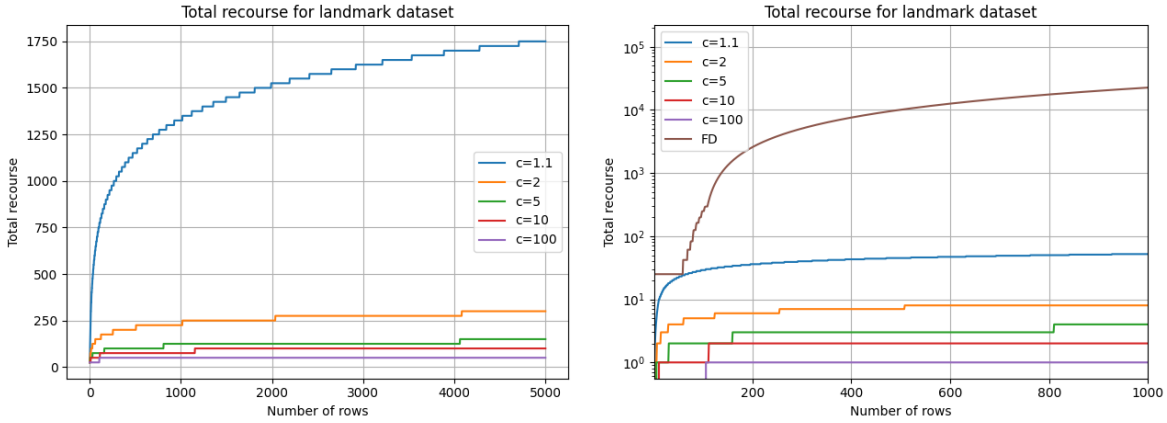
5 实证评估

我们介绍了在一个大型真实世界数据集上的实证评估，并将我们算法所得解的质量与最优低秩近似解的质量进行了比较。

我们讨论了在合成数据集和真实世界数据集上开展的一系列额外实验。所有实验均使用 Python 3.10.4 版本，在搭载 AMD Ryzen 7 5700U 中央处理器的 64位操作系统上进行。该系统配备 8GB 内存和 8 个核心，每个核心的基础时钟频率为 1.80 GHz。代码可公开访问：https://github.com/samsonzhou/consistent-LRA。

k	$(1+\varepsilon)$	中位数	标准差	Mean
25	1.1	1.000	0.0000	1.0000
	2	1.000	0.0000	1.0000
	5	1.0000	0.0367	1.0016
	10	1.0000	2.4463	1.598
	100	1.1907	51.9353	7.8882

表 1：landmark 数据集在 150 至 5000 次更新期间，不同准确率参数取值下成本比率的中位数、标准差和均值



(a) 重调度比较　　(b) 重调度比较

图 1：$k=25$、$c=(1+\varepsilon)\in\{1.1、2.5、5、10、100\}$的补救比较

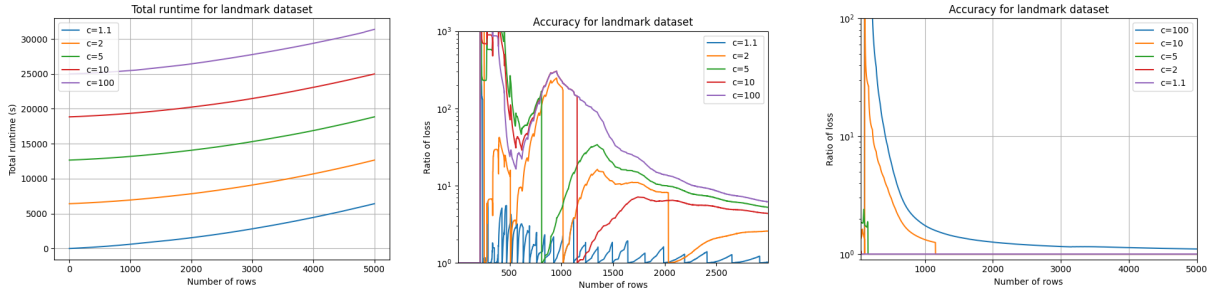
5.1 landmark 数据集

在本节中，我们重点评估算法 2在 SuiteSparse矩阵集合中的 landmark 数据集上的表现 [DH 11]，该数据集常用于低秩近似的基准比较，例如 [BWZ19]。该数据集总共包含71952 行和 $d=2704$ 个特征。

实验设置。鉴于理论结果证明我们的算法只需少量重调度，我们首先比较算法 2输出的解的代价与最优低秩近似的代价。然而，确定每个时刻的最优代价在计算上开销很高，是主要瓶颈。具体而言，对于长度为n的数据流，基线算法的运行时间为 $n\cdot\mathcal{O}(n^\omega)=\Omega(n^3)$，其中 $n\approx d$，$\omega\approx 2.37$ 是矩阵乘法

矩阵乘法 [ADV+25]。因此，我们取数据流中的前 $n = 5000$ 行，目标是对每个大小为 $n' \times d$且满足 $n' \leq n$ 的前缀矩阵执行低秩近似。具体来说，我们计算了 $k = 25$在 $c = (1+\varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$条件下两种代价的运行时间和比值，结果见图 2；相应的中心统计量列于表 1，尽管算法 2 只能保证加性误差，而非乘法误差。最后，我们在图 1 中比较了我们算法的重调度情况，以及用于在线低秩近似的标准算法 FrequentDirections（标记为 FD） [GLPW16]。

结果与讨论。 我们的结果表明，Frequent Directions 等在线低秩近似方法与我们专为实现低回溯而设计的算法，在质量方面存在显著差距。具体而言，对于 $n = 5000$，Frequent Directions 的重调度次数为 121904，而我们的算法的重调度次数为 100 至 300，相差超过 400 倍。此外，我们的结果表明，我们算法的近似保证在实践中其实相当不错，尤其是行数增加时；我们认为，图 2b中的较大方差，是因为最优低秩近似代价相对于加性 Frobenius 误差而言很小。因此，我们的实证评估似乎有力地证明，与现有在线低秩近似算法相比，我们的算法能显著减少重调度；以下还将介绍若干其他实验。



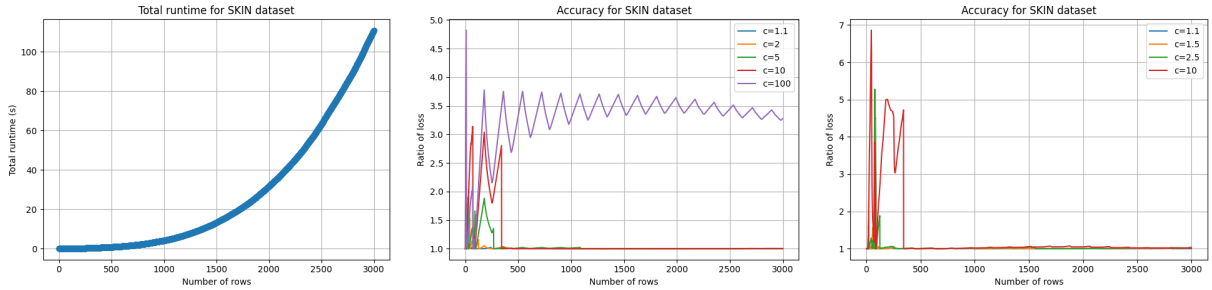
(a) 运行时间对比　(b) 近似效果对比　(c) 近似效果对比

图 2：在 landmark 数据集上的运行时间和近似效果，针对 $k = 25$，$c = (1+\varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$

5.2 SkinSegmentation 数据集

我们评估算法 2在皮肤分割（SKIN）数据集上的表现 [BD12] ，该数据集来自 UCI 存储库 [MK87], ，常用于无监督学习任务的基准比较，例如 [BEL+20、EMNZ23、WZZ23]。该数据集共包含 245057 张以 B、G、R 值编码的人脸图像，采集自 Color FERET 图像数据库以及德克萨斯大学达拉斯分校 Productive Aging 实验室的 PAL 人脸数据库。这些人脸来自不同年龄组（青年、中年和老年）、种族组（白人、黑人和亚裔）以及不同性别。在该数据集中，50859 张图像为皮肤样本，其余 194198 张图像为非皮肤样本，任务是对每张图像所属的类别进行分类。

实验设置。 在实验中，我们仅使用了前 3000 张皮肤图像，并移除了标签，因此目标是对剩余皮肤图像的 B、G、R 值进行低秩近似



(a) SKIN 数据集上的运行时间　(b) $k = 1$ 的近似　(c) $k = 2$ 的近似

图 3：SKIN 数据集上的运行时间和近似。图 3a 考察 $k = 1$ 和 $c = 1.1$，图 3b 考察 $k = 1, c = (1 + \varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$，图 3c 考察 $k = 2, c = (1 + \varepsilon) \in \{1.1, 1.5, 2.5, 10\}$

。由于我们的理论保证表明，解只会改变少数几次，因此我们比较了算法 2 输出的解的代价与最优低秩近似的代价。具体而言，我们计算了两种代价的比值：对于$k = 1$，在 $c = (1 + \varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$ 条件下；对于 $k = 2$，在 $c = (1 + \varepsilon) \in \{1.1, 1.5, 2.5, 10\}$ 条件下，尽管算法 2 的形式化保证涉及对加性误差进行上界估计。

结果与讨论。在图 3中，我们绘制了数据流持续期间每个时刻算法 2 输出的解的代价与最优低秩近似代价之比。随后，我们给出该比值在不同 k 取值及 SKIN 数据集准确率参数下的中心统计量，即均值、标准差和最大值，见表 2。

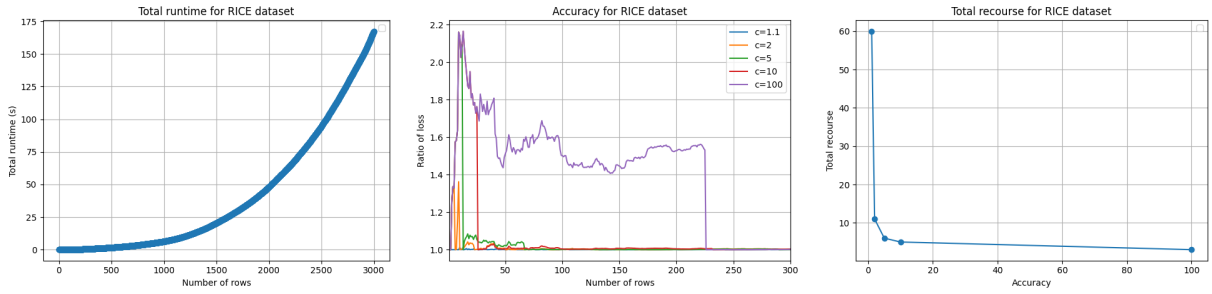
结果表明，正如预期，随着 $c = (1 + \varepsilon)$ 从100 降至1，我们的算法对最优解的近似效果会更好。当最优低秩近似代价足够大时，我们的算法实现了良好的乘法近似。因此，我们认为，图 3c 开头出现尖峰的主要原因，是最优低秩近似代价相对于加性 Frobenius 误差而言非常小。有些出人意料的是，尽管最坏情况理论保证表明我们的算法只能提供100倍近似，但它的实际表现要好得多，即实现了大约4倍近似。因此，我们的实证评估似乎提供了一个简单的概念验证，表明我们的最坏情况理论保证在实践中甚至可以更强。

5.3 RICE 数据集

接下来，我们考虑 RICE 数据集 [Ric19]，该数据集来自 UCI 仓库 [MK87]，，其目标是对土耳其种植的两种稻米进行分类。第一种是 Osmancik 品种，自 1997 年起便有大面积种植；第二种是 Cammeo 品种，种植历史始于 2014 [Ric19]。该数据集共包含两种稻米的 3810 张稻米颗粒图像，每粒稻米提取了 7 个形态学特征。具体而言，这些特征包括稻米颗粒的面积、周长、长轴长度、短轴长度、偏心率、凸面积和范围。

k	$(1+\varepsilon)$	Mean	标准差	Max
1	1.1	1.0006	0.0022	1.0383
	2	1.0088	0.0479	1.7870
	5	1.0374	0.1341	2.6038
	10	1.1324	0.4167	4.8201
	100	3.2971	0.4532	4.8201
2	1.1	1.0016	0.0069	1.1521
	1.5	1.0148	0.1327	5.1533
	2.5	1.0371	0.2430	5.2778
	10	1.3175	0.9238	6.8602

表 2：SKIN 数据集中不同 k 取值和准确率参数下成本比率的平均值、标准差和最大值。



(a) RICE 上的运行时间　　(b) RICE 上的近似　　(c) RICE 上的重调度

图 4：RICE 数据集上的运行时间和近似。 图 4a 考察 $k=1$， $c=10$，而图 4b 和 图 4c 考察 $k=1$，$c=(1+\varepsilon)\in\{1.1,\ 2.5,\ 5,\ 10,\ 100\}$

评估摘要。在实验中，我们对 RICE 数据集提供的七项特征进行了低秩近似。我们将算法 2输出解的代价与 $k=1$ 在 $c=(1+\varepsilon)\in\{1.1、2.5、5、10、100\}$下最优低秩近似的代价进行了比较。结果汇总于图 4，其中图 4a 绘出了算法的运行时间，算法 2输出解的代价与最优低秩近似代价之比则绘于图 4b。

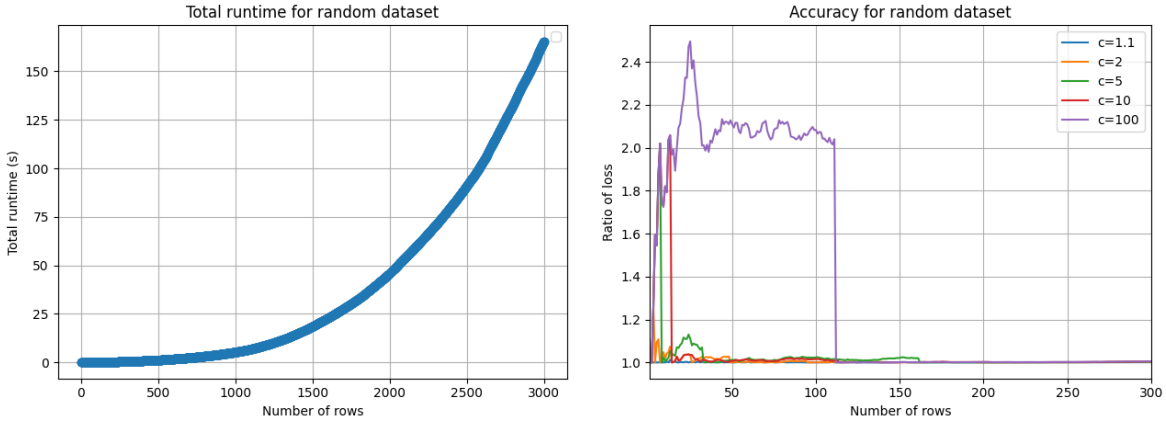
我们的结果表明，与皮肤分割数据集上的情况类似，随着 $c=(1+\varepsilon)$ 从 100 降至 1，我们的算法对最优解的近似效果更好。 图 4b 在行数较少时再次出现多个尖峰，这很可能是因为最优低秩近似代价相对于加性 Frobenius 误差而言非常小。此外，我们的结果再次呈现出一个有些出人意料的现象：与最坏情况理论保证相比，我们的算法提供了相对较好的近似效果；也就是说，尽管最坏情况保证仅能达到 2近似，100近似。

最后，我们注意到，正如预期，随着对因子的精度要求降低，我们算法的重调度量也随之减少。这是因为，随着矩阵不断变化，较粗略的因子表示所需的更新频率较低。

5.4 随机合成数据集

我们生成了一个随机合成数据集，包含 3000 行和 4 列，整数项介于 0 到 100 之间，随后按列进行了归一化。我们再次比较了 算法 2 输出的解的代价与最优低秩近似的代价，对 $k = 1$ 所有$c = (1+\varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$。我们在 图 5 中总结了结果。具体而言，我们首先在 图 5a 中绘制了算法的运行时间。然后，我们在图 5b中绘制了 算法 2输出的解的代价与最优低秩近似代价之比。

与皮肤分割和水稻数据集类似，随着 $c = (1+\varepsilon)$ 从 100 降至 1，我们的算法给出的近似结果越来越接近最优解；当行数较少时，可能会出现一些尖峰，这很可能是因为最优低秩近似代价相较于加性 Frobenius 误差非常小。此外，我们的结果明显优于最坏情况理论保证，近似比约为 2.5，而理论保证为 100 倍近似。



(a) 随机数据集上的运行时间　　(b) 随机数据集上的近似结果

图 5：随机数据集上的运行时间和近似结果。 图 5a 展示 $k = 1$， $c = 10$，而图 5b 展示 $k = 1$， $c = (1+\varepsilon) \in \{1.1, 2.5, 5, 10, 100\}$

致谢

David P. Woodruff 获美国海军研究办公室项目（编号 N000142112647）及西蒙斯研究员奖的部分资助。Samson Zhou 获美国国家科学基金会 CCF-2335411 项目的部分资助。Samson Zhou 谨此感谢橡树岭联合大学（ORAU）拉尔夫·E·鲍尔青年教师发展奖提供的资助。 引理 2.3 的结果是在 Gemini DeepThink 的 AI 辅助下得出的，并由作者独立验证。

参考文献

[AA15] Charu C Aggarwal 和 Charu C Aggarwal。文本数据挖掘。施普林格，2015年。3

[ADV+25] Josh Alman、Ran Duan、Virginia Vassilevska Williams、Yinzhan Xu、Zixuan Xu 和 Renfei Zhou。更大的非对称性带来更快的矩阵乘法。载于《2025年ACM-SIAM离散算法年会论文集》，SODA，第2005–2039页，2025年。25[BBLS23] Sayan Bhattacharya、Niv Buchbinder、Roie Levin 和 Thatchaphol Saranurak。追逐正体。载于第64届IEEE计算机科学基础年会，FOCS，第1694–1714页，2023。2[BBMN15] Nikhil Bansal、Niv Buchbinder、Aleksander Madry 和 Joseph Naor。一种用于k-服务器问题的多对数竞争算法。《ACM学报》，62(5)：40:1–40:49，2015年。2[BCG+24] Sayan Bhattacharya、Martín Costa、Naveen Garg、Silvio Lattanzi 和 Nikos Parotsidis。具有快速更新时间和较小重调度量的全动态k-聚类。载于第65届IEEE计算机科学基础年会，FOCS，第216–227页，2024。6[BCR23] Sébastien Bubeck、Christian Coester 和 Yuval Rabani。随机化 k-服务器猜想是错误的！载于第55届ACM计算理论研讨会论文集，STOC，第581–594页，2023。2[BD12] Rajen Bhatt 和 Abhinav Dhall。皮肤分割。UCI机器学习资料库，2012年。数字对象标识符：https://doi.org/10.24432/C5T30C。25[BDM+20] Vladimir Braverman、Petros Drineas、Cameron Musco、Christopher Musco、Jalaj Upad- hyay、David P. Woodruff 和 Samson Zhou。在线模型和滑动窗口模型中的近最优线性代数。载于第61届IEEE计算机科学基础年会，FOCS，第517–528页，2020年。5、6、7、8、15[BEL+20] Michele Borassi、Alessandro Epasto、Silvio Lattanzi、Sergei Vassilvitskii 和 Morteza Zadimoghaddam。k聚类问题的滑动窗口算法。载于《神经信息处理系统进展》第33卷：神经信息处理系统年会，NeurIPS，2020年。5、25[BGL+18] Vladimir Braverman、Elena Grigorescu、Harry Lang、David P. Woodruff 和 Samson Zhou。滑动窗口上的近最优不同元素计数与重击者检测。载于《近似、随机化与组合优化：算法与技术》，APPROX/RANDOM，第7:1–7:22页，2018年。5[BGW+26] Vladimir Braverman、Sumegha Garg、Chen Wang、David P. Woodruff 和 Samson Zhou。滑动窗口模型中基于有限信息的在线学习。载于ACM-SIAM离散算法研讨会论文集，SODA，2026年。5[BHM+21] Vladimir Braverman、Avinatan Hassidim、Yossi Matias、Mariano Schain、Sandeep Silwal 和 Samson Zhou。通过重要性采样提升流式算法的对抗鲁棒性。载于《神经信息处理系统进展》第34卷：神经信息处理系统年会，NeurIPS，2021年。2[BKV07] Robert M Bell、Yehuda Koren 和 Chris Volinsky。Netflix奖的BellKor方案。KorBell团队致Netflix的报告，2007年。2

[BKV08] Robert M Bell、Yehuda Koren 和 Chris Volinsky。Netflix奖的 BellKor 2008 年解决方案。AT&T 研究院统计研究部，1(1)，2008年。2[BKV10] Robert M Bell、Yehuda Koren 和 Chris Volinsky。现在大家齐心协力：Netflix奖视角。Chance，23(1):24–29，2010年。2[BLMZ23] Jeremiah Blocki、Seunghoon Lee、Tamalika Mukherjee 和 Samson Zhou。滑动窗口模型中的差分隐私 l_2-重击者。发表于第十一届国际学习表征会议（ICLR），2023。5[Bra06] Matthew Brand。薄奇异值分解的快速低秩修正。《线性代数及其应用》，415(1):20–30，2006年。14[BW18] Ainesh Bakshi 和 David P. Woodruff。距离矩阵的亚线性时间低秩近似。发表于神经信息处理系统进展第 31 卷：神经信息处理系统年会（NeurIPS），2018年。2[BWZ19] Frank Ban、David P. Woodruff 和 Qiuyi (Richard) Zhang。正则化加权低秩近似。发表于神经信息处理系统进展第 32 卷：2019年神经信息处理系统年会（NeurIPS），第 4061–4071 页，2019年。24[BWZ21] Vladimir Braverman、Viska Wei 和 Samson Zhou。滑动窗口上的对称范数估计与回归。发表于计算与组合数学——第27届国际会议 COCOON 论文集，2021。5[CGK+17] Flavio Chierichetti、Sreenivas Gollapudi、Ravi Kumar、Silvio Lattanzi、Rina Panigrahy 和 David P. Woodruff。低秩近似算法。发表于第34届国际机器学习会议论文集（ICML），第 806–814 页，2017年。2[CHP+19] Vincent Cohen-Addad、Niklas Hjuler、Nikos Parotsidis、David Saulpic 和 Chris Schwiegelshohn。完全动态一致设施选址。发表于神经信息处理系统进展第 32 卷：神经信息处理系统年会（NeurIPS），第 3250–3260 页，2019年。2[CJWZ25] T.-H. Hubert Chan、Shaofeng H.-C. Jiang、Tianyi Wu 和 Mengshi Zhao。具有近乎最优一致性的在线聚类。发表于第十三届国际学习表征会议（ICLR），2025年。6[CJY+25] Vincent Cohen-Addad、Shaofeng H.-C. Jiang、Qiaoyuan Yang、Yubo Zhang 和 Samson Zhou。滑动窗口模型中的公平聚类。发表于第十三届国际学习表征会议（ICLR），2025年。5[CLMP22] Vincent Cohen-Addad、Silvio Lattanzi、Andreas Maggiori 和 Nikos Parotsidis。在线一致相关聚类。发表于国际机器学习会议（ICML），第 4157–4179 页，2022年。6[CMM17] Michael B. Cohen、Cameron Musco 和 Christopher Musco。通过岭杠杆分数采样实现输入稀疏度时间低秩近似。发表于第二十八届 ACM-SIAM 离散算法研讨会论文集（SODA），第 1758–1777 页，2017年。2, 6, 22

[CMS13] Michael S. Crouch、Andrew McGregor 和 Daniel M. Stubbs。滑动窗口模型中的动态图。发表于算法——ESA 第21届欧洲年会论文集，第 337–348 页，2013年。5[CW09] Kenneth L. Clarkson 和 David P. Woodruff。流式模型中的数值线性代数。发表于第41届 ACM计算理论研讨会论文集（STOC），第 205–214 页，2009年。9[CW17] Kenneth L. Clarkson 和 David P. Woodruff。输入稀疏度时间内的低秩半正定近似。发表于第二十八届 ACM-SIAM 离散算法研讨会论文集（SODA），第 2061–2072 页，2017年。2[DFL+24a] Paul Duetting、Federico Fusco、Silvio Lattanzi、Ashkan Norouzi-Fard 和 Morteza Zadimoghaddam。一致性次模最大化。发表于第四十一届国际机器学习会议（ICML），OpenReview.net，2024。6[DFL+24b] Paul Dütting、Federico Fusco、Silvio Lattanzi、Ashkan Norouzi-Fard、Ola Svensson 和 Morteza Zadimoghaddam。一致性的代价：具有常数重调的次模最大化。CoRR，abs/2412.02492，2024。6[DGIM02] Mayur Datar、Aristides Gionis、Piotr Indyk 和 Rajeev Motwani。维护滑动窗口上的数据流统计。SIAM计算杂志，31(6):1794–1813，2002。5[DH11] Timothy A Davis 和 Yifan Hu。佛罗里达大学稀疏矩阵集合。ACM数学软件汇刊（TOMS），38(1):1–25，2011。24[EMMZ22] Alessandro Epasto、Mohammad Mahdian、Vahab S. Mirrokni 和 Peilin Zhong。通过基于分桶的草图改进聚类和覆盖的滑动窗口算法。发表于2022年 ACM-SIAM 离散算法研讨会论文集（SODA），第 3005–3042 页，2022年。5[EMNZ23] Alessandro Epasto、Vahab Mirrokni、Shyam Narayanan 和 Peilin Zhong。具有基于距离的隐私保护的 k均值聚类。发表于神经信息处理系统进展第 36 卷：神经信息处理系统年会（NeurIPS），2023。25[EY36] Carl Eckart 和 Gale Young。用一个低秩矩阵近似另一个矩阵。Psychometrika，1(3):211–218，1936。10[Fis05] Steve Fisk。关于厄米矩阵特征值的柯西交错定理的一个极简证明。arXiv预印本 math/0502408，2005。11[FKL+91] Amos Fiat、Richard M. Karp、Michael Luby、Lyle A. McGeoch、Daniel Dominic Sleator 和 Neal E. Young。竞争式分页算法。《算法杂志》，12(4):685–699，1991。2[FLNS21] Hendrik Fichtenberger、Silvio Lattanzi、Ashkan Norouzi-Fard 和 Ola Svensson。一般度量空间中的一致性 k-聚类。发表于2021年 ACM-SIAM 离散算法研讨会论文集（SODA），第 2660–2678 页，2021年。2, 6

[FS25] Sebastian Forster 和 Antonis Skarlatos。一种具有最优重调度的动态一致性 k-中心聚类。载于《2025年ACM-SIAM离散算法年会》论文集，SODA，第 212–254 页，2025年。6[GGK16] Albert Gu、Anupam Gupta 和 Amit Kumar。延迟处理的力量：在线维护常数竞争比斯坦纳树。《SIAM计算杂志》，45(1):1–28，2016年。2[GK14] Anupam Gupta 和 Amit Kumar。支持删除操作的在线斯坦纳树。载于第二十五届ACM-SIAM离散算法年会论文集，SODA，第 455–467 页，2014年。2[GKLX21] Xiangyu Guo、Janardhan Kulkarni、Shi Li 和 Jiayi Xian。一致性 k-中位数：更简单、更优且更稳健。载于第二十四届人工智能与统计国际会议，AISTATS，第 1135–1143 页，2021年。6[GKS14] Anupam Gupta、Amit Kumar 和 Cliff Stein。在线维护分配：匹配、调度与流。载于第二十五届ACM-SIAM离散算法年会论文集，SODA，第 468–479 页，2014年。2[GL20] Anupam Gupta 和 Roie Levin。具有有界重调整的全动态子模覆盖。载于第61届IEEE计算机科学基础年会，FOCS，第 1147–1157 页，2020年。2[GLPW16] Mina Ghashami、Edo Liberty、Jeff M. Phillips 和 David P. Woodruff。Frequent Directions：简单且确定性的矩阵草图。《SIAM计算杂志》，45(5):1762–1792，2016年。5, 6, 25[Hwa04] Suk-Geun Hwang。厄米矩阵特征值的柯西交错定理。《美国数学月刊》，111(2):157–159，2004年。11[Ira96] Sandy Irani。分页置换的竞争分析。载于《在线算法：最新进展》，《计算机科学讲义》第 1442 卷，第 52–73 页。施普林格，1996年。2[JKLV19] Mohammad Reza Karimi Jaghargh、Andreas Krause、Silvio Lattanzi 和 Sergei Vassilvitskii。一致性在线优化：凸优化与子模优化。载于第二十二届人工智能与统计国际会议，AISTATS，第 2241–2250 页，2019年。6[JRP04] Anil K Jain、Arun Ross 和 Salil Prabhakar。生物特征识别导论。《IEEE视频技术电路与系统汇刊》，14(1):4–20，2004年。3[JWZ22] Rajesh Jayaram、David P. Woodruff 和 Samson Zhou。面向数据流和滑动窗口的真正完美采样器。载于 PODS ’22：国际数据管理会议，第 29–40 页，2022年。5[KMS26] Michael Kapralov、Cameron Musco 和 Kshiteej Sheth。汉克尔矩阵的亚线性时间低秩近似。载于《2026年ACM-SIAM离散算法年会》论文集，SODA，第 3201–3248 页，2026年。2[Kor09] Yehuda Koren。Netflix大奖的 BellKor 方案。Netflix奖文档，81(2009):1–10，2009年。2

[LHG+23] Jakub Lacki、Bernhard Haeupler、Christoph Grunau、Václav Rozhoň 和 Rajesh Jayaram。完全动态一致性 k 中心聚类，2023。2, 3, 6[LOP+15] Jakub Lacki、Jakub Ocwieja、Marcin Pilipczuk、Piotr Sankowski 和 Anna Zych。动态距离预言机的力量：斯坦纳树的高效动态算法。载于第四十七届 ACM 计算理论研讨会年会论文集，STOC，第 11–20 页，2015年。2[LV17] Silvio Lattanzi 和 Sergei Vassilvitskii。一致性 k 聚类。载于第 34 届国际机器学习会议，ICML，第 1975–1984 页，2017。2, 3, 6[Mir60] Leon Mirsky。对称规范函数与酉不变范数。《数学季刊》，11(1):50–59，1960。10[MK87] Kolby Nottingham Markelle Kelly、Rachel Longjohn。UCI机器学习资料库，1987。25, 26[MMS90] Mark S. Manasse、Lyle A. McGeoch 和 Daniel Dominic Sleator。服务器问题的竞争性算法。《算法杂志》，11(2):208–230，1990。2[MS24] Cameron Musco 和 Kshiteej Sheth。托普利茨矩阵的亚线性时间低秩近似。载于《2024年ACM-SIAM离散算法研讨会》论文集，SODA，第 5084–5117 页，2024。2[MSVW16] Nicole Megow、Martin Skutella、José Verschae 和 Andreas Wiese。重调度对在线最小生成树和旅行商问题的作用。《SIAM计算杂志》，45(3):859–880，2016年。2[MW17a] Cameron Musco 和 David P. Woodruff。核低秩近似能否达到输入稀疏度时间？载于《神经信息处理系统进展 30：神经信息处理系统年会》，2017。2[MW17b] Cameron Musco 和 David P. Woodruff。正半定矩阵的亚线性时间低秩近似。载于第 58 届 IEEE 计算机科学基础年会，FOCS，第 672–683 页，2017。2[NA12] Mark S. Nixon 和 Alberto S Aguado。计算机视觉中的特征提取与图像处理。学术出版社，2012年。3[NPW+26] Soham Nagawanshi、Shalini Panthangi、Chen Wang、David P. Woodruff 和 Samson Zhou。时间衰减模型上的学习增强矩估计。载于第十四届国际学习表征会议，ICLR，2026。5[Pro07] John G Proakis。数字信号处理：原理、算法与应用，第 4 版。培生印度教育，2007年。3[Ric19] Rice（Cammeo 和 Osmancik）。UCI机器学习资料库，2019。数字对象标识符：https://doi.org/10.24432/C5MW4Z。26[SMR08] Hinrich Schütze、Christopher D Manning 和 Prabhakar Raghavan。信息检索导论，第 39 卷。剑桥大学出版社，剑桥，2008 年。3

[SVWZ24] Zhao Song、Ali Vakilian、David P. Woodruff 和 Samson Zhou。关于社会公平的低秩近似与列子集选择。载于神经信息处理系统进展第 38 卷：神经信息处理系统年会，NeurIPS，2024 年。2[VVWZ23] Ameya Velingker、Maximilian Vötsch、David P. Woodruff 和 Samson Zhou。用于二元矩阵分解的快速（$1+\epsilon$）近似算法。载于国际机器学习会议（ICML），第 34952–34977 页，2023 年。2[WF02] Ian H Witten 和 Eibe Frank。数据挖掘：实用机器学习工具与技术（含 Java 实现）。ACM SIGMOD Record，31(1)：76–77，2002 年。3[WY23] David P. Woodruff 和 Taisuke Yasuda。在线 Lewis 权重采样。载于 2023 年 ACM-SIAM 离散算法研讨会论文集（SODA），第 4622–4666 页，2023 年。5[WZ21] David P. Woodruff 和 Samson Zhou。利用差分估计器获得对抗鲁棒数据流和滑动窗口的紧确界。载于第 62 届 IEEE 计算机科学基础年会（FOCS 2021），第 1183–1196 页，2021 年。5[WZZ23] David P. Woodruff、Peilin Zhong 和 Samson Zhou。滑动窗口模型中的近最优 k-聚类。载于神经信息处理系统进展第 36 卷：神经信息处理系统年会（NeurIPS），2023 年。5, 25