arXiv:2602.12209v1 [cs.CR] 12 Feb 2026

保守秘密需要良好的记忆：私有算法的空间下界

Alessandro Epasto [1] **Xin Lyu** [2] **Pasin Manurangsi** [3]

摘要

我们从内存效率的角度研究差分隐私的计算成本。尽管准确度与差分隐私之间的权衡已得到充分理解，但隐私在内存使用方面的固有成本仍很少受到探索。本文首次通过引入一种基于多参与者通信博弈的新颖证明技术，为用户级差分隐私确立了一个无条件空间下界。

我们方法的核心是，这种博弈以形式化方式将低内存隐私算法的困难性与“贡献封顶”的必要性联系起来，也就是追踪并限制那些对数据集产生不成比例影响的用户。我们证明，要赢得这一通信博弈，必须传递与过度活跃用户数量成正比的信息，这直接转化为空间下界。

举例来说，我们将这一框架应用于一项基础问题：估计数据流中不同元素的数量，并证明，在该问题的承诺变体中，要达到特定的误差率，任何差分隐私算法都需要至少 $\tilde{\Omega}(T^{1/3})$ 空间。这解决了文献中的一个开放问题（Jain 等人（Jain 等人，2023a）和 Cummings 等人（Cummings 等人，2025年）提出），并首次证明了在一项自然的统计估计任务中，差分隐私算法的空间复杂度与非隐私算法的 $O(1)$ 对应复杂度之间存在指数级分离。此外，我们还证明，这种基于通信理论的技术可推广到广泛的问题类别，从而为隐私中位数、分位数和最大值选择问题给出下界。

[1]Google Research，美国 [2]加州大学伯克利分校，美国 [3]Google Research，泰国。通讯作者：Xin Lyu <xinlyu@berkeley.edu>。

预印本。*2026*年*2*月*13*日。

1. 引言

随着人工智能日益成为日常生活的核心，科技公司必须处理越来越多的隐私数据。为确保这些数据得到妥善保护，机器学习领域的一项重要工作致力于设计既足够准确、具有实用价值，又足够高效、能够部署于真实系统中的隐私保护算法。在此背景下，黄金标准是设计在用户级差分隐私（以下简称 DP）方面具有强理论保证的算法。该领域的代表性成果包括针对众多基础机器学习和数据分析问题的高效实用算法，例如私密训练深度学习模型（Abadi 等人，2016）、私密生成合成数据（Jordon 等人，2018）、私密统计估计（Liu 等人，2022）、私密组合优化（Gupta 等人，2010）等等。

尽管取得了这些积极成果，研究人员仍经常遇到难以用所需效率实现隐私保护解决方案的问题。例如，考虑在数据流中隐私地估计不同元素数量这一基本问题，本文将其作为贯穿全文的示例。先前研究表明，利用输入流上的多项式空间，可以构造在误差方面经证明近乎最优的隐私算法（Cummings 等人，2025年；Jain 等人，2023a），但仍有一个开放问题：类似于非隐私场景，是否可能实现使用多对数空间且准确度相同的隐私算法?

隐私算法的下界。 在这类情况下，理想的效率目标是否在数学上确实与隐私要求不相容，仍难以确定。为此，最好通过建立可证明的不可能性结果来回答这一问题。幸运的是，关于准确度，已有多项研究阐明了准确度与隐私之间固有的权衡，并提出了通用的下界技术（例如，单向边际查询Bun 等人(2018）；Dwork 等人(2015），这些技术已被用于界定许多问题中隐私算法的准确度。至于效率，算法设计中最常见的两个衡量指标是：运行时间和内存。尽管已有一系列研究表明，某些任务在有隐私保护时比没有隐私保护时需要更多运行时间（Dwork 等人，2009年；Ullman 和 Vadhan，2020；Blocki

等人，2025年），但人们对隐私与内存之间权衡的理解还远远不足。

具体而言，就隐私与内存之间的权衡而言——这正是本文关注的重点——据我们所知，目前唯一已有的结果是Dinur 等人（2023）的开创性工作，该工作首次给出了差分隐私算法的空间界。该研究表明，在某些密码学假设下，一个经过专门设计、涉及加密的流式问题，若要以隐私方式求解，就需要大量空间（而非隐私方式求解时，所需空间则少得多，且呈指数级差距）。尽管这证明了隐私与低空间可能无法兼得，但该结果仍有一些局限，而本文试图加以解决。首先，这一下界依赖密码学假设，而理想情况下，我们希望证明内在于问题本身的无条件信息论界。其次，也是更重要的一点，证明技术依赖于人为构造一个困难问题，无法推广到现实世界中遇到的标准问题（例如标准统计估计问题）。最后，理想情况下，这一下界还应直观地解释为何某些自然的隐私算法问题需要大量空间。

我们的贡献。我们的工作克服了这些局限，并解决了先前工作提出的开放问题（Jain 等人，2023a；Cummings 等人，2025年）。我们提出了一种新颖的证明技术，通过设计一种新颖的通信博弈，为隐私算法建立无条件空间界。我们证明，该技术适用于多种广泛研究的自然统计估计问题，包括对数据流中不同元素的数量进行隐私计数、计算中位数和分位数，以及解决最大值选择问题（正式定义见第2节）。

具体而言，对于轮流模型中的不同元素问题[1]，，我们证明，先前由Cummings 等人（2025年）完成的工作在某些情形下基本达到了空间最优。在流式不同元素计数问题的一个承诺版本中，每位用户对数据流的贡献都有上限（详情见第2.1 节），Cummings 等人（2025年）展示了如何设计一种差分隐私算法，其误差为 $o(T^{1/3})$，空间复杂度为 $\widetilde{O}(T^{1/3})$ [2]。我们证明，为了利用这一承诺，（Cummings 等人，2025年）的空间使用量几乎是最优的：任何准确度相当的算法都必须使用 $T^{1/3-o(1)}$ 比特的空间。这首次表明，在一个自然的计算问题中，隐私会导致内存需求出现指数级分离。我们认为，我们的通信博弈技术可用于证明更广泛一类问题的界。

直觉：对贡献进行“封顶”需要空间。从实践角度来看，我们的证明技术将低内存隐私算法的困难性与一种广泛观察到的现象联系起来：实现用户级隐私通常需要对用户对算法的贡献进行封顶。事实上，许多隐私保护算法的设计中，一种广为人知的技术是限制任一用户能够向数据集提供的更新次数（这通常称为贡献界定或封顶，例如，Amin 等人（2019））。这样做通常足以实现强隐私保护（在添加适当缩放的噪声之后），因为限制离群用户对解决方案做出不成比例的贡献，可以保护其身份不被泄露。直观而言，追踪 k 已“过度活跃”且˜耗尽贡献预算的用户，至少需要 $O(k)$比特内存。我们工作的关键在于证明，对于某些问题，这种广泛使用（但十分耗费内存）的策略不仅有用，而且基本上不可或缺。这个结果或许有些违反直觉。的确，人们可能会设想许多低空间的替代方案，例如采用激进的下采样，在大幅减少空间使用的同时有效限制用户贡献。

为达到这一下界，我们设计了一种多参与者通信博弈，其中每位参与者都掌握来自一部分（可能重复的）用户的贡献。每位参与者需要（私下地）为其数据子集解决该问题，然后可以向下一位参与者传递有关哪些用户（最多有 k 个，占 n）已超出其贡献限额的有用（非隐私）信息。我们证明，要赢得˜这样的博弈，参与者需要通信 $O(k)$比特的信息，并将这一要求与私下解决该问题的不可能性结果联系起来。

组织结构。 在证明主要结果（定理 2.1）之前，我们首先简要回顾相关工作。接下来，在第 2节中，我们正式定义流式设置中的用户级差分隐私，并介绍我们的困难性构造。最后，在第 3节中，我们正式证明主要结果。

1.1. 相关工作

我们的工作涵盖多个领域，包括差分隐私中的下界技术，以及具有隐私保护的数据流处理。

隐私算法的下界。大量研究致力于理解差分隐私中的准确性—隐私权衡。这些研究包括重构攻击（Dinur 和 Nissim，2003年）及后续基于差异的下界（Muthukrishnan 和 Nikolov，2012；Nikolov 等，2013）；打包论证（Hardt 和 Talwar，2010）；指纹方法（Bun 等，2018；Dwork 等，2015年；Steinke 和 Ullman，2017；Lyu 和 Talwar，2025年）；以及其他研究（Bun 等，2015年；Acharya 等，2021；Cohen 等，

[1]在旋转门模型中（Muthukrishnan，2005），用户可以在每个时间步加入或离开集合。形式化定义见第2 节。[2]全文中，我们使用 $\tilde{O}$ 记号来忽略 T中的多对数因子。

2024）。读者还可参阅全面的教科书（Vadhan，2017），以进一步了解这一主题。

据我们所知，目前唯一已知能够区分差分隐私与非差分隐私流式算法的空间下界来自 Dinur 等人（Dinur 等人，2023）。在该研究中，作者设计了一种新颖的、与密码学相关的流式问题（其中加密密钥按特定顺序接收），并利用该问题证明（在密码学假设下），空间受限的算法无法以差分隐私方式解决该问题。相比之下，我们论文的目标是设计信息论的下界（即除了准确度之外不依赖任何假设），并将其应用于文献中已有大量研究的自然问题。

数据流处理算法。非隐私流式算法的相关文献浩如烟海，感兴趣的读者可参阅该领域的综述，例如 Muthukrishnan（2005），以了解更多详情。尽管如此，我们要指出的是，我们研究的问题——不同元素计数和分位数——是该领域研究最充分的问题之一。（例如，参见 Kane 等人（2010）；Cormode 等人（2023）及其中的参考文献。）

与我们的工作尤其相关的是将差分隐私应用于流式算法这一领域。在许多现实场景中，都存在数据流处理应用需要跟踪大量可能涉及隐私的事件的统计信息，包括数字广告、网络监控和数据库系统。这推动了流式处理中的持续发布差分隐私算法这一领域不断发展（Dwork 等人，2010）。除了前文已提及的 Jain 等人（2023a）和 Cummings 等人（2025）在旋转门模型中对不同元素计数的研究外，Bolot 等人（2013）和 Ghazi 等人（2023）还研究了更严格的收银机模型（用户只能加入集合，不能离开）中的这一问题及其变体。近期，Andersson 等人（2026）和 Aamand 等人（2026）改进了不同元素隐私计数的误差率，但并未着重改进 Cummings 等人（2025）所需的空间。此外，相关问题也已得到广泛研究，例如求和与直方图估计（Henzinger 等人，2024；2023）、矩估计（Epasto 等人，2023）以及图统计（Jain 等人，2024）。

2. 问题设定

在介绍主要结果之前，我们先在持续发布数据流的背景下引入用户级差分隐私的概念，并介绍本文为之给出空间下界的问题。

一般设定。我们研究的所有问题都共享以下持续发布设定。考虑一个由 N 名用户组成的集合，记为 u_1，……，u_N。这些用户向一个由 T 个步骤 x_1，……，x_T 组成的数据流 x 作出贡献。在数据流的每一步 t，要么有一个空更新（$x_t = \perp$），要么该步骤由一名用户 $u_{i(t)}$ 提交一个特定于问题的更新 $j(t)$（记为 $x_t = (u_{i(t)}, j(t))$）。接收到更新 x_t 后，算法以输出 a_t 作出响应，该输出力求逼近步骤 t 的真实答案 a_t^*。这些问题（定义见下文）在具体输入和真实答案上有所不同。

准确性要求。对于所有问题，我们都希望问题 a_t 的答案在所有步骤 t，$a_t \in (1 \pm \tau)a_t^* \pm \eta$ 中，均为真实答案 a_t^* 的 (τ, η)-近似。为简化记号，我们令 $\tau = .1$，并仅关注加性误差 η。

隐私约束。对于所有问题，我们都希望设计用户级差分隐私算法。在用户级差分隐私中，如果一条流可以通过将另一条流中某个用户 u 的所有更新替换为 $\perp$ 而得到，则这两条流 x，x' 被视为相邻；随后，我们要求算法在 x 和 x' 上、跨越所有 T 步骤的输出分布满足 (ϵ, δ)-不可区分性。形式化地，记 $A(x)$（分别为 $A(x')$）为算法在输入流 x（分别为 x'）上的响应所组成的长度为 T 的向量。对于结果向量上的任意事件 S，(ϵ, δ)-用户级差分隐私要求：

$$\Pr[A(x) \in S] \leq \Pr[A(x') \in S] \cdot e^{\epsilon} + \delta.$$

不同元素计数 问题。现在我们可以正式定义第一个问题。对于 CountDistinct 问题，数据流 x_t 的更新要么为空（$x_t = \perp$），要么由一位用户 $u_{i(t)}$ 提交一个“+1”或“−1”（记作 $x_t = (u_{i(t)}, \pm 1)$）。在任意时间步 $t \in [T]$，我们可以定义频率向量 $f^t \in \mathbb{R}^N$，其中 f_i^t 表示用户 u_i 截至时间 t 提交的更新总和（假设没有更新的用户，其总和为 0）。我们的下界对于一个限制更严格的问题版本同样成立：对所有用户 u_i 和所有时间 t，频率 $f_i^t \in \{0, 1\}$ 均为偶数，因此这个下界更强。

CountDistinct问题要求算法在每个步骤 t之后输出一个值 a_t，以近似表示 $a_t^* = \|f^t\|_0$（即截至时间 t总和非零的用户数量）。

请注意，与以往工作（Cummings 等人，2025年）一样，这种表述允许估计值同时存在加性误差和乘性误差。鉴于针对差分隐私算法已知的强信息论下界（例如，参见（Jain 等人，2023b；a）），加性误差是必要的。即使在非隐私设定下，要为此问题设计空间高效的流式算法（例如 HyperLogLog、MinHash 等），也已知必须允许乘性误差（Alon 等人，1996；Kane 等人，2010）。请注意，这种表述自然地适用于例如统计任意时刻当前活跃用户数的情形（其中，更新表示用户激活，而 $+1$ -1 更新表示用户在某个时刻停用）。

The MaxSelect 问题。在 MaxSelect 问题中（Jain

等人，2023b），有一个公开已知的 d 个特征。每位用户的输入是一个二进制向量 $v_u \in \{0,1\}^d$，其初始值为全零向量，用户可以通过每次翻转一位来更新输入。更正式地说，在第 t 步，算法会收到一次更新，该更新要么为空（$x_t = \perp$），要么由一位用户 $u_{i(t)}$提交，翻转其向量中位置 0 到 1（或反之）的比特 $j(t)$（记作 $x_t = (u_{i(t)}, j(t))$）。

在每个时间步 t，流式算法都需要在d 个选项中追踪近似最“受欢迎”的特征。更准确地说，令 $\sum_u v^t_{u,j}$ 表示项 j 在时间 t的值。用 $j^*(t)$ 表示在时间 t具有真实最大值的特征。稍微滥用一下记号，我们令 $a^*_t = \sum_u v^t_{u,j^*(t)}$ 表示真实答案。算法旨在输出一个项 a_t，使得 $\sum_u v^t_{u,a_t}$ 是 a^*_t 的 (τ,η)-近似。

MaxSelect 的一种示例实例化如下：图书爱好者平台上有 d 本书，每当用户开始（或完成）阅读一本书，就对应于该用户对该书进行一次翻转。平台希望追踪当时正在阅读的“最受欢迎”图书。

分位数问题。存在一个公开已知且有序的全集 $[U] := \{1,\dots,U\}$。每个用户的输入都是一个项 $v_u \in [U]$，且这些输入可能随数据流变化。具体而言，在时间步 t，算法会收到一次更新，该更新要么为空（$x_t = \perp$），要么由某个用户$u_{i(t)}$ 将其项更改为 $v(t)$（即 $x_t = (u_{i(t)}, v(t))$）。

此外，在每个时间步，流式算法都需要回答一个分位数查询：也就是说，给定一个秩参数 $0 \le k \le N$，算法需要返回一个项 $a_t \in U$，使得 a_t 相对于用户项集合的秩近似为 k。具体而言，记 $\hat{k} = |\{i \in [N] : v_{u_i} \le a_t\}$ 为 a_t 相对于 $\{v_{u_1},\dots,v_{u_N}\}$ 的秩[3]。我们要求 k 是 k 的 (τ, η)-近似。我们还指出，在此问题设定中，查询参数 k 被视为一项公开信息，算法无需保护其隐私（否则该问题将无法解决）。

我们指出，这两个问题都存在满足 $\widetilde{O}(T^{1/3})$ 加性误差的差分隐私算法（尽管需要较大的多项式空间），此外也存在空间复杂度为 $\mathrm{poly}(\log(n)\ d, \log(U))$、具有较小常数乘法误差的非隐私流式算法（例如，参见（Cormode 等人，2023；Jain 等人，2023a；Cummings 等人，2025年））。

[3]如果存在一个或多个与 a_t 相同的 v_{u_i}，则为了评估正确性，a_t 在这些 v_{u_i} 中的秩可以任意选取。

2.1. 不同元素计数：出现次数界定技术

下文我们将只关注不同元素计数问题，它是我们下界技术的主要示例。为了深入了解不同元素计数问题的困难性，我们将首先回顾以往工作中为该问题设计多项式空间算法的方法。从这个问题中获得的直觉将有助于理解所有其他情形下的下界。

对于这个问题，若不对用户对数据流的贡献作任何限制，并采用用户级差分隐私，已知（Jain 等人，2023a）该问题的极小极大（加性）误差率按 $\widetilde{\Theta}(T^{1/3})$ 的量级变化（对于任意小的常数乘法近似）。先前的工作（Cummings 等人，2025年）能够在任意数据流上以高概率达到这一误差率，所需空间为 $\widetilde{\Theta}(T^{1/3})$。此外，（Cummings 等人，2025年）针对该问题的一个承诺变体设计了一种改进算法，该变体对用户对数据流的贡献设定了上限。对此，我们将通过出现次数这一概念加以形式化，接下来将对此进行讨论。（另请参见Jain 等人(2023a) 中与之相关的翻转率概念。）

数据流中的出现次数。用户出现次数定义为用户 u_i 在数据流中出现的更新次数。若数据流中最多有（w，k）出现次数有界，则该数据流中更新次数严格大于 k 的用户不超过 w 个。这一定义受到实践中众所周知的长尾用户分布的启发：众所周知，少数用户的活跃程度远高于绝大多数用户（Mahanti 等人，2013；Clauset 等人，2009年）。

先前的工作（Jain 等人，2023a；Cummings 等人，2025年）证明，如果保证数据流满足（w，0）-出现次数有界（即没有用户的 w 更新次数），则可达到 O（$w^{1/2} \cdot \mathrm{polylog}(T)$）的加性误差（且以高概率实现）。

对数据流中的出现次数进行封顶。然而，现实世界数据集很少严格满足（w,0）有界；少数用户（例如 k）的出现次数高于 w 是很常见的。解决这类情况的一种自然方法见于（Cummings 等人，2025）：首先识别更新次数超过 w 的“重度”用户，然后忽略这类用户之后的所有更新。通过将所有用户的更新次数有效封顶至 w，我们便得到一个（w,0）有界的数据流。因此，对于（w, k）-出现次数有界的数据流，假设将出现次数封顶至 w 并移除其中的 k 个重度用户，则可获得 $O(w^{1/2} \cdot \mathrm{polylog}(T)) + k$ 的加性误差。

注意，任意数据流都是 $(T^{2/3}, T^{1/3})$-出现次数有界的，因为至多有 $k = T^{1/3}$ 个用户可以进行 $w = T^{2/3}$次更新（鉴于更新总次数为 T）。基于此，Cummings 等人(2025年) 使用上述截断技术（的近似版本）来获得一个

算法，其加性误差为 $\tilde{O}(w^{1/2}+k) = \tilde{O}(T^{1/3})$，且对任意输入流所需空间为 $\tilde{O}(T^{1/3})$。

出现次数界定的空间复杂度。这种方法面临的挑战是，算法事先并不知道哪些用户属于重度用户，而且必须在运行过程中识别他们的 ID。直观来说，这需要大量内存。一个显而易见的解决方案是为每个用户维护一个计数器，以跟踪其更新次数。每当某个用户的计数器超过预先设定的阈值，算法就将该用户标记为重度用户，并在后续分析中忽略他们。更复杂的算法可以利用采样或草图方法来估算这些计数，并识别“重击者”（例如著名的 CountSketch（Charikar 等人，2002年））。尽管如此，所有已知方法似乎都需要$\Omega(k)$ 比特的工作内存——毕竟，仅仅记住已识别出的重度用户集合就至少需要$\Omega(k)$ 比特。事实上，正如Cummings 等人（2025年）所示，即使允许一定误差，要识别所有更新次数超过 w 的用户，也至少需要 $O(T/w)$ 空间。

2.2. 我们的主要结果

尽管已有工作表明出现次数界定在空间上是低效的，但这不足以为我们的问题（以及其他问题）建立无条件的空间下界。事实上，更好的不同元素计数解决方案可能完全不涉及识别重度用户。

我们的主要结果表明，上述内存开销实际上是必要的，无论是否明确采用出现次数界定技术。更正式地说，我们将证明，任何用户级差分隐私算法，只要加性误差小于$(kw)^{0.5-\Omega(1)}$，就必须使用$(kw)^{\Omega(1)}$ 比特内存，在(w,k)-出现次数有界的不同元素计数流上。这应与信息论上界$\tilde{}$(Cummings 等人，2025年)相对照，其中我们知道 $O(\sqrt{w}+k)$ 加性误差可通过多项式空间算法实现。也就是说，我们展示了不同元素计数问题在空间复杂度上一个显著的信息-计算差距。

现在可以陈述我们的主定理[4]。

定理 2.1。设 $\gamma_w, \gamma_k, \gamma_h \in (0,1)$ 满足 $\frac{1}{2}\gamma_w \le \gamma_k \le \gamma_h \le \gamma_w$ 和 $3\gamma_h < 1$，且 $T \ge 1$ 足够大。设 $w = T^{\gamma_w}, k = T^{\gamma_k}, h = T^{\gamma_h}$。假设 A 是一个$(\varepsilon = 0.5, \delta = \frac{1}{T^2})$-差分隐私算法，用于处理$(w,k)$-出现次数有界数据流上的流式 *Count-Distinct* 问题，并具有如下准确性保证：对于任意固定数据流x，以 $1-\frac{1}{T^2}$的概率，$A(x)$ 的响应在整个数据流上都是对真实答案的$(\tau = \frac{1}{10}, \eta = h)$-近似。那么，$A$ 至少需要使用

$$M \ge \tilde{\Omega}\left(\frac{kw}{h^2}\right) = \tilde{\Omega}\left(T^{\gamma_w+\gamma_k-2\gamma_h}\right)$$ 比特。

我们将这一结果解读为如下下界：使用$k^{o(1)}$ 比特内存时，所能达到的最佳误差与最佳上界之间必然相差一个多项式因子。例如，考虑 $\sqrt{w} \approx k$的情形。已知的差分隐私算法Cummings 等人 (2025年) 在这一情形下，以加性误差 $O((\sqrt{w}+k)\cdot \mathrm{polylog}(T)) = O(k)$ 解决该问题，所需空间为 $O(k)$。根据定理2.1，我们知道，任何达到误差$h \le \sqrt{kw} \approx k^{3/2}$ 的算法，都至少需要 $k^{\Omega(1)}$ 的空间。

定理 2.1 的一个值得注意的推论如下：令$\alpha > 0$ 足够小。通过设置$\gamma_w = \frac{2}{3} - 4\alpha, \gamma_k = \frac{1}{3} - 2\alpha, \gamma_h = \frac{1}{3} - \alpha$，以及，我们得到一个量级为 $\tilde{\Omega}(T^{1/3-4\alpha})$的下界。形式化地说：

推论 2.2. 令 $\alpha \in (0, \frac{1}{9})$。对于在$(w = T^{2/3-4\alpha}, k = T^{1/3-2\alpha})$出现次数有界数据流上计算不同元素计数的任意$(\varepsilon = 0.5, \delta = \frac{1}{T^2})$-差分隐私算法，若其以概率$1 - \frac{1}{T^2}$提供$(\tau = \frac{1}{10}, \eta = T^{1/3-\alpha})$-近似保证，则必须使用 $\tilde{\Omega}(T^{1/3-4\alpha})$ 比特的空间。

可以使用Cummings 等人，2025年）中的算法，在不同元素计数 上处理$(w = T^{2/3-4\alpha}, k = T^{1/3-2\alpha})$-$\tilde{}$出现次数有界、误差为 $O(\sqrt{w}+k) \approx T^{1/3-2\alpha}$数据流。（Cummings 等人，2025年）中的算法使用 $T^{1/3+O(\alpha)}$ 比特空间，而定理2.2 指出，任何准确度优于 $T^{1/3-\alpha}$ 的算法都必须使用$T^{1/3-4\alpha}$ 比特空间，这表明（Cummings 等人，2025年）中的算法在这一情形下几乎是紧的。

除了选择 h、k、w之外，我们在定理2.1 中得到的权衡不一定是紧的。在任何出现次数有界的数据流上，找到内存与准确度之间的紧致权衡，仍是一个有趣的开放问题。

扩展到更多流式问题。我们针对 MaxSelect 和分位数问题给出了类似的结果。

定理 *2.3*。设 $T \ge 1$、$w = T^{\gamma_w}$、$k = T^{\gamma_k}$、$h = T^{\gamma_h}$ 和 $\varepsilon = 0.5$、$\delta = \frac{1}{T^2}$的设置方式与定理 *2.1* 中相同。假设 A 是一个 *(ε, δ)*-差分隐私算法，对于 *(w, k)*-出现次数有界的数据流，它以概率 $1-\frac{1}{T^2}$ 为 *MaxSelect*（分别为分位数）实现 *(τ = 1/10, η = h)*-近似保证。如果 $d \ge 2$ 对于 *MaxSelect* 成立（分别地，如果 $U \ge 2$ 对于分位数成立），那么算法 A 必须使用至少 $\Omega(T^{\gamma_w+\gamma_k-2\gamma_h})$ 比特的空间。

我们注意到，在定理 2.3 中，只需问题具有轻微的非平凡性（即，$d \ge 2$ 对于MaxSelect 以及$U \ge 2$对于分位数），就能得到我们的下界。

本节其余部分将重点介绍定理证明中使用的困难实例。

[4]为便于阐述，我们将隐私参数固定为一种常见且标准的取值 $(\varepsilon = 0.5, \delta = \frac{1}{T^2})$，并不关注我们的结果如何随 ε, δ 变化。

2.3. 困难实例的构造

我们在此为 CountDistinct 问题设计困难实例，并指出，只需进行最小修改，类似思想也适用于另外两个示例问题。设 w, k, h 满足 $\sqrt{w} \le k \le h \le T^{1/3}$。我们的目标是构造一组 (w, k)-出现次数有界的输入流，使得任何在这些数据流上解决 CountDistinct 估计问题的低空间差分隐私算法都必须产生较大的误差。

我们将考虑以下困难实例的构造。设 $[N]$ 为用户全集，并从中均匀随机选取一个大小为 k 的子集 $C \subseteq_{T} [N]$。C 中的元素将被视为重度用户集合。设 $P = \frac{T}{2w(3h+k)}$。根据我们对 w、k 和 h 的选择，我们得到 $P \gg \frac{h^2}{w}$，这是分析中需要满足的一个关键条件。

设 S_1、……、S_P 是 P两两不相交的 $[N]\backslash C$ 子集的集合，每个子集大小为 $3h$，且均为随机抽取。

数据流。我们将构造一个长度为 $T = 2P \cdot w(3h + k)$ 的数据流。具体而言，对于每个阶段 $i = 1$、……、P以及每次重复 $j = 1$、……、w，我们执行以下操作：

1. 自然采样 $p(i, j) \sim [0, 1]$。然后，对于每个 $u \in S_i \cup C$，自然独立地采样一个比特 $b(u, i, j) \sim \text{Ber}(p(i, j))$（即，$b(u, i, j) = 1$ 以$p(i, j)$的概率取 1，否则取 0）。

2. 考虑来自 $S_i \cup C$ 的所有用户。按照 ID 递增的顺序处理这些用户：对于每个 $u$such that$b(u, i, j) = 1$的用户，插入一次更新 $(u, +1)$。对于每个 $u$such that $b(u, i, j) = 0$，插入一次更新 $\perp$。

3. 完成所有插入后，查询算法 A 以估计当前的不同元素计数，令 $r(i, j)$ 为算法的输出，我们假设该输出已截断至范围 $[0, 3h + k]$。我们还将 $a(i, j) = r_{(i,j)}/(3h+k) \in [0, 1]$ 称为归一化输出计数。

4. 按照插入的相同顺序，撤销（即删除）步骤（2）中发生的所有插入。

直观解释。这里我们解释为什么这种构造能够抵抗低空间的差分隐私算法。本质上，我们可以将该构造看作由 P 个阶段组成，在每个阶段中，数据流会对来自 $C \cup S_i$ 的用户提出 w 个随机单向边际查询：也就是说，对于 $i \in [P]$ 和 $j \in [w]$，假设我们为每个用户 $u \in C \cup S_i$ 创建一个属性。构造的步骤（1）意味着，我们以概率 $p(i, j)$ 将该属性设为“一”，并且对每个$u \in C \cup S_i$独立进行。步骤（2）插入所有具有“一”属性的用户。随后，步骤（3）查询当前的不同元素计数，这恰好对应于具有“一”属性的用户数量。最后，步骤（4）撤销我们在步骤（2）中执行的更新，以便算法为下一次单向边际查询做好准备。

不难验证，我们的构造确实生成了一个 $(2w, k)$-出现次数有界的数据流：除 C 中的用户（共有 k 个）外，每个其他用户 $u \in S_i$ 至多参与 w 个单向边际查询，其中每个查询要求他们最多发送两次更新（一次插入和一次删除）。

现在，为了以至多 h 的加性误差回答查询，例如可以直接使用拉普拉斯噪声添加机制。在这里，仅统计 S_i 组中的用户就能得到几乎同样准确的估计：由此产生的额外误差至多为 $|C| \le k$。为了保护隐私，这是更可取的选择：如果算法在每个阶段都将 C 中的用户纳入考虑，C 的隐私在经历几个阶段后就会迅速受到损害。然而，为了使算法不统计 C 中的元素，算法似乎必须知道C 的身份。由于我们在实现步骤（2）时已谨慎地将 C 隐藏在 $C \cup S_i$ 之中，算法似乎唯一可行的选择就是先通过最初几个阶段学习集合 C，并将其存入内存，以便在后续分析中有意忽略这些用户的贡献。请注意，存储 C 需要大量内存，因为它是 $[N]$ 的一个随机大子集。我们的目标是证明，算法不存在其他低空间的替代方案。

3. 主要结果的证明

本节通过分析第 2.3 节中的构造，证明我们的主要结果。

我们的证明策略如下。在第 3.1 节中，我们介绍一种名为 AvoidHeavyHitters 的通信博弈，并证明其通信复杂度下界。接着，在第 3.2 节中，我们证明，任何精确的低空间差分隐私算法都能导出该博弈的低通信协议。最后，在第 3.3 节中，我们将这些结果结合起来，推导出定理 2.1。（所有省略的证明均放在附录中。）

3.1. 避免重度命中者的多方通信博弈

以下抽象通信博弈将在我们的分析中发挥关键作用。

AvoidHeavyHitters 博弈。设 w、k、h、P、N如上所述，且 M、ϵ_1、ϵ_2 为额外参数。考虑以下有 P 名参与者的通信博弈：按照第 C、S_1、……、S_P节所述生成子集 2.3。然后，第 i 名参与者将无序集合 $C \cup S_i$ 作为其输入。这些 P 名参与者与一名裁判共同进行博弈：从 $i = 1$ 开始依次进行，第 i名参与者会收到一条

来自参与者（$i-1$）的消息（如果 $i>1$），然后需要向裁判发送一个大小至少为（$\frac{1}{2}+\epsilon_1$）的集合 $Y_i \subseteq S_i \cup C$ $\cdot |S_i \cup C|$。之后，如果 $i<P$，该参与者会向下一位参与者（$i+1$）传递一条长度为 M 比特的消息。第 P 位参与者向裁判提交其子集后，博弈结束。如果没有任何 $u \in C$ 出现在超过（$\frac{1}{2}+\epsilon_2$）个P已提交的子集中，所有参与者便共同赢得博弈。

请注意，这个博弈相当抽象，并未（明确地）涉及 CountDistinct 问题。因此，我们相信，这项技术可以适用于论文所研究问题之外的多个问题。此外，还要注意，除了消息的大小（M）之外，我们没有对协议的资源（内存、时间）施加任何限制。

我们注意到，如果 $\epsilon_1=\epsilon_2$，那么这里的一种简单策略是，对于每个参与者 i，直接从 $S_i \cup C$中随机选择一个大小为 $(\frac{1}{2}+\epsilon_1)\cdot |S_i \cup C|$的子集 Y_i。这种基线协议完全不需要通信。直观地说，对于更大的 $\epsilon_2>\epsilon_1$，由于这种策略会失效，博弈变得更具挑战性，每个参与者都必须传递一些信息，以帮助下一位参与者（部分地）识别 C。通过仔细的信息论论证，我们证实了这一直觉，并证明通信量必须增长为 $\Omega\left(k(\epsilon_1-\epsilon_2)^2\right)$:

引理3.1。 考虑有P名参与者、参数为 h、$k \geq 1$ 和 ϵ_1、$\epsilon_2 \in (0,1)$ 的避免重度命中者博弈，其中 $|S_i|=3h$、$|C|=k$。那么，为使参与者以概率 $1-\frac{1}{10P}$ 赢得博弈，通信复杂度必须满足

$$M \geq \log\binom{3h+k}{k} - \log\binom{(3h+k)(\frac{1}{2}+\epsilon_1)}{(\frac{1}{2}-\epsilon_2)k} - \log\binom{(3h+k)(\frac{1}{2}-\epsilon_1)}{(\frac{1}{2}+\epsilon_2)k} - O\left(\log(h+k)\right).$$

当 $h \geq k$和 ϵ_1、$\epsilon_2 \in (\frac{1}{k}$、$1-\frac{1}{10})$时，我们可以使用斯特林近似得到

$$M \geq \Omega\left(k(\epsilon_1-\epsilon_2)^2\right) - O(\log(h+k)).$$

3.2. 从差分隐私流算法到通信协议

在为避免重度命中者博弈建立了通信下界之后，我们将其与用于不同元素计数的差分隐私流算法联系起来。

回顾第2.3节，$r(i, j)$是算法产生的答案。下面的主要引理展示了如何将不同元素计数的准确答案转换为集合Y_i，使其满足避免重度命中者博弈的约束：

引理3.2。 假设算法 A 满足定理 *2.1*中的准确性保证。那么，存在一个过程 舍入，其输入为 $r(i,1),\ldots,r(i,w)$ $S \cup C_i$，并生成 Y_i，使得以下结论以概率 $1-\frac{1}{10P^2}$ 成立：$|Y_i| \geq \left(\frac{1}{2}+\epsilon_1\right)\cdot|S_i \cup C|$，并且不存在出现于超过 $\left(\frac{1}{2}+\epsilon_2\right)P$ 个已提交内容中的 $u \in C$。

subsets for $\epsilon_1=\Theta\left(\frac{\sqrt{w}}{h\sqrt{\log(Ph)}}\right), \epsilon_2=O\left(\frac{1}{\sqrt{P}}\right)$.

在介绍这一“舍入”过程之前，我们先指出，只需让参与者模拟算法在数据流某一阶段上的运行，就可以轻松地将其转化为通信协议，具体如下所述。

推论 3.3。 设参数与定理 *3.2*中的相同。假设算法 A 满足定理 *2.1*中的准确性保证，且令 M 表示其空间复杂度。那么，存在一个通信复杂度为 M的协议，以概率 $1-\frac{1}{10P}$ 赢得 *AvoidHeavyHitters*。

证明。 对于第 i 位参与者，协议的工作方式如下：

- 参与者接收算法的内存配置，该配置来自第 $(i-1)$ 位参与者（若 $i>1$），并从裁判处接收集合 $S_i \cup C$。
- 随后，参与者构造数据流的 i阶段，如第 2.3 节所述。重要的是，参与者只需知道 $S_i \cup C$便能构造输入，而无需告知 S_i 除 C之外的任何信息。
- 随后，参与者将更新输入给 A，并从中收集响应 $r(i,1),\ldots,r(i,w)$。之后，参与者使用舍入计算 $Y_i \subseteq S_i \cup C$。
- 如果 $|Y_i| < \left(\frac{1}{2}+\epsilon_1\right)|S_i \cup C|$，参与者将中止协议（并输掉博弈）。否则，参与者向裁判发送 Y_i，并将内存配置传递给下一位参与者（如果 $i<P$）。

根据定理3.2以及对所有阶段应用并集界，该协议以概率 $1-\frac{1}{10P}$赢得博弈。我们指出，在归约中，我们仅使用的空间界来限制参与者发送和接收的消息长度。特别是，为了构造查询、收集和处理响应，参与者可以使用任意大的工作内存和时间。

□

3.2.1. 相关性分析与舍入算法

本小节余下部分将介绍并分析舍入。

指纹引理。 舍入所需的主要工具是著名的指纹引理：

引理 3.4（例如，参见 Bun 等人（2018）；Dwork 等人（2015年））。 设 $f: \{0,1\}^n \to [0,1]$ 是一个函数。假设

对每个 $x \in \{0,1\}^n$，都有 $|f(x) - \frac{1}{n}\sum_i x_i| \le \frac{2}{5}$。那么，有

$$\mathop{\mathbb{E}}_{p\in[0,1]} \mathop{\mathbb{E}}_{x\sim \mathrm{Ber}(p)^n} \left[f(x) \sum_{1\le i\le n} (x_i - p) \right] \ge \frac{1}{10}. \quad (1)$$

这个简单却有力的引理通常可以这样理解。设想 f 是一个均值估计器：它接收 n 个输入比特，并估计它们的均值。定理 3.4 的假设 f 应当是一个精度尚可的估计器。另一方面，该引理的结论意味着，任何这样的估计器都必须与其输入具有显著较高的相关性。

度量相关性。设 $\beta = \frac{1}{T^2}$ 是一个用于分析的小参数。考察第 2.3 节中的构造，令 $u \in C$ 为一个重度用户。考虑以下思想实验：我们以与抽取 $b'(u,i,j)$ 相同的方式，独立地抽取 $b(u,i,j)$。那么，以下量

$$\xi' = \sum_{i\in[P], j\in[w]} a(i,j)\cdot(b'(u,i,j) - p(i,j)) \quad (2)$$

可以理解为 $P\cdot w$ 个有界（于 $[-1,1]$ 内）、均值为零且相互独立的随机变量（随机性来自 $p(i,j)$ $a(i,j)$ $b'(u,i,j)$）之和。因此，很容易看出它是 $(\sqrt{Pw})$-次高斯的[5]，并且有 $\Pr[\xi' \le \sqrt{Pw\log(2/\beta)}] \ge 1-\beta$。

如果该算法满足 (ϵ,δ)-差分隐私，则对于任意单个用户 u，它不应过度依赖 $b(u,i,j)$'s。也就是说，当我们将输入[6]从 $b(u,i,j)$ 改为$b'(u,i,j)$ 时，算法的输出分布不应变化太大。因此，定义随机变量 $\xi = \sum_{i,j} a(i,j)\cdot(b(u,i,j) - p(i,j))$，我们有

$$\begin{aligned}&\Pr[\xi > \sqrt{Pw\log(2/\beta)}]\\ &\le e^{\varepsilon}\Pr[\xi' > \sqrt{Pw\log(2/\beta)}] + \delta\\ &< 2\beta + \delta.\end{aligned}$$

类似地，对于任意 $i \in [P]$ 和轻用户 $u \in S_i$，量 $\sum_{j\in[w]} a(i,j)(b(u,i,j) - p(i,j))$ 以概率 $1-2\beta-\delta$ 被 $\sqrt{w\log(2/\beta)}$ 所界定。

该舍入过程。现在我们可以介绍舍入技巧：在一个阶段 $i \in [P]$ 结束后，我们可以测量该量

$$\sum_{u\in S_i\cup C} \sum_{j\in[w]} a(i,j)\cdot(b(u,i,j) - p(i,j)). \quad (3)$$

对于每个 $u \in S_i \cup C$，内层和的绝对值以概率 $1 - O(\beta+\delta)$不超过 $R = \sqrt{w\log(2/\beta)}$。我们以所有内层和均被 R界定这一事件为条件，并按如下方式采样一个随机集合 Y_i：对于每个 $u \in S_i \cup C$，以概率$\frac{R+\sum_{j\in[w]} a(i,j)\cdot(b(u,i,j)-p(i,j))}{2R} \in [0,1]$将 u 纳入 Y_i。

结果表明，通过采样过程，我们通常会生成大小为 Y_i（$|S_i\cup C|\Omega(\sqrt{w}/h)$）的集合。为了快速建立定量直觉，请注意，如果算法足够准确，我们可以利用定理 3.4 推导出，(3) 的期望下界为 $\Omega(w)$（因为式 (3) 对应于 w 个独立查询）。这意味着，采样子集的大小与 1/2 相差 $R\cdot\frac{w}{(3h+k)} \approx \frac{\sqrt{w}}{h}$。另一方面，根据算法的差分隐私性质，对于每个重度用户 $u \in C$，其获得的总相关性分数（如式 (2) 所示）至多为 $\sqrt{Pw}\log(2/\beta)$。因此，在期望意义下，该用户出现在至多 $\frac{1}{2}P + \frac{\sqrt{Pw}}{R}\log(2/\beta) = \frac{1}{2}P + O(\sqrt{P})$ 次提交中。关于如何将期望计算转化为高概率结论，请参见附录 A.2 节。

3.3. 定理2.1

定理2.1 由定理3.1和3.3可得。回忆一下$\epsilon_1 = \Theta\left(\frac{\sqrt{w}}{h\sqrt{\log(Ph)}}\right)$，$\epsilon_2 = O\left(\frac{1}{\sqrt{P}}\right)$来自定理3.3。对于定理2.1中的参数，我们有

$$\frac{\epsilon_1}{\epsilon_2} \ge \Omega\left(\frac{\sqrt{w}\cdot\sqrt{T}}{h\cdot\sqrt{\log T}\cdot\sqrt{w(3h+k)}}\right) \ge 2.$$

因此，根据定理3.1，我们得到下界

$$M \ge \widetilde{\Omega}(k(\epsilon_1-\epsilon_2)^2) = \widetilde{\Omega}\left(\frac{kw}{h^2}\right).$$

4. 结论

在这项工作中，我们首次为用户级差分隐私数据流算法建立了无条件空间下界。这解决了一个近期的开放问题：我们证明，现有差分隐私算法所需的多项式空间基本上已是最优的（在不同元素计数的某些范围内），并且与非隐私算法相比存在指数级分离。我们的证明依赖于一种新的通信博弈，该博弈表明，在许多情况下，跟踪“过度活跃”用户这一内存密集型任务对于实现隐私而言从根本上不可或缺。此外，我们证明了这一框架可推广到 MaxSelect 和分位数估计等问题，表明高内存开销是广泛一类隐私连续监测任务所固有的。我们认为，这种证明技巧可以应用于许多迄今仍无法通过低空间隐私算法解决的问题。

[5]定义和关于次高斯随机变量的性质，参见 附录 A.1 节 。

[6]严格来说，$b(u,i,j)$并不直接构成输入流。然而，从构造中可以看出，这些输入是直接由 $b(u,i,j)$构造而成的。因此，我们将 b 视为该算法的“真正隐藏输入”。

影响声明

本文旨在推动机器学习领域的发展。我们的工作可能会对社会产生许多影响，但我们认为其中没有任何影响需要在此特别指出。

参考文献

Aamand, A., Chen, J. Y., 和 Silwal, S. 规避隐私转弯stile流中的加性误差障碍。载于国际学习表征会议 *(ICLR)*，2026年。

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., 和 Zhang, L. 差分隐私下的深度学习。载于*CCS*，第308–318页，2016年。

Acharya, J., Sun, Z., 和 Zhang, H. 差分隐私Assouad、Fano 和 Le Cam。载于*ALT*，第48–78页，2021年。

Alon, N.、Matias, Y. 和 Szegedy, M.《频率矩近似的空间复杂度》。载于 *STOC*，第 20–29 页，1996年。

Amin, K.、Kulesza, A.、Munoz, A. 和 Vassilvtiskii, S.《界定用户贡献：差分隐私中的偏差-方差权衡》。载于 *ICML*，第 263–271 页，2019年。

Andersson, J. D.、Jain, P. 和 Sivakumar, S.《通过矩阵分解提高完全动态流中隐私持续基数估计的准确度》。*arXiv* 预印本*arXiv:2601.02257*，2026年。

Blocki, J.、Fichtenberger, H.、Grigorescu, E. 和 Mukherjee, T.《差分隐私与亚线性时间有时并不相容》。载于 *ITCS*，第 19:1–19:18 页，2025年。

Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., 和 Taft, N. 流上的隐私衰减谓词和。见*ICDT*，第284–295页，2013。

Bun, M., Nissim, K., Stemmer, U., 和Vadhan, S. 阈值函数的差分隐私发布与学习。见*FOCS*，第634–649页，2015年。

Bun, M., Ullman, J. R., 和 Vadhan, S. P. 指纹编码与近似差分隐私的代价。*SIAM*计算杂志，47(5):1888–1938，2018。

Charikar, M., Chen, K., 和 Farach-Colton, M. 在数据流中寻找频繁项。见国际自动机、语言与编程研讨会，第693–703页，2002年。

Clauset, A.、Shalizi, C. R. 和 Newman, M. E. 经验数据中的幂律分布。*SIAM*评论，51(4)：661–703，2009年。

Cohen, E.、Lyu, X.、Nelson, J.、Sarl´os, T. 和 Stemmer, U. 持续观察和在线阈值查询下差分隐私的下界。载于*COLT*，第 1200–1222 页，2024。

Cormode, G.、Karnin, Z. S.、Liberty, E.、Thaler, J. 和 Vesel´y, P. 相对误差流式分位数。*ACM*杂志，70 (5)：30:1–30:48，2023。Cummings, R.、Epasto, A.、Mao, J.、Mukherjee, T.、Ou, T. 和 Zhong, P. 旋转门模型中用于不同元素计数的差分隐私空间高效算法。载于*ICML*，2025年。

Dinur, I. 和 Nissim, K. 在保护隐私的同时揭示信息。见*PODS*，第202–210页，2003年。

Dinur, I.、Stemmer, U.、Woodruff, D. P. 和 Zhou, S. 关于有界空间下的差分隐私与自适应数据分析。载于欧洲密码学会议，第 35–65 页，2023。

Dwork, C.、Naor, M.、Reingold, O.、Rothblum, G. N. 和 Vadhan, S. P. 关于差分隐私数据发布的复杂性：高效算法与困难性结果。载于*STOC*，第 381–390 页，2009年。

Dwork, C.、Naor, M.、Pitassi, T. 和 Rothblum, G. N. 持续观察下的差分隐私。载于*STOC*，第 715–724 页，2010。

Dwork, C.、Smith, A. D.、Steinke, T.、Ullman, J. R. 和 Vadhan, S. P. 基于微量痕迹的鲁棒可追踪性。载于*FOCS*，第 650–669 页，2015年。

Epasto, A.、Mao, J.、Medina, A. M.、Mirrokni, V.、Vassil- vitskii, S. 和 Zhong, P. 流式频率矩估计的差分隐私持续发布。发表于*ITCS*，第 48–1 页，2023。

Ghazi, B.、Kumar, R.、Nelson, J. 和 Manurangsi, P. 时间窗口内不同项及出现 k 次的项的隐私计数。发表于*ITCS*，第 55:1–55:24 页，2023。

Gupta, A.、Ligett, K.、McSherry, F.、Roth, A. 和 Talwar, K. 差分隐私组合优化。发表于*SODA*，第 1106–1125 页，2010。

Hardt, M. 和 Talwar, K. 差分隐私的几何。发表于 *STOC*，第 705–714 页，2010。

Henzinger, M., Sricharan, A., 和 Steiner, T. A. 在持续观察下的差分隐私直方图、前驱和集合基数。*arXiv* 预印本 *arXiv:2306.10428*，2023。

Henzinger, M., Upadhyay, J. 和 Upadhyay, S. 持续观察下差分隐私求和的统一框架。载于*SODA*，第 995–1018 页，2024 年。

Jain, P., Kalemaj, I., Raskhodnikova, S., Sivakumar, S. 和 Smith, A. 持续观察下旋转门模型中的不同元素计数与差分隐私。*NeurIPS*，第 4610–4623 页，2023a。Jain, P., Raskhodnikova, S., Sivakumar, S. 和 Smith, A. 持续观察下差分隐私的代价。载于*ICML*，第 14654–14678 页，2023b。Jain, P., Smith, A. 和 Wagaman, C. 时间感知投影：持续观察下真正实现节点隐私的图统计。载于*S&P*，第 127–145 页，2024 年。Jordon, J., Yoon, J. 和 Van Der Schaar, M. Pate-gan：生成具有差分隐私保障的合成数据。载于*ICLR*，2018 年。Kane, D. M., Nelson, J. 和 Woodruff, D. P. 不同元素问题的最优算法。载于*PODS*，第 41–52 页，2010 年。Liu, X., Kong, W. 和 Oh, S. 高维下的差分隐私与稳健统计。载于*COLT*，第 1167–1246 页，2022 年。Lyu, X. 和 Talwar, K. 指纹编码与几何相结合：隐私查询发布和自适应数据分析的改进下界。载于*STOC*，第 2374–2385 页，2025年。

Mahanti, A.、Carlsson, N.、Mahanti, A.、Arlitt, M. 和 Williamson, C. 尾部的故事：互联网测量中的幂律。*IEEE Network*, 27(1):59–64, 2013. Muthukrishnan, S. 数据流：算法与应用。*Found. Trends Theor.Comput. Sci.*, 1(2), 2005. Muthukrishnan, S. 和 Nikolov, A. 基于差异度的最优差分隐私半空间计数。载于 *STOC*, pp. 1285–1292, 2012. Nikolov, A.、Talwar, K. 和 Zhang, L. 差分隐私的几何：稀疏情形与近似情形。载于*STOC*, pp. 351–360, 2013. Peter, N.、Tsfadia, E. 和 Ullman, J. R. 通过填充与置换指纹编码得到的差分隐私算法平滑下界。载于 *COLT*, pp. 4207–4239, 2024. Steinke, T. 和 Ullman, J. 差分隐私选择的紧下界。载于 *FOCS*, pp. 552–563, 2017. Ullman, J. R. 和 Vadhan, S. P. 概率可检验证明与生成合成数据的困难性。*J. Cryptol.*, 33(4):2078–2112, 2020. Vadhan, S. 差分隐私的复杂性。载于密码学基础教程：献给 *Oded Goldreich*, pp. 347–450. Springer, 2017.

附录

A. 第 3 节中缺失的证明3

A.1. 技术预备知识

定义 A.1 （次高斯随机变量）. 实值随机变量 X 称为 b-次高斯的，如果对每个 $t>0$都有 $\Pr[|X| > t] \le 2\exp(-t^2/b^2)$。

引理 A.2(次高斯随机变量之和)。设$X_1,\dots,X_N$是 N 个独立随机变量。若对于 $i \in [N]$，X_i是 b_i-次高斯随机变量，则 $\sum X_i$ 是 $\sqrt{\sum {b_i}^2}$-次高斯随机变量。

引理 A.3 (Azuma 不等式)。设$X_0, X_1, \dots, X_N$是一个次鞅[7],，满足$|X_i - X_{i-1}| \le B$ 几乎必然成立。那么

$$\Pr[X_N - X_0 \le -t] \le \exp\left(-\frac{t^2}{2NB^2}\right).$$

下面记录我们所用的定理A.3 的一个便捷变体。

引理 A.4。假设 $X_0, X_1, \dots, X_N$ 是一个次鞅，且 $|X_i - X_{i-1}| \le B$ 几乎必然成立，并且 $\mathbb{E}[X_{i+1} \mid X_i] \ge X_i + c$。假设 $(X'_0, X'_1, \dots, X'_N)$ 是一列随机变量，使得 $d_{\mathrm{TV}}((X_i)_{0\le i\le N}; (X'_i)_{0\le i\le N}) \le \beta$。那么

$$\Pr[X'_N - X'_0 \ge cN - t] \ge 1 - \exp\left(-\frac{t^2}{2NB^2}\right) - \beta.$$

证明。在 β 的全变差距离范围内，我们可以将X'_i替换为 X_i。为了分析 X_0、X_N，我们定义$Y_i = X_i - i\cdot c$并利用定理 A.3。 □

A.2. 舍入分析

接续第3.2节的讨论，我们现在对抽样子集的大小给出下界。回想一下，每位参与者 $i \in [P]$ 都会向流式算法提交 w 个查询。

考虑参与者 i 的第 j 次查询。我们将第 j 次查询之前的算法历史记录定义为算法与前 $i-1$ 位参与者之间的交互，以及当前参与者 i 的前 $j-1$ 次查询。

对于每个 $1 \le j \le w$，如果在给定查询 j 之前的历史记录的条件下，对于所有可能被采样的 $p(i,j) \in [0,1]$，都有以下结论成立，则称该算法对参与者 i 的第 j 次查询有用：

$$\left|\mathop{\mathbb{E}}_{b(u,i,j)\sim \mathrm{Ber}(p(i,j))}[a(i,j) \mid \text{history}] - p(i,j)\right| \le \frac{2}{5}.$$

[7]回想一下，($\mathbb{E}[X_{i+1} \mid , \dots\dots,) \ge X_i]$ $0 \le i \le N-1$ X_0 X_N 是一个次鞅，当且仅当对每个.

通俗地说，如果在算法收到第 j 个查询之前，其内部状态足以确保它能够准确回答查询，那么该算法对第 j 个查询就是有用的。关于这个定义有两点需要说明：第一，由于参与者负责定义查询并就地模拟算法，因此可以根据算法当前的内存配置判断它是否有用。第二，如果算法始终能以 $(\eta = 1 + \frac{1}{10},\ \tau = h)$-近似保证回答查询，那么该算法对每个查询都会是有用的：原因是，在我们的构造中，我们可以将 $(\eta,\ \tau)$-近似保证转换为 $1\,10(3h+k) + h$ 加性误差，这低于来自 $S_i \cup C$ 的用户比例的 $2\,5$。

我们将“对第 j 次查询有用”视为一个取决于算法历史记录的事件。根据算法的准确性保证，我们知道算法以概率 $1 - \frac{1}{T^2}$ 对每次查询都保持有用。设想一个假想世界：每当发现算法无用时，参与者就将其转换为绝对正确的均值估计器。在这个假想世界中，算法确实始终有用。此外，现实世界与假想世界之间的全变差距离至多为 $\frac{1}{T^2}$。

现在，将（3）重写为

$$\sum_{j\in[w]}\left(\sum_{u\in S_i\cup C} a(i,j)\cdot(b(u,i,j) - p(i,j))\right). \quad (4)$$

这是由 w 项组成的和，每次查询贡献一项。在假想世界中，根据定理 3.4，在给定前 $(j-1)$ 项的条件下，下一项的上界为$[-\sqrt{5h\log(1/10\beta)},\ \sqrt{5h\log(1/10\beta)}]$，其均值为 $1\,10$。随着 j 增大，我们监控（4）的部分和。这显然是一个次鞅，我们可以利用 Azuma 不等式（采用定理 A.4中的变体）推导出真实世界 中的结论：

$$\begin{aligned}
&\Pr\left[\sum_{j\in[w]}\left(\sum_{u\in S_i\cup C} a(i,j)\cdot(b(u,i,j) - p(i,j))\right) > \frac{w}{20}\right] \\
&\ge 1 - \exp\left(-\Omega\left(\frac{w^2}{w\cdot h\cdot\log(1/\beta)}\right)\right) - \frac{1}{T^2} \\
&\ge 1 - \exp(-\Omega(w/h)) - \frac{1}{T^2} \\
&\ge 1 - O(\beta).
\end{aligned}$$

因此，我们看到，方程（3）中的每个内层求和的平均值为 $\frac{w}{100h}$。根据 R 的选择，我们看到，采样过程生成的集合 $Y_i \subseteq S_i \cup C$ 的期望大小为 $|S_i \cup C|\left(\frac{1}{2} + \Theta\left(\frac{\sqrt{w}}{h\sqrt{\log(1/\beta)}}\right)\right)$。利用

集中不等式可得， Y_i 的大小以概率 $1-\exp\left(-\Omega\left(\frac{w\cdot h}{h^2\log(1/\beta)}\right)\right)\geq 1-\beta$ 至少为 $\left(\frac{1}{2}+\frac{1}{2}\Theta\left(\frac{\sqrt{w}}{h\sqrt{\log(1/\beta)}}\right)\right)$。

另一方面，根据该算法的隐私性质，对于每个 $u\in C$，界 $\sum_i\sum_j a(i,\ j)\cdot(b(u,\ i,\ j)-p(i,\ j))\leq\sqrt{Pw\log(1/\beta)}$ 以概率 $1-2\delta+\beta$ 成立。根据我们对 $R=\sqrt{w\log(2/\beta)}$ 的选择，可以推出，以概率 $1-O(\beta)$，用户 u 至多出现在 $\frac{1}{2}P+O(\sqrt{P})$ 个 Y_i 中，这是由我们对 R 的选择得到的。

根据我们对 $\beta=\frac{1}{T^2}$ 的选择，我们可以对所有集合 Y_i 应用并集界，并推断集合 Y_i 以至少 $1-\frac{1}{T}\geq 1-\frac{1}{10Ph}$ 的概率满足定理 3.2 的要求。

A.3. 定理3.1的证明

为了证明定理3.1，我们将用到下面的断言（稍后证明）。

引理 A.5。 设 $(A,m)\in\binom{[N]}{k}\times\{0,1\}^M$ 是一个联合随机变量，且 A 的边缘分布在 $[N]$的 k-子集上均匀分布。假设 $f\colon\{0,1\}^M\to\Delta\left(\binom{[N]}{k'}\right)$ 是一个随机映射，其输入为$m\in\{0,1\}^M$，输出为 $[N]$的一个 k'-子集。记 $Z=\mathbb{E}_{A,m,f(m)}[|f(m)\cap A|]$。则有

$$M\geq\log\binom{N}{k}-\log k-\log\binom{k'}{Z}-\log\binom{N-k'}{k-Z}$$

假设该断言成立，我们证明定理3.1。

定理 *3.1*的证明。 我们可以假设每位参与者总是提交一个大小恰为 $\left(\frac{1}{2}+\epsilon_1\right)|S_i\cup C|$的子集。根据我们的假设，每个 $u\in C$ 以概率 $1-\frac{1}{10P}(\frac{1}{2}+\epsilon_2)P$出现在不超过个已提交集合中。由此可知，期望而言，每个 $u\in C$ 出现在少于$(\frac{1}{2}+\epsilon_2)P+1$ 次提交中。总共有 P位参与者，因此至少存在一位参与者 i，使得该参与者提交的子集Y_i 满足 $\mathbb{E}[|Y_i\cap C|]\leq(\frac{1}{2}+\epsilon_2)k$。固定这位参与者 i^*。

我们知道，参与者 i^* 根据消息 Y_{i^*} 和底集 m_{i^*-1} 生成其子集 $S_{i^*}\cup C$。此外，集合 Y_{i^*}具有如下性质：$|Y_{i^*}|$被固定为 $(\frac{1}{2}+\epsilon_1)(3h+k)$且 $\mathbb{E}[|Y_{i^*}\cap C|]\leq(\frac{1}{2}+\epsilon_2)k$。接着，我们应用定理 A.5于 i^*，从而推导出

$$\begin{aligned}M\geq&\log\binom{3h+k}{k}-\log\binom{(3h+k)(\frac{1}{2}+\epsilon_1)}{(\frac{1}{2}-\epsilon_1)k}\\&-\log\binom{(3h+k)(\frac{1}{2}-\epsilon_1)}{(\frac{1}{2}+\epsilon_1)k}-O\left(\log(h+k)\right),\end{aligned}$$

[8]我们用 $\Delta(U)$ 表示集合 U 上的概率单纯形。正如所述。 □

下面给出定理 A.5 的证明。

定理 A.5 的证明。我们使用编码论证。首先，记作 $B=f(m)$，并考虑马尔可夫链$A\to m\to B$。现在，在给定 B 的条件下对集合 A 进行编码，其中使用一些我们将在下文引入的额外辅助信息 C。首先，令 $I:=|A\cap B|$。然后，参照 I 和 B 描述集合 $A\cap B$ 和 $A\setminus B$ （也就是说， $A\setminus B$ 表示为 $[N]\setminus B$ 的子集，而 $A\cap B$ 表示为B 的子集）。最后，将 C 定义为元组 $(I,A\cap B,A\setminus B)$。

现在考虑联合随机变量 (A,m,B,C)。我们看到， $H(B,C)\geq H(A)$，因为只给定 B 和 C 就可以准确恢复 A。根据数据处理不等式，我们还有 $H(B)\leq H(m)\leq M$。最后，我们知道 $H(B,C)=H(B)+H(C\mid B)$。综合起来，我们得到

$$M+H(C\mid B)\geq H(B)+H(C\mid B)\geq H(A)=\log\binom{N}{k}.$$

因此，只需给出 $H(C\mid B)$ 的上界。记 $I=|A\cap B|$。我们有

$$\begin{aligned}H(C\mid B)&\leq H(I\mid B)+H(A\mid B,I)\\&\leq\log(k)+\mathop{\mathbb{E}}_I\left[\log\binom{|B|}{I}+\log\binom{N-|B|}{k-I}\right]\\&\leq\log(k)+\log\binom{|B|}{Z}+\log\binom{N-|B|}{k-Z},\end{aligned}$$

其中，最后一个不等式来自二项式系数的对数凹性。将 $H(C)$ 的上界代入上述不等式，即可得出结论。 □

A.4. 定理 2.3 的证明

这里我们将说明如何将证明推广到其他问题。为了证明定理 2.3，我们将用 (Peter et al., 2024) 最近提出的“强”变体替代指纹引理（定理 3.4）。形式上，他们的引理表述如下：

引理 A.6 (Peter et al., 2024 的强指纹引理)。 设 $f\colon\{0,1\}^n\to[0,1]$ 是一个函数，满足$f(0^n)\leq 0.1$ 且 $f(1^n)\geq 0.9$。则有

$$\mathop{\mathbb{E}}_{\substack{t\sim[-\log(5n),\log(5n)]\\ p:=\frac{e^t}{e^t+1}}}\ \mathop{\mathbb{E}}_{x\sim\mathrm{Ber}(p)^n}\left[f(x)\sum_{1\leq i\leq n}(x_i-p)\right]\geq\Omega\left(\frac{1}{\log(n)}\right).\tag{5}$$

我们快速比较一下定理A.6 与定理3.4。定理A.6(之所以

被称为“强”，是因为我们对函数的要求宽松得多 f：具体而言，将其视为经验均值估计器时，我们只需要 f 在两个极端输入上准确 0^n 和 1^n。我们付出的“代价”是需要改变先验 p 的定义，而且得到的结论比定理 3.4 弱一个对数因子。不过，由于证明下界时我们关注的是多项式因子，因此在这里损失一个“log”是可以接受的。

借助定理 A.6，接下来只需从 分位数或MaxSelect流式算法中设计一个“均值估计器”。回想一下，对于 不同元素计数，这非常简单：我们只需将所有对某个查询有“贡献”的用户插入数据流。通过查询当前的不同元素计数，我们便能推断出有多少用户对该查询做出了贡献。对于新的问题，我们将改用以下构造：

- 分位数：假设在全部 $[N]$ 个用户中，有一组 Q 个活跃用户，每个活跃用户持有一个私有比特；我们希望区分他们是否全都持有 0，还是全都持有 1（如果意见不一致，我们的估计器可以输出任意结果）。我们构造一个包含两个项的全集 U，即 $U = \{0, 1\}$。然后，我们为每个活跃用户分配一个与其私有比特相对应的项。对于来自 $[N] \setminus Q$ 的用户，我们只需将其项设为 1。

  现在，考虑查询所有用户中秩为 $\frac{Q}{2}$ 的项。如果算法的加性误差小于 $\frac{Q}{2}$，我们就可以看出，如果所有活跃用户持有的都是同一个比特，算法就必须输出相同的比特。这正是定理 A.6 所要求的。

- MaxSelect：同样，假设有 Q 个活跃用户，每个用户都持有一个私有比特。为 MaxSelect问题创建两个特征（即，$d = 2$），并将第一个特征设为所有用户中一半用户的 1（随机选择这半用户，并将其视为公开信息）。然后，对于每个输入为“1”的活跃用户，将该用户的第二个特征设为 1。将非活跃用户的特征向量设为全零。

  现在我们查询两种特征中更受欢迎的一种。假设算法的加性误差小于 $\frac{Q}{2}$，我们可以看出，如果没有用户支持第二种特征，算法就必须输出特征 1。另一方面，如果所有用户都支持第二种特征，算法就需要输出特征 2。因此，这个归约也给出了一个符合定理 A.6 要求的“估计器”。

构造概述。给定这些构件，我们可以轻松修改第 2.3 节中的构造，从而将“回答单向边际查询”归约为“回答分位数或 MaxSelect 查询”。具体来说，给定参数 w、k、h、T，我们按照第 2.3 节中的相同方式生成集合 C、S_1、…、S_P。然后设计输入流：在步骤（1）中，我们根据定理 A.6 给出的规则采样p （即，设置 $n = 3h + k$，选择 $t \sim [-\log(5n), \log(5n)]$，并设置$p = \frac{e^t}{e^t+1}$）。在步骤（2）中，我们根据上述规则创建更新。在步骤（3）中，我们进行一次查询，并将其答案转换为一个比特。最后，在步骤（4）中，我们撤销步骤（2）中执行的更新。

分析概述。给定这一构造，我们采用与第3.2节和第3.1节中类似的分析方法。具体而言，我们可以定义相关项，并证明根据算法的隐私性质，这些相关项以高概率有界。同时，新的定理 A.6 指纹引理给出了这些相关项的下界。随后，我们利用相关项对若干子集进行采样，从而像第3.1节中那样，为 AvoidHeavyHitters 构造一个通信协议。最后，利用已建立的 AvoidHeavyHitters 下界，即可直接推出所需的空间下界。新分析在定量方面基本与之前相同，只有一个细节略有不同：我们从定理 A.6 得到的相关项下界，比从定理 3.4 得到的弱一个 $\log(n)$ 因子。因此，AvoidHeavyHitters 的参数 ϵ_1 缩小了 $\log(n)$ 倍，最终使我们的下界弱了 $\log^2(n)$ 倍。尽管如此，我们强调，这不会影响定理 2.3 的主要结果，因为这些对数因子可以被“吸收”到 O 表示法中。