arXiv:2603.02488v1 [cs.DS] 3 Mar 2026

时间衰减模型上的学习增强矩估计

Soham Nagawanshi* Shalini Panthangi† Chen Wang ‡
David P.Woodruff§ Samson Zhou¶

2026年3月4日

摘要

受机器学习的普及及其成功应用的推动，近期有一系列研究开始探讨流模型中的学习增强算法。这些研究表明，对于通过机器学习模型实现的自然且实用的预言机，我们可以获得空间效率更高的流式算法，而这些算法在其他情况下已被证明是不可能实现的。另一方面，当数据项具有不等权重时，例如在滑动窗口模型中，由于隐私法规的要求，较早的数据必须从数据集中删除，我们对相关问题的理解就要有限得多。在本文中，我们利用数据集重击者的预言机，为时间衰减场景中的若干基础问题提供学习增强算法，包括范数/矩估计、频率估计、级联范数和矩形矩估计。我们还通过多项实证评估补充理论结果，展示了这些算法在真实数据集和合成数据集上的实际效率。

1 引言

计算的流模型是在线学习和大规模学习算法中最基础的模型之一。在该模型中，我们考虑一个底层频率向量$\mathbf{x} \in \mathbb{R}^n$，其初始值为零向量 0^n。数据以 m 更新流的形式依次到达，其中时刻 $t \in [m]$ 的每次更新都表示为（t, σ_t）。每次 σ_t 更新都会通过增加或减少某个值，来修改频率向量中某个 $i \in [n]$ 对应的坐标 $\mathbf{x}_i$。通常，目标是使用远小于输入数据集大小的内存，计算该底层频率向量的函数 $f(\mathbf{x})$。数据流模型有着广泛的应用，例如交通监测[CWX21], 、传感器网络 [GG07], 、数据挖掘 [GZK05, AAH19],以及视频分析 [XXC12],等。数据流研究历史悠久，始于[AMS99]的开创性工作。数据流模型中研究最充分的一些问题通常与频率（矩）估计有关，即给定频率向量$\mathbf{x}$，计算其

$$F_p \qquad \| \|_p^p = \text{频率}$$

$\sum_{i=1}^n | \ _i|^p$
$\mathbf{xx}$。大量研究深入探讨了与频率估计相关的流式算法及其局限性（例如参见 [AMS99, CKS03, BYJKS04, CCF04, Woo04, CM05,

*德克萨斯农工大学。 ndsoham@gmail.com
†卡内基梅隆大学。 shalinipanthangi04@gmail.com
‡伦斯勒理工学院。 chen.wang.research@gmail.com§卡内基梅隆大学。
dpwoodru@gmail.com
¶德克萨斯农工大学。 samsonzhou@gmail.com

IW05, Li08, AKO11, KNPW11, BO13, BKSV14, BVWY18, WZ21b, WZ21a, INW22, BGL+24]及其中的参考文献）。

对于 $p \geqslant 2$，著名的计数草图框架 [CCF04, IW05] 可用于构造流式算法，以 $\widetilde{O}(n^{1-2/p}p^2/\mathrm{poly}(\varepsilon))$[1] 空间实现对 F_p 频率矩的 $(1 \pm \varepsilon)$-近似 [CCF04, IW05, AKO11]。此后，已有证明表明，该界在相差多对数因子的范围内是紧的 [CKS03, BYJKS04, Woo04, WZ21a, BGL+24]。因此，对于非常大的 p，任何流式算法本质上都需要 $\Omega(n)$ 空间。这一概念性结论相当悲观，我们自然会想知道，能否采用某种超越最坏情况分析的方法来克服空间下界。

学习增强算法。学习增强算法已成为突破最坏情况算法难度障碍的一种流行框架。这些算法利用现代机器学习模型的预测能力，获取一些额外的"提示"。学习增强算法已应用于频率估计问题 [HIKV19, JLL+20, CEI+22, ACG+25]、度量聚类 [EFS+22, HFH+25]、图算法 [BDSW24, CdG+24, DPV25, BEWZ25], 以及数据结构问题 [LLW22, FNS+25]。值得注意的是，[JLL+20] 证明，对于具有 $p \geqslant 2$ 的 F_p 频率矩问题，在存在一种自然且实用的重击者预言机时，我们可以获得空间为 $\widetilde{O}(n^{1/2-1/p}/\mathrm{poly}(\varepsilon))$ 的 $(1\pm\varepsilon)$-近似算法——若没有学习增强型预言机，这样的空间界是不可能实现的。[JLL+20] 还为相关问题改进了空间界，例如矩形 F_p 频率矩和级联范数，凸显了学习增强预言机的有效性。

时间衰减数据流。[JLL+20]及相关工作 [HIKV19, CEI+22, ACG+25]为在流式频率估计中使用学习增强预言机带来了非常有前景的启示。另一方面，这些结果几乎都只关注估计整个数据流的频率。因此，它们并未考虑数据流的时效性效应。在实践中，数据流的近期更新通常更为相关，而较早的更新可能被认为不那么重要，甚至已经失效。例如，由于流行趋势，近期的歌曲和电影在娱乐平台上通常具有更高的权重。隐私问题也是时效性效应的一个例子。为保护用户隐私，欧盟的《通用数据保护条例》（GDPR）要求在"必要"期限过后删除用户数据 [GDP16]。此外，一些互联网公司，如 [App21], [Fac21], [Goo25], 和 [Ope24]，也制定了各自的用户数据保留期限政策。

流模型中的时间衰减框架是捕捉时效性效应的理想选择。在该模型中，除了数据流之外，我们还额外给定一个函数 w，定义在 $[0,1]$ 上，用于表示过去数据流更新的重要性。具体而言，在每个时间步 t，我们都会将 w 应用于先前的时间步 $t' < t$，以有可能降低先前更新的贡献，即对于 $\tau \geqslant 1$ [2]，$w(\tau) \leqslant w(1)$。我们的目标是针对 $t \in [m]$，计算加权数据流在第 t个时间步更新后的频率函数（例如 F_p）。一般而言，适用于标准数据流的算法并不能直接推出适用于时间衰减模型的算法。因此，一个有趣的研究方向是探究学习增强型重频项预言机是否也能助力时间衰减模型中的频率估计。

[1]全文中，除非另有说明，我们使用 $\widetilde{O}(\cdot)$和 $\widetilde{\Omega}(\cdot)$来隐藏多对数项。[2]此处，步骤 t' 以 $w(t-t'+1)$作为输入。通过这种方式，w 可以定义为一个非递增函数，即$w(t-t+1) = w(1)$。

典型的时间衰减模型包括多项式衰减模型 [KP05, CTX07, CTX09, BLUZ19]，其中更新的重要性以 $1/\tau^s$ 的速率衰减（ s为某个固定常数）；以及指数衰减模型 [CS03, CKT08, CTX09, BLUZ19]，其中衰减速度关于 $1/s^\tau$ 快得多（ s为某个固定常数）。频率估计的研究常与时间衰减模型结合进行。除[KP05, BLUZ19]，等文献研究的空间界外，还有多篇论文从实践角度探讨了这一问题[XWP22, PEC21]。然而，据我们所知，一般时间衰减流模型在学习增强设定下尚未得到充分研究。

时间衰减框架的一个重要特例是滑动窗口数据流模型 [DGIM02, LT06a, LT06b, BO07, CMS13, BLLM15, BLLM16, BWZ21, BGL$^+$18, BDM$^+$20, WZ21b, BEL$^+$20, EMMZ22, JWZ22, BLMZ23, WY23, WZZ23, CJY$^+$25, BGW$^+$26]。在此模型中，给定窗口大小 W，时间衰减函数变为二元函数： $w(t') = 1$ 若 t' 位于大小为 W 的窗口内（即 $t' \geqslant t - W + 1$），则取该值；否则取 $w(t') = 0$。针对这一特殊应用， [SSM24] 研究了适用于滑动窗口数据流的学习增强型窗口紧凑空间节省算法（WCSS算法）。尽管这项开创性工作在实证性能方面表现出色，但 [SSM24] 中的WCSS算法存在两个问题： i)。它没有对空间复杂度给出任何形式化保证；以及 ii)。出于技术原因，该论文偏离了 [JLL$^+$20]中的重击者预言机，转而使用一种不那么自然、且可能更难实现的“下次出现”预言机。具体而言，流式算法现有下界中的困难实例涉及识别L_p 重击者并近似其对 F_p 矩 [WZ21a]的贡献。由于 [SSM24] 的“下次出现”预言机并不执行这项任务，因此尚不清楚他们的方法如何用于改进标准的流式处理和草图技术。此外，我们也不清楚他们的算法如何扩展到本文研究的一般时间衰减模型。因此，若能为一般时间衰减算法取得成果，就意味着滑动窗口模型也能取得改进的成果，从而使这一开放问题更具吸引力。

我们的结果。本文通过为时间衰减模型（分别地，滑动窗口模型）中的 F_p 频率估计问题及相关问题设计近乎最优的算法，对这一开放问题给出了肯定的回答。我们的主要结果总结如下（所有界限均适用于多项式衰减、指数衰减和滑动窗口设置）。

- F_p 频率：我们提出一种学习增强算法，该算法给定重击者预言机和数据流中的频率向量$\mathbf{x}$ 后，可在 $\widetilde{O}(\frac{n^{1/2-1/p}}{\varepsilon^{4+p}} \cdot p^{1+p})$空间内计算 F_p 频率$\|\mathbf{x}\|_p^p = \sum_{i=1}^n |\mathbf{x}_i|^p$ 的$(1 \pm \varepsilon)$-近似。

- 矩形 F_p 频率：当全集为 $[\Delta]^n$ 且数据流元素对超矩形中的所有坐标进行更新时，该数据流的频率矩问题称为矩形 F_p频率。我们提出一种学习增强算法，该算法结合重击者预言机，可在 $\widetilde{O}(\frac{\Delta^d}{(1/2-1/p)\varepsilon^{4+p}} \cdot \mathrm{poly}(\frac{p^p}{\varepsilon}, d))$空间内计算矩形 F_p频率的$(1 \pm \varepsilon)$-近似。

- (k, p)-级联范数：作为 F_p 频率矩问题的一种推广，当频率数据以 $n \times d$ 矩阵 $\mathbf{X}$ 的形式给出，且数据流每次更新一个坐标 $\mathbf{X}_{i,j}$时，我们将 $f(\mathbf{X}) = (\sum_{i=1}^n (\sum_{j=1}^d \|\mathbf{X}_{i,j}\|^p)^{k/p})^{1/k}$ 定义为(k, p)-级联范数（即各行的 p-范数的k-范数）。我们给出一种学习增强算法，可在空间$\widetilde{O}_{k,p}(n^{1-\frac{1}{k}-\frac{p}{2k}} \cdot d^{\frac{1}{2}-\frac{1}{p}})$内计算$(1 \pm \varepsilon)$-近似的$(k, p)$-级联范数。我们使用$\widetilde{O}_{k,p}(\cdot)$来隐藏多项式

关于$(kp)^{kp}$, ε和 $\log n$ 的项。 [3]

根据 [JLL+20] 中的下界，任何对 F_p 阶矩实现 $(1\pm\varepsilon)$-近似的学习增强流算法都需要 $\Omega(n^{1/2-1/p}/\varepsilon^{2/p})$ 的空间。由于流式处理场景可视为时间衰减模型的一种特殊情况，我们的 F_p 频率算法在指数 n 方面是最优的。

Task	空间界	模型	备注
F_p 频率 $(p \geqslant 2)$	$\tilde{O}(n^{1/2-1/p}/\varepsilon^4)$	流式处理	[JLL+20]
	$\Omega(n^{1/2-1/p}/\varepsilon^{2/p})$	Any	下界, [JLL+20]
	未指定	滑动窗口	[SSM24]
	$\tilde{O}(n^{1/2-1/p}\cdot p^{1+p}/\varepsilon^{4+p})$	一般时间衰减（例如，滑动窗口）	本文工作, 定理 1
矩形 F_p 频率	$\tilde{O}(\Delta^{d(1/2-1/p)}\cdot \mathrm{poly}(\frac{d}{\varepsilon}, d))$	流式处理	[JLL+20]
	$\tilde{O}(\Delta^{d(1/2-1/p)}\cdot \mathrm{poly}(\frac{p^p}{\varepsilon^p}, d))$	一般时间衰减（例如，滑动窗口）	本文工作, 定理 3
（k, p）-级联 Norm	$\tilde{O}(n^{1-\frac{1}{k}-\frac{p}{2k}}\cdot d^{\frac{1}{2}-\frac{1}{p}})$	流式处理	[JLL+20]
	$\tilde{O}_{k,p}(n^{1-\frac{1}{k}-\frac{p}{2k}}\cdot d^{\frac{1}{2}-\frac{1}{p}})$	一般时间衰减（例如，滑动窗口）	本文工作, 定理 4

表 1：结果及其与现有工作的比较总结。定理指向滑动窗口算法，以作说明。

我们的技术。我们的方法与以往学习增强滑动窗口算法（例如 [SSM24]）的工作有本质区别。具体而言，我们曾考虑过一种直接将流式算法转换为时间衰减算法的方法。值得注意的是，我们观察到，时间衰减流式处理文献中的许多方法都基于函数的平滑性（例如 [BO07、BLUZ19]）。粗略而言，这些方法遵循一个框架，在不同后缀上维护流式算法的多个副本，并删除被视为“过时”的副本。如果函数满足某些“平滑性”性质，时间衰减数据流的正确性便可得到保证。我们在这一框架下推导出若干白盒算法改编方案。我们证明，只要学习增强型预言机具有后缀兼容性，也就是说，它

[3]对于多项式和指数衰减模型，计算（k, p）级联范数需要行到达。对于滑动窗口数据流，更新可以针对点进行。

也能够预测后缀流 $[t:m]$ 中的重元素，该框架就能以与无预言机设置相同的方式应用于学习增强设置。因此，我们可以应用 [JLL+20] 中的流式学习增强算法，得到所需的时间衰减算法。

我们指出，另一种可行的选择是将 [WZ21b] 的差分估计器框架推广，以纳入建议。尽管这种方法在 ε 中能得到更好的依赖关系，但整体算法相当复杂，不太容易实现；从某种意义上说，这恰恰是最初引入机器学习建议的全部原因。因此，我们着重研究具有可证明理论保证的实用实现。

实验。对于滑动窗口数据流这一特殊情形，我们针对学习增强型 F_p 频率估计开展了实验。我们实现了多种后缀兼容重频项预言机，例如计数草图算法 [CCF04]：这使我们能够以极小的空间开销计算数据流不同后缀的重频项。我们测试了基于算法 2 的 F_p 频率算法，以及 [AMS99]（AMS 算法）和 [IW05] 的实现，并比较了使用和不使用学习增强预言机的情况。测试数据集包括从二项分布中抽样得到的合成数据集，以及 CAIDA 和 AOL 的真实互联网数据集。实验表明，学习增强方法可以显著提升频率估计算法的性能，有时甚至能得到与真实值极为接近的结果。此外，我们的方法对于更新过程中发生的分布偏移具有较强的鲁棒性，而当分布发生变化时，缩放等其他启发式方法则会导致性能下降。 [4]

2 预备知识

时间衰减模型与滑动窗口模型。我们说明时间衰减模型及相关记号。我们假设底层数据 $\mathbf{x} \in \mathbb{Z}^n$ 是一个频率向量，其中每个坐标 $\mathbf{x}_i$表示相应项的频率。该频率向量初始化为 0^n，并且在每一时刻，向量按 (i, Δ) 更新，使得对某个 $\Delta \geqslant 0$ 有 $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta$，从而所有更新只能增大频率向量的各个坐标。本文不失一般性地假设 $\Delta = 1$。令 m 为数据流中更新的总数，我们假设其至多以全集大小 n 的某个多项式为上界。在时间衰减模型中，我们还额外给定一个权重函数 $w: \mathbb{R} \to \mathbb{R}^{\geq 0}$，时刻 $t \in [m]$ 的一次更新向坐标 $i_t \in [n]$的权重贡献权重$w(m-t+1)$。这里， w是一个满足 $w(1) = 1$ 的非递增函数，而 $s > 0$是在数据流开始之前就已固定的某个参数。我们分别有 $w(\tau) = 1/\tau^s$用于多项式衰减，以及 $w(\tau) = s^\tau$ 用于指数衰减。对所有 $i \in [n]$，第 t 个时刻的底层加权频率向量定义为 $\mathbf{x}_i^t = \sum_{t' \in [t]: i_{t'} = i} w(t - t' + 1)$。

在滑动窗口模型这一特殊情形下，我们还会给定窗口大小 W。在每一步 $t \in (W-1, m]$，我们将 $\mathbf{x}^{W,t}$ 定义为最近 W 更新步骤内的频率向量。设$f: \mathbb{R}^n \to \mathbb{R}$ 是给定函数，我们的目标是对所有 $t \in (W-1, m]$输出 $f(\mathbf{x}^{W,t})$。

除了我们旨在计算的子向量 $\mathbf{x}^{W,t}$，我们还定义$\mathbf{x}^{t_1:t_2}$，表示计入从步骤 t_1 到 t_2的所有更新后得到的向量。具体而言，$\mathbf{x}^{1:t_1}$ 和$\mathbf{x}^{t_1:m}$ 分别表示计入从数据流开始到 t_1 的更新，以及从 t_1 到数据流结束的更新后得到的频率向量。

[4] The 代码 for the 实验 are 可用 at https://github.com/ndsoham/learning-augmented-fp-time-decay

待计算的函数。我们旨在滑动窗口流模型中计算以下目标函数（定义为映射 $\mathbb{R}^n \to \mathbb{R}$）。

- F_p 频率函数： $f(\mathbf{x}) = \|\mathbf{x}\|_p^p = \sum_i |\mathbf{x}_i|^p$。
- 矩形 F_p 频率函数：这是 F_p 频率问题的一个特例，其中我们假设对于某个整数 Δ，$\mathbf{x} \in [\Delta]^n$。

此外，我们还研究高维频率的级联范数函数，即输入"频率"是一个 $n \times d$ 矩阵$\mathbf{X}$，其中每一行对应于频率的某种广义概念。（k， p）级联范数函数 f： $\mathbb{R}^{n\times d} \to \mathbb{R}$ 定义为 $f(\mathbf{X}) = \left(\sum_{i=1}^n \left(\sum_{j=1}^d |\mathbf{X}_{i,j}|^p\right)^{k/p}\right)^{1/k}$。在流模型中，在每个时间点 t，数据流中都会给出关于坐标$\mathbf{X}_{i,\ j}$ 的一次更新。我们可以类似地定义时间衰减模型和滑动窗口模型所对应的输入。

学习增强框架。 我们研究学习增强型流算法，并假设有一个预言机，可以预测$\mathbf{x}_i$ 是否为重频项。根据函数 f 的不同，我们可以通过以下多种方式定义重频项。

定义 1（重击者预言机）。 我们按照以下规则称元素$\mathbf{x}_i$ 为重频项。

(a). 如果 f 是 F_p 频率，则当 $|\mathbf{x}_i|^p \geqslant \frac{1}{\sqrt{n}} \cdot \|\mathbf{x}\|_p^p$ 时，我们称 $\mathbf{x}_i$ 为重频项。(b). 如果 f 是 F_p 中的矩形 $[\Delta]^d$频率，则当 $|\mathbf{x}_i|^p \geqslant \|\mathbf{x}\|_p^p / \Delta^{d/2}$ 时，我们称 $\mathbf{x}_i$ 为重频项。(c). 如果 f 是 (k,p)-级联范数，则当 $|\mathbf{X}_{i,j}|^p \geqslant \|\mathbf{X}\|_p^p / (d^{1/2} \cdot n^{1-p/2k})$ 时，我们称 $\mathbf{X}_{i,j}$ 为重频项，其中 $\|\mathbf{X}\|_p^p$ 是由 $\mathbf{X}$ 中各元素构成的向量的向量范数。

重频项预言机$\mathcal{O}$ 是一种学习增强型预言机，在查询 $\mathbf{x}_i$时，会回答$\mathbf{x}_i$是否满足重频项定义。若$\mathcal{O}$ 总能正确预测$\mathbf{x}_i$ 是否为重频项，我们称$\mathcal{O}$ 为确定性预言机。相反，若对于每个坐标$\mathbf{x}_i$，预言机以至少 $1-\delta$ 的概率独立地返回$\mathbf{x}_i$ 是否为重频项，我们称$\mathcal{O}$ 为成功概率为 $1-\delta$ 的随机预言机。

对于时间衰减算法，我们还需要预言机具备后缀兼容性，即能够针对所有后缀流 $[t:\ m]$, $t \in (0,\ m-1]$，返回$\mathbf{x}_i$ 是否为重频项。我们对这类预言机作如下形式化定义。

定义 2（后缀兼容重频项预言机）。 若对于数据流$[t$ 的每个后缀： $m]$，其中 $t \in (0,\ m-1]$，以及每个频率向量$\mathbf{x}(t:\ m)$，重频项预言机$\mathcal{O}$ 都能够回答$\mathbf{x}(t:\ m)_i$是否为重频项（随机情况下，回答正确的概率至少为 $1-\delta$），则称$\mathcal{O}$ 为确定性（分别为随机）后缀兼容学习增强型预言机。

关于后缀兼容重频项预言机的补充讨论。 带有重击者预言机的学习增强算法由 [JLL+20] 进行了探索。我们的设定与他们的一致，并且与 [JLL+20] 类似，此类预言机易于出于实际目的加以实现。在附录 A 中，我们提供了学习此类预言机的一般框架。

我们注意到， [SSM24] 讨论了使用布隆过滤器为每个窗口获取预测时遇到的一些困难。我们强调，后缀兼容性并不要求预测每个窗口，而只要求预测数据流的后缀（仅 $m-W+1$ 此类窗口）。与 [SSM24]讨论的问题相比，这是一个宽松得多的设定。此外，我们在第 5 节中的实验表明，只需利用一小部分流式更新，就能轻松学习后缀兼容的预言机。

3 滑动窗口模型的算法与分析

我们首先讨论滑动窗口算法这一特殊情形，因为其算法和分析清晰易懂，便于呈现。对于这一设定，我们利用了 [BO07]提出的平滑直方图框架。从较高层次来说，若函数 f 满足以下条件，则称其为平滑的。设$\mathbf{x}_A$ 和$\mathbf{x}_B$ 分别是数据流 A 和 B的频率向量，其中 B是 A的后缀。若 $f(\mathbf{x}_A)$和 $f(\mathbf{x}_B)$已经足够接近，那么在添加任意公共更新后缀后，它们仍然接近；也就是说，将 C 分别追加到 $A \cup C$ 和 $B \cup C$，得到新的频率向量$\mathbf{x}_{A\cup C}$和$\mathbf{x}_{B\cup C}$后， $f(\mathbf{x}_{A\cup C})$与 $f(\mathbf{x}_{B\cup C})$之间的差异仍然很小。[BO07] 已经证明了 F_p 频率的平滑性，我们进一步证明了矩形 F_p 频率和平滑级联范数的平滑性。

现在，我们将更详细地讨论该框架，首先介绍公共后缀增广频率向量这一概念。

定义 3 （公共后缀增广频率向量）。 设$\mathbf{x}_A$ 和$\mathbf{x}_B$ 是从带有后缀 B的数据流 A 中获得的频率向量。此外，设$\mathbf{x}_C$ 是 A 和 B的公共后缀 C 的频率向量。若$\mathbf{x}_{A\cup C} = \mathbf{x}_A + \mathbf{x}_C$ 且$\mathbf{x}_{B\cup C} = \mathbf{x}_B + \mathbf{x}_C$，则称 $\mathbf{x}_{A\cup C}$ 和$\mathbf{x}_{B\cup C}$ 是一对公共后缀增广频率向量。

换句话说，设 A， B， C分别是$\mathbf{x}_A$、$\mathbf{x}_B$和$\mathbf{x}_C$的流元素。在定义 3 中，我们有 $A \subseteq B$，而$\mathbf{x}_{A\cup C}$和$\mathbf{x}_{B\cup C}$的流元素分别为 $A \cup C$和 $B \cup C$。现在，我们可以正式定义（α， β）-光滑函数如下。

定义 4 ((α, β)**-平滑函数，** [BO07] **的定义 1)。** 函数 f： $\mathbb{R}^n \to \mathbb{R}$ 对于频率向量是 (α, β)-平滑的，如果以下性质成立。

- $f(\mathbf{x}) \geqslant 0$ 对任意频率向量 $\mathbf{x}$。
- $f(\mathbf{x}) \leqslant \text{poly}(n)$，对某个固定多项式。
- 设 $\mathbf{x}_B$ 是由$\mathbf{x}_A$的一个后缀得到的频率向量，我们有 (1) $f(\mathbf{x}_A) \geqslant f(\mathbf{x}_B)$。

(2) 对任意 $\varepsilon \in (0, 1)$，存在 $\alpha = \alpha(f, \varepsilon)$ 和 $\beta = \beta(f, \varepsilon)$，使得

  - $0 \leqslant \beta \leqslant \alpha < 1$。
  - 如果 $f(\mathbf{x}_B) \geqslant (1 - \beta) f(\mathbf{x}_A)$，则对定义 3 所规定的任意公共后缀增广频率向量$\mathbf{x}_{A\cup C}$和 $\mathbf{x}_{B\cup C}$，有 $f(\mathbf{x}_{B\cup C}) \geqslant (1 - \alpha) f(\mathbf{x}_{A\cup C})$。

根据（α， β）-平滑函数的定义，我们能够推导出以下框架，将学习增强范式下的流式算法转换为滑动窗口算法。

3.1 滑动窗口框架

[BO07] 针对（α， β）-平滑函数，将滑动窗口流式算法归约为普通的流式（近似）算法。此类归约的表述如下。

命题 1 （精确算法， [BO07]中的定理 1）。设 f 是一个（α， β）-平滑函数，且设 ALG 是一个流式算法，在数据流结束时输出 f（$\mathbf{x}$），其中$\mathbf{x}$ 是该数据流的频率向量。假设 ALG 使用 g 空间，并且每次流式更新执行 h次操作。

因此，存在一种滑动窗口流式算法 ALG′，它使用 $O(^{(g+\log n)\cdot \log_\beta n})$空间，并且每次流式更新执行 $O(^{h}\log_\beta n)$次操作，从而计算滑动窗口的$(1\pm\alpha)$近似。

命题 1 针对的是精确且确定性流式算法。事实证明，该框架的适用性要广泛得多，我们也可以使用近似且随机化流式算法获得类似结果。新的表述如下。

命题 *2*（近似算法，[*BO*07] 的定理 *2* 和 *3*）。设 f 是一个（α, β）-光滑函数，且 ALG 是一个流式算法，它输出 $f'(x)$，使得在数据流结束时，以至少 $1-\delta$ 的概率有 $(1-\varepsilon)\cdot f(x) \leqslant f'(x) \leqslant (1+\varepsilon)\cdot f(x)$，其中 x 是数据流的频率向量。假设 ALG 使用 $g(\varepsilon, \delta)$ 空间，并且每次流更新执行 $h(\varepsilon, \delta)$ 次操作。

那么，存在一种滑动窗口流式算法 ALG′，以至少 $1-\delta$ 的概率计算滑动窗口的（$1 \pm$（$\alpha+\varepsilon$））-近似，使用 $O\left(\left(g(\varepsilon, \delta') + \log_{\beta} n \quad n\right)\right) \cdot \log$ 空间，并且每次流更新执行 $O\left(h(\varepsilon, \delta') \log_{\beta} n\right)$ 次操作，其中 $\delta' = \delta\beta \log n$。

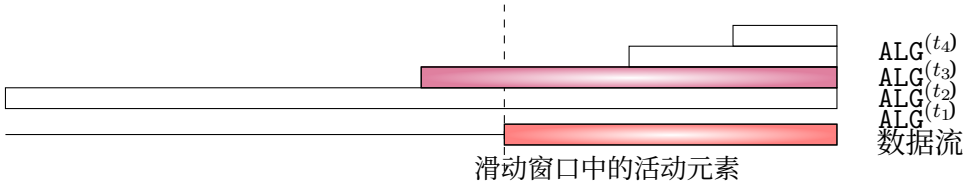


图 1：平滑直方图框架示例。此处 ALG$^{(t_2)}$ 与 ALG$^{(t_3)}$ 将活跃元素夹在中间，因此是滑动窗口的良好近似。

从高层来看，[BO07] 的算法采用了光滑直方图的思想。该算法通过以不同的起始时间运行流式算法并丢弃冗余副本来构建光滑直方图。算法 1 给出了该算法的概述，图 1 给出了示意图。

算法 1。命题 1 和命题 2 所规定框架的算法。输入：包含 m 次更新的元素流；窗口大小 W。输入：流式算法 ALG，其空间为 $g(\varepsilon, \delta)$，更新时间为 $h(\varepsilon, \delta)$。维护 ALG 的一组 $\mathcal{A}$ 存活副本

- 对于每次更新 $t \in [m]$:

  (1) 创建 ALG 的一个新副本（称之为 ALG$^{(t)}$）从第 t 次更新开始。(2) 用 (t, σ_t) 更新所有 ALG $\in \mathcal{A}$。

  (3) **剪枝：**

  (a) 从索引最小的算法 ALG$^{(\ell)} \in \mathcal{A}$ 开始，其中索引为 ℓ。(b) 当 $\ell < t-1$时：i. 找到满足 ALG$^{(k)} \geqslant (1-\beta)\cdot$ ALG$^{(\ell)}$ 的最大索引 k。ii. 剪除 $\mathcal{A}$ 中索引为 $(\ell, k-1]$ 的所有算法。iii. 令 $\ell \leftarrow k$ 并继续循环。iv. 如果 ℓ 和 t 之间没有存活副本，则跳出循环。

- 输出 ALG(t_j)，对于满足 $t_j \leqslant m - W + 1$ 的最大剩余索引 t_j

引理 3.1. 设 f 是一个(α,β)-光滑函数，并设 ALG 是任意一个具有以下性质、会查询重击者预言机 O 的学习增强流算法：

- ALG 在数据流结束时以至少 $1-\delta$ 的概率输出 $f\prime(\mathbf{x})$，使得 $(1-\varepsilon)\cdot f(\mathbf{x}) \leqslant f\prime(\mathbf{x}) \leqslant (1+\varepsilon)\cdot f(\mathbf{x})$；
- O 满足定义 2 所规定的后缀兼容性；并且
- ALG 使用 $g(\varepsilon,\delta)$ 空间，并且每次流更新执行 $h(\varepsilon,\delta)$ 操作。

那么，存在一个滑动窗口流式算法 ALG′，它以至少 $1-\delta$ 的概率计算滑动窗口的 $(1\pm(\alpha+\varepsilon))$-近似，使用 $O(\frac{(g(\varepsilon,\delta')+\log\beta^{n}\quad n}{})\cdot\log)$ 空间，并且每次流更新执行 $O(\frac{h(\varepsilon,\delta')\log\quad n}{\beta})$ 操作，其中 $\delta'=\frac{\delta\beta}{\log n}$。

证明。 命题 1 与命题 2 中归约的正确性依赖于平滑直方图对滑动窗口上的函数进行恰当近似。

设 W 为窗口大小， $t^*=m-W+1$ 为活动窗口的起始索引。目标是估计从 t^*开始的数据流后缀上的 f 。该算法维护一组活动实例 $\mathcal{A}=\{\mathtt{ALG}^{(t_1)}\ \mathtt{ALG}^{(t_2)}\ldots,\mathtt{ALG}^{(t_k)}\}$，其起始时间为 $t_1<t_2<\ldots<t_k$。

由于该算法仅通过剪枝删除冗余实例，因此在$\mathcal{A}$中存在两个相邻实例，记为 $\mathtt{ALG}^{(t_j)}$ 和 $\mathtt{ALG}^{(t_{j+1})}$，它们“夹住”真实窗口的起始时间，即 $t_j\leqslant t^*<t_{j+1}$。

设 $S^{(t)}$ 表示从时间 t开始的数据流后缀。注意到 $S^{(t_j)}\supseteq S^{(t^*)}\supseteq S^{(t_{j+1})}$。根据后缀兼容性假设，预言机 $\mathcal{O}$ 向 $\mathtt{ALG}^{(t_j)}$ 和$\mathtt{ALG}^{(t_{j+1})}$均提供有效建议。因此，这些实例正确输出值 $v_j\approx f(S^{(t_j)})$ 和 $v_{j+1}\approx f(S^{(t_{j+1})})$。

算法 1 中的剪枝条件确保，如果两个实例都保留在 $\mathcal{A}$ 中，那么 $v_{j+1}\geqslant(1-\beta)\cdot v_j$。由于 f 是（α,β）-平滑且单调的，因此条件 $f(S^{(t_{j+1})})\geqslant(1-\beta)f(S^{(t_j)})$ 与夹逼性质 $S^{(t_j)}\supseteq S^{(t^*)}\supseteq S^{(t_{j+1})}$ 相结合，意味着值 v_j 是真实窗口值 $f(S^{(t^*)})$ 的（$1\pm\alpha$）-近似。

因此，后缀兼容性确保各个实例都是正确的，而平滑直方图确保输出实例 $\mathtt{ALG}^{(t_j)}$ 是准确的近似。

最后，对于空间复杂度和操作次数，我们说明只保留 $O(\log\ {}_{\beta}{}^{n})$ 个 ALG 副本。注意，我们会删除满足 $\mathtt{ALG}^{(k)}\geqslant(1-\beta)\cdot\mathtt{ALG}^{(\ell)}$ 的副本，而更新总次数最多为 $m=\mathrm{poly}(n)$。因此，维护的副本总数至多为 $O(\log\frac{1}{1-\beta}(m))=O(\frac{\log n}{\beta})$。我们将 $\delta'=\frac{\delta\beta}{\log n}$ 进行缩放以确保成功概率，并为每个算法副本使用 $O(\log n)$ 的空间开销来维护时间戳。综合以上界限，即可得到所需的空间和操作复杂度。

□

接下来，我们将展示如何利用引理 3.1 中的框架，构造学习增强型滑动窗口算法。我们指出，该框架给出的界与窗口大小 W无关。因此，本节的界不包含参数 W 。我们针对 $p>2$ 的 F_p 估计所设计的学习增强型滑动窗口算法，在使用完美和错误预言机时，均具有以下保证。

定理 1 （学习增强型 F_p 频率矩估计算法）。存在一种滑动窗口流式算法，给定滑动窗口中的元素流以及一个固定参数

$p \geqslant 2$，以及一个确定性后缀兼容重频项预言机 O（如定义 2 所述），该算法以至少 $99/100$ 的概率输出 F_p 频率的（$1+\varepsilon$）-近似，所需空间为$O\left(\frac{n^{1/2-1/p}}{\varepsilon^{4+p}} \cdot p^{1+p} \cdot \log^4 n \cdot \log(\frac{p}{\varepsilon})\right)$。

定理 2 (学习增强型 F_p 频率算法与随机预言机)。 存在一种滑动窗口流式算法，给定滑动窗口中的元素流、固定参数 $p \geqslant 2$，以及成功概率为$1-\delta$的随机后缀兼容重频项预言机 O（如定义 2 所规定），该算法以至少 $99/100$ 的概率，在空间<中输出一个 $1+\varepsilon$近似的 F_p 频率矩估计

- $O\left(\frac{(n\delta)^{1-1/p}}{\varepsilon^{4+p}} \cdot p^{1+p} \cdot \log^4 n \cdot \log(\frac{p}{\varepsilon})\right)$ *bits if* $\delta = \Omega(1/\sqrt{n})$.
- $O\left(\frac{n^{1/2-1/p}}{\varepsilon^{4+p}} \cdot p^{1+p} \cdot \log^4 n \cdot \log(\frac{p}{\varepsilon})\right)$ *bits if* $\delta = O(1/\sqrt{n})$.

对于任意常数 p 和 ε，对于成功概率足够高的任意学习增强型预言机，滑动窗口 F_p 频率矩估计问题的空间界为 $\widetilde{O}(n^{1/2-1/p})$。即使在流式处理场景中，该界也是最优的，依据是 [JLL+20]；因此，我们得到了学习增强型 F_p 频率估计问题的近乎最优算法。

从整体上看，该算法是将 [JLL+20] 中的学习增强型 F_p 频率流式算法应用于我们在引理 3.1 中讨论的框架。 [JLL+20] 中算法的保证可描述如下。

引理 3.2 ([JLL+20])。 对于任意给定的数据流、固定参数 $p \geqslant 2$ 以及成功概率为 $1-\delta$ 的随机重频项预言机 O，以至少 $99/100$ 的概率，算法 2 使用以下空间计算出频率的 $(1+\varepsilon)$-近似

- $O\left(\frac{(n\delta)^{1-1/p}}{\varepsilon^4} \cdot \log^2 n\right)$ *bits if* $\delta = \Omega(1/\sqrt{n})$.
- $O\left(\frac{n^{1/2-1/p}}{\varepsilon^4} \cdot \log^2 n\right)$ *bits if* $\delta = O(1/\sqrt{n})$.

为了应用命题 2 的归约，我们需要了解 F_p频率函数的平滑性，这是先前研究已确立的一个标准结论。

引理 3.3 ([BO07])。 F_p 频率函数是$(\varepsilon, \varepsilon^p/p^p)$-平滑的。

定理 *1* 和定理 *2* 的证明收尾。 我们将引理 3.1 应用于引理 3.2 中的算法，并采用引理 3.3 中给出的平滑性保证。关于成功概率，我们证明，通过 $O(\log(1/\delta))$ 倍的空间开销，可以使引理 3.2 中的算法以至少 $1-\delta$ 的概率成功。这可以通过经典的中位数技巧实现：运行 $O(\log(1/\delta))$ 个流式算法副本，并取输出频率的中位数。根据切尔诺夫界，失败概率至多为 δ。

设 $\beta = \varepsilon^p/p^p$；对于确定性预言机情形，我们可以使用 $g = \frac{n^{1/2-1/p}}{\varepsilon^4} \cdot \log^2 n$ 以及失败概率 $\delta' = O(\beta/\log\ n)$，从而得到流式算法所需的空间至多为

$$g(\varepsilon, \delta') = O\left(\frac{n^{1/2-1/p}}{\varepsilon^4} \cdot \log^2 n \cdot \log(n/\beta)\right) \leqslant O\left(\frac{n^{1/2-1/p}}{\varepsilon^4} \cdot p \cdot \log^3 n \cdot \log(p/\varepsilon)\right).$$

因此，我们需要的空间为 $O\left(g(\varepsilon, \delta') \cdot \log_\beta n\right) = O\left(\frac{n^{1/2-1/p}}{\varepsilon^{4+p}} \cdot p^{1+p} \cdot \log^4 n \cdot \log(\frac{p}{\varepsilon})\right)$，正如所需。最后，对于使用随机学习增强预言机的情况，我们只需将界中的 $n^{1/2-1/p}$项替换为 $(n\delta)^{1-1/p}$，即可得到所需结论。 □

我们简要介绍算法 2 中的算法；该算法又将以下结果作为黑盒使用。

命题 3. [[AMS99,IW05,AKO11]] 存在一种随机化算法，使得在给定$\mathbf{x} \in \mathbb{R}^n$ 作为更新流时，可计算 $\|\mathbf{x}\|_p^p$ 的$(1 \pm \varepsilon)$-近似，概率至少为$99/100$，所用空间为 $O(\frac{n^{1-2/p}}{\varepsilon^{2+4/p}} \cdot \log^2 n)$ 空间。

算法 2. 用于学习增强流式处理 F_p 矩的算法。输入：以更新流给出的 $\mathbf{x}$ 输入：命题 3 中算法所用的独立副本 $\mathtt{ALG}_1$ 和 $\mathtt{ALG}_2$ 。

- 对于$\mathbf{x}_i$上的每一次元素更新：

  (1) 查询$\mathbf{x}_i$ 是否为重频项，即 $|\mathbf{x}_i|^p \geqslant \frac{1}{\sqrt{n}} \cdot \|\mathbf{x}\|_p^p$。(2) 若是，则对具有预测 $|\mathbf{x}_i|^p \geqslant \frac{1}{\sqrt{n}} \cdot \|\mathbf{x}\|_p^p$ 的项使用 $\mathtt{ALG}_1$ 。(3) 否则，按如下方式计算非重击元素的 F_p 频率。(a) 以概率 $\rho = 1/\sqrt{n}$采样$\mathbf{x}_i$ 。(b) 令 $\tilde{\mathbf{x}}$ 为从采样得到的非重击元素所获得的频率向量。(c) 使用 $\mathtt{ALG}_2$ 计算 $\tilde{\mathbf{x}}$ 的 F_p 频率，并用 ρ重新加权。_

- 汇总 $\mathtt{ALG}_1$ 和 $\mathtt{ALG}_2$ 的结果后输出。

3.2 当 $p \geqslant 2$ 时矩形 F_p 频率

现在，我们转向针对 $p \geqslant 2$ 的学习增强型矩形 F_p 频率算法。我们将确定性预言机和随机预言机的算法陈述结合如下。

定理 3. 存在一种滑动窗口流式算法，给定滑动窗口中来自 $[\Delta]^d$ 的元素流、固定参数 $p \geqslant 2$，以及成功概率为 $1-\delta$ 的随机后缀兼容重频项预言机 O（如定义 2 所规定），该算法以至少 $99/100$ 的概率输出矩形 F_p 频率的 $(1+\varepsilon)$-近似，所需空间为

- $O\left(\frac{\Delta^{d(1-1/p)} \cdot p^p \cdot \delta^{1-1/p}}{\varepsilon^{4+p}} \cdot \mathrm{poly}(\frac{p}{\varepsilon}, d, \log \Delta)\right)$ *bits if* $\delta = \Omega(1/\sqrt{n})$.
- $O(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{4+p}} \cdot p^p \cdot \mathrm{poly}(\frac{p}{\varepsilon}, d, \log \Delta)$ *bits if* $\delta = O(1/\sqrt{n})$.

此外，假设使用确定性预言机，该滑动窗口算法处理每个元素所用的时间至多为 $O(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{4+p}} \cdot p^p \cdot \mathrm{poly}(\frac{p}{\varepsilon}, d, \log \Delta))$。

证明。“此外”部分之前的定理陈述可直接由定理 2 得出。特别要注意，矩形 F_p 频率问题可以表述为 F_p 频率问题，其中$n \leqslant \Delta^d$；代入该数值即可立即得到所需的空间界。

对于处理时间，[JLL+20] 针对矩形 F_p 范数有一个 $(1 \pm \varepsilon)$-近似算法，每次更新处理时间为 $O\left(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^4} \cdot \mathrm{poly}(\frac{p}{\varepsilon}, d, \log \Delta)\right)$ 时间 ~~（不包括 p^p/ε^p 项）~~，成功概率为 $99/100$。我们可以使用中位数技巧来提高成功概率

至 $1-\delta$，空间开销为 $O(\log(1/\delta))$，且没有时间复杂度开销（我们可以并行处理算法的多个副本）。因此，应用命题 2，并采用与引理 3.3 相同的平滑性保证（矩形 F_p 是 ℓ_p 频率的子族），即可在滑动窗口模型中得到所需的$O(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{4+p}} \cdot p^p \cdot \mathrm{poly}(\frac{p}{\varepsilon}, d, \log \Delta)$处理时间。 □

3.3 $(k,\ p)$级联范数

现在我们给出学习增强滑动窗口（$k,\ p$）级联范数算法的结果。该算法的保证如下。

定理 4。存在一种滑动窗口流式算法，对于表示为坐标 $\mathbf{X}_{i,\ j}$ 的插入和删除流的 $n \times d$ 矩阵 $\mathbf{X}$，给定固定参数 $k \geqslant p \geqslant 2$ 以及一个（确定性的）后缀兼容重频项预言机 O，以至少 $99/100$ 的概率输出 (k,p)级联范数的$(1+\varepsilon)$近似，其空间为 $O(n^{1-\frac{1}{k}-\frac{p}{2k}} \cdot d^{\frac{1}{2}-\frac{1}{p}} \cdot \mathrm{poly}(\frac{1}{\varepsilon^k}, \log n))$。

对于 p、k 和 ε 的任意常数取值，我们的界在渐近意义上与学习增强流式算法的最优内存界相匹配。我们的算法利用了 [JW09] 和 [JLL+20] 的框架，并结合命题 1 和 2 中的平滑直方图框架。用于流式处理学习增强级联范数的算法相当复杂。因此，我们在算法 3 中给出了一个概要，并请感兴趣的读者参阅 [JW09] 和 [JLL+20] 了解更多详情。下文中，我们用 $F_p(\mathbf{X})$ 表示 $\mathbf{X}$ 中元素的 F_p 范数。

算法 3. 学习增强型流式处理(k,p)级联范数。输入：一个作为输入的 $n \times d$ 矩阵 $\mathbf{X}$；参数 k 和 p输入：一个重频项预言机，用于预测 $|\mathbf{X}_{i,j}|^p \geqslant \|\mathbf{X}\|_p^p/(d^{1/2} \cdot n^{1-p/2k})$

- 参数:
  - $Q = O(n^{1-1/k})$，使得 $T = (nd \cdot Q)^{1/2} = d^{1/2} \cdot n^{1-p/2k}$；
  - 层数 $\ell \in [O(\log n)]$；层 $t \in [O(\log n/\zeta)]$；$T_\ell = T/2^\ell$。
  - 用于分层和采样的参数 ζ, η, θ, B（依据 [JW09]）。
- 在数据流处理过程中应用计数草图类型的算法（例如命题 3 中的算法），以维护由**逐层预处理**步骤采样的元素。
- **按层处理**，层级为 $\ell \in [O(\log n)]$:

  （1）以概率 $1/2^\ell$对每一行进行采样；令$\mathbf{X}^{(\ell)}$ 为所得矩阵。（2）将$\mathbf{X}^{(\ell)}$ 中的元素划分到各层中：每层包含幅度属于 $[\zeta\eta^{t-1}, \zeta\eta^t]$的元素。（3）如果 $|S_t(\mathbf{X}^{(\ell)})|(\zeta\eta^t)^p \geqslant F_p(\mathbf{X}^{(\ell)})/(B\theta)$，则层 t 为贡献层，其中 $S_t(\mathbf{X}^{(\ell)})$是层 t中的元素。（4）进一步将贡献层中的元素划分为重元素和非重击元素。由此得到完全由重元素构成的贡献层，以及完全由非重击元素构成的贡献层。（5）对于每个贡献层 t:

（a）如果该层由非重击元素（轻元素）构成，且元素数量 N_t 大于$\beta_t = \theta Q|S_t(\mathbf{X}^{(\ell)})|(\zeta\eta^t)^p/F_p(\mathbf{X}^{(\ell)})$，则以速率 $\theta Q/T_\ell$进行下采样。（b）令 j 为满足 $|S_t(\mathbf{X}^{(\ell)})|/2^j \leqslant \beta_t < |S_t(\mathbf{X}^{(\ell)})|/2^{j-1}$的参数。（c）以速率 $1/2^j$ 对该层的每个元素进行采样，得到$\mathbf{Y}_t$ 作为所得矩阵。（6）使用 Count-Sketch 算法聚合所有$\mathbf{Y}_t$ 元素，得到预处理向量$\mathbf{Y}^{(\ell)}$。

- 将所有$\mathbf{Y}^{(\ell)}$ 相加，得到$\mathbf{Y}$ ，并以相同方式对$\mathbf{Y}$ 执行“ℓ_p-采样”过程~，如 [JW09] 所示，从而得到$\mathbf{Y}$.~
- 计算 $F_k(F_p(\mathbf{Y}))$作为估计值。

[JLL+20] 为带有重击者预言机的原始流式算法给出了算法 3 的保证。

引理 3.4 ([JLL+20]). 设参数 $\varepsilon > 0$ 和 $k \geqslant p \geqslant 2$ 已给定。此外，设$\mathbf{X}$ 是一个$n \times d$矩阵，其坐标$\mathbf{X}_{i,j}$以插入和删除流的形式给出。那么，算法 *3* 以至少 $99/100$ 的概率输出(k,p)-级联范数的$(1 \pm \varepsilon)$近似，使用$O(n^{1-\frac{1}{k}-\frac{p}{2k}} \cdot d^{\frac{1}{2}-\frac{1}{p}} \cdot \mathrm{poly}(\frac{1}{\varepsilon}, \log n))$空间。

接下来，我们需要界定（k,p）-级联范数的平滑性。下文中，我们将（k,p）-级联范数写成一个“范数的范数”函数，即形式为 $F_k(F_p(\mathbf{X})) := \left(\sum_{i=1}^n \left((\sum_{j=1} |\mathbf{X}_{i,j}|^p)^{1/p}\right)^k\right)^{1/k}$ for $k \geqslant p \geqslant 2$。关于这类函数平滑性的技术引理如下。

引理 3.5。 对于任意常数 $k \geqslant p \geqslant 2$，$(k,p)$-级联范数 $F_k(F_p(\mathbf{X}))$ 是$(\varepsilon, \varepsilon^k/k)$-平滑的。

证明。令 $\mathbf{X}^A$ 为通过 $\mathbf{X}^B$ 的更新后缀得到的矩阵。此外，令 $\mathbf{X}^C$ 为 $\mathbf{X}^A$ 和 $\mathbf{X}^B$ 的一个公共后缀。我们还将更新划分为 $B = A \cup (B \setminus A)$，而 $B \setminus A$ 可以是空集。进一步地，我们令 $\mathbf{X}_{i,:}$（分别为 $\mathbf{X}^A_{i,:}$、$\mathbf{X}^B_{i,:}$ 和 $\mathbf{X}^{B\setminus A}_{i,:}$）表示 $\mathbf{X}$ 的第i行向量的更新。我们将 $\left(F_k(F_p(\mathbf{X}^B))\right)^k$ 的值下界估计如下。

$$
\begin{aligned}
\left(F_k(F_p(\mathbf{X}^B))\right)^k &= \left|\sum_{i=1}^n \left(F_p(\mathbf{X}^B_{i,:})\right)^k\right| \\
&= \left|\sum_{i=1}^n \left(\left(\sum_{j=1}^d \left|\mathbf{X}^A_{i,j} + \mathbf{X}^{B\setminus A}_{i,j}\right|^p\right)^{1/p}\right)^k\right| \\
&\geqslant \left|\sum_{i=1}^n \left(\left(\sum_{j=1}^d \left|\mathbf{X}^A_{i,j}\right|^p + \sum_{j=1}^d \left|\mathbf{X}^{B\setminus A}_{i,j}\right|^p\right)^{1/p}\right)^k\right| && \text{(by s超可加性（针对 } p \geqslant 1\text{)} \\
&\geqslant \left|\sum_{i=1}^n \left(\left(\sum_{j=1}^d \left|\mathbf{X}^A_{i,j}\right|^p\right)^{k/p} + \left(\sum_{j=1}^d \left|\mathbf{X}^{B\setminus A}_{i,j}\right|^p\right)^{k/p}\right)\right| && \text{(by s对 } k \geqslant p \text{ 的超可加性)}
\end{aligned}
$$

$$= \left(F_k(F_p(\mathbf{X}^A))\right)^k + \left(F_k(F_p(\mathbf{X}^{B\setminus A}))\right)^k.$$

因此，对于任意（可能为空的）$\mathbf{X}^{B\setminus A}$，如果 $F_k(F_p(\mathbf{X}^A))$ 和 $F_k(F_p(\mathbf{X}^B))$ 是 β-接近的，也就是说 $F_k(F_p(\mathbf{X}^A)) \geqslant (1-\beta)\cdot F_k(F_p(\mathbf{X}^B))$，那么我们有

$$\begin{aligned}\left(F_k(F_p(\mathbf{X}^{B\setminus A}))\right)^k &\leqslant \left(F_k(F_p(\mathbf{X}^B))\right)^k - \left(F_k(F_p(\mathbf{X}^A))\right)^k \\ &\leqslant \left(1-(1-\beta)^k\right)\cdot\left(F_k(F_p(\mathbf{X}^B))\right)^k \\ &\leqslant k\beta\cdot\left(F_k(F_p(\mathbf{X}^B))\right)^k. \qquad \text{（根据伯努利不等式）}\end{aligned}$$

取 $1/k$ 次幂，得到 $F_k(F_p(\mathbf{X}^{B\setminus A})) \leqslant (k\beta)^{1/k}\cdot F_k(F_p(\mathbf{X}^B))$.

现在引入公共后缀 $\mathbf{X}^C$，并对 $F_k(F_p(\mathbf{X}^{A\cup C}))$ 的值作如下下界估计。根据闵可夫斯基不等式，由于 $k\geqslant p\geqslant 2$，我们有

$$F_k(F_p(\mathbf{X}^{B\cup C})) \leqslant F_k(F_p(\mathbf{X}^{A\cup C})) + F_k(F_p(\mathbf{X}^{B\setminus A})).$$

因此，由于 $\mathbf{X}^C$ 是一个非负频率矩阵，我们有 $F_k(F_p(\mathbf{X}^B)) \leqslant F_k(F_p(\mathbf{X}^{B\cup C}))$。因此，我们可以得到

$$\begin{aligned}F_k(F_p(\mathbf{X}^{A\cup C})) &\geqslant F_k(F_p(\mathbf{X}^{B\cup C})) - F_k(F_p(\mathbf{X}^{B\setminus A})) \\ &\geqslant F_k(F_p(\mathbf{X}^{B\cup C})) - (k\beta)^{1/k}\cdot F_k(F_p(\mathbf{X}^B)) \\ &\geqslant \left(1-(k\beta)^{1/k}\right)\cdot F_k(F_p(\mathbf{X}^{B\cup C})).\end{aligned}$$

令 $\beta = \varepsilon^k/k$，我们有 $(k\beta)^{1/k} = \varepsilon$，从而得到

$$F_k(F_p(\mathbf{X}^{A\cup C})) \geqslant (1-\varepsilon)\cdot F_k(F_p(\mathbf{X}^{B\cup C})),$$

这就是 $(\varepsilon, \varepsilon^k/k)$-平滑性的期望性质。 □

定理 *4* 的证明收尾。我们再次将命题 2（结合引理 3.1）应用于引理 3.4 中的算法。根据引理 3.5，$(k,\ p)$-级联范数是 $\left(\varepsilon,\ \frac{\varepsilon^k}{k}\right)$-光滑的。

采用与定理 1 的证明中相同的中位数技巧，可以证明，为确保算法 3 以至少 $1-\delta$ 的概率成功，我们只需在空间上增加 $O(\log(1/\delta))$ 倍的空间开销。因此，令 $\beta = \frac{\varepsilon^k}{k}$，即可得到

$$\begin{aligned}g(\varepsilon,\delta') &= O\left(n^{1-\frac{1}{k}-\frac{p}{2k}}\cdot d^{\frac{1}{2}-\frac{1}{p}}\cdot \mathrm{poly}\left(\frac{1}{\varepsilon},\log n\right)\cdot\log(n/\beta)\right) \\ &\leqslant O\left(n^{1-\frac{1}{k}-\frac{p}{2k}}\cdot d^{\frac{1}{2}-\frac{1}{p}}\cdot \mathrm{poly}\left(\frac{1}{\varepsilon},\log n\right)\right).\end{aligned}$$

因此，我们所需的空间为

$$O\left(g(\varepsilon,\delta'))\cdot\frac{\log n}{\beta}\right) = O\left(n^{1-\frac{1}{k}-\frac{p}{2k}}\cdot d^{\frac{1}{2}-\frac{1}{p}}\cdot \mathrm{poly}\left(\frac{1}{\varepsilon^k},\log n\right)\right).$$

由此便得到了定理陈述所要求的界。 □

4 通用时间衰减模型的算法与分析

本节将介绍我们针对一般时间衰减场景的算法。在该场景中，时间 $t' \in [m]$ 发生的一次更新会为时间 t 时坐标 $i_t \in [n]$ 的权重贡献权重 $w(t-t'+1)$。步骤 t 的底层加权频率向量 $\mathbf{x}^t$ 定义为

$$\mathbf{x}_i^t = \sum_{t' \in [t]: i_t = i} w(t-t'+1).$$

时间衰减模型及其函数的形式化定义见第 2 节。给定一个非递减函数 $G: \mathbb{R} \to \mathbb{R}^{\geqslant 0}$，且 $G(0) = 0$，我们将 G-矩估计定义为估计

$$G(\mathbf{x}) = \sum_{i \in [n]} G(\mathbf{x}_i).$$

上下文明确时，我们也使用小写的 x作为函数 G的输入。

我们提出了一个通用框架，可将用于解决 G-矩估计问题的线性草图流式算法转换为用于解决 G-矩估计问题的时间衰减算法。

定义 5（时间衰减模型中的平滑性）。给定 $\varepsilon > 0$，如果权重函数 w 和G-矩函数 G 满足以下条件，则称它们是（ε, ν, η）-平滑的：

(1) $G((1+\eta)x) - G(x) \leqslant \frac{\varepsilon}{4} \cdot G(x)$ for all $x \geqslant 1$.

（2）存在一个整数 $m_\nu \geqslant 0$，使得 $\sum_{i \in [m_\nu, m]} w(i) \leqslant \nu$ 且 $G(x+\nu) - G(x) \leqslant \frac{\varepsilon}{4} \cdot G(1)$ 对所有 $x \geqslant 1$均成立。换言之，对于长度为 m 的数据流，早于当前时间超过 m_ν个时间步到达的所有更新都可以忽略。

下一个定理为多项式衰减模型和指数衰减模型提供了一个框架。相关算法见算法 4。

定理5. 给定一个流式算法，它使用具有 k 行的线性草图，对 G-矩估计提供 $(1+\varepsilon)$ 近似；若函数 G 和 w 满足(ε, ν, η)-平滑性条件（定义 5），则存在一种适用于一般时间衰减的算法，对 G-矩估计提供$(1+\varepsilon)$近似，且所用空间不超过 $O\left(\frac{k}{\eta} \log n \log \frac{1}{\nu}\right)$ 比特。

此外，只要预言机 O 是后缀兼容的，该结论对学习增强型算法同样成立。

证明. 考虑一个固定的 $a \in [n]$ 以及所有满足 $t_{a_1}, \ldots, t_{a_r} \leqslant t$ 且对 a 有更新的时刻。于是，a 在时刻 t 的权重为 $\sum_{j \in [r]} w(t - t_{a_j} + 1)$。设线性草图分配给时刻 t 的权重为 w'。我们声称

$$\frac{1}{\sqrt{1+\eta}} \cdot \sum_{j \in [r]} w'(t - t_{a_j} + 1) - \nu \leqslant \sum_{j \in [r]} w'(t - t_{a_j} + 1) \leqslant \sum_{j \in [r]} w(t - t_{a_j} + 1).$$

考虑一个固定的块 B_i。首先，根据η 的定义以及构造方式，每个块中所有索引的权重之差至多为乘性因子 $\sqrt{1+\eta}$。块 B_i中的所有元素都被赋予权重 w_i'，使其为 B_i中最新元素权重的 $\frac{1}{\sqrt{1+\eta}}$ 倍。因此，对于块 B_i中的任意更新 t_{a_j}到 a，我们有$\sqrt{1+\eta} \cdot w(t - t_{a_j} + 1) \leqslant w_i' \leqslant w(t - t_{a_j} + 1)$。最后，对于

任何位于没有草图的块中的更新 t_{a_j}，都必须满足 $t - t_{a_j} + 1 \geqslant n_\nu$。根据定义，所有此类更新的权重至多为 ν。因此，我们有

$$\frac{1}{\sqrt{1+\eta}} \cdot \sum_{j\in[r]} w'(t - t_{a_j} + 1) - \nu \leqslant \sum_{j\in[r]} w'(t - t_{a_j} + 1) \leqslant \sum_{j\in[r]} w(t - t_{a_j} + 1),$$

如所需。

$\widehat{G(x_i)}$ 是通过此过程隐式赋予坐标 $i \in [n]$ 的权重。根据 、η 和 ν 的定义，可得 $(1 - \frac{\varepsilon}{4}) \cdot G(x_i) \leqslant G(x_i) \leqslant G(x_i)$。对所有 $i \in [n]$ 求和，得到

$$\sum_{i\in[n]} \left(1 - \frac{\varepsilon}{4}\right) \cdot \sum_{i\in[n]} G(x_i) \leqslant \widehat{G(x_i)} \leqslant \sum_{i\in[n]} G(x_i).$$

因此，只需获得频率向量按 w' 加权后的 G 阶矩的 $(1 + \frac{\varepsilon}{4})$-近似。由于 $\mathbf{A}$ 是一个线性草图，且 $w'_i \cdot \mathbf{v}_i$ 恰好是块 B_i 按 w' 加权后的频率向量，因此这正是后处理函数 f 所实现的功能。因此，该算法是正确的。

剩下需要分析空间复杂度。每个线性草图 $\mathbf{A}\cdot\mathbf{v}_i$ 使用 $O(k \cdot \log n)$ 比特空间，假设所有权重和频率都可以用 $O(\log n)$ 比特空间表示。针对特定函数 $w(\cdot)$ 和 $G(\cdot)$，还可以进一步优化；我们将在具体应用中这样做。一般而言，注意到我们至多维护三个块，其中的权重相差不超过乘性因子 $(1+\eta)$。块中索引的最小权重至少为 $\frac{\nu}{1+\eta} \geqslant \frac{\nu}{2}$，而根据假设，$w(1) = 1$。因此，由于 w 单调不增，块的数量至多为 $3\log_{(1+\eta)} \frac{2}{\nu}$。故该算法至多使用 $O\left(\frac{k}{\eta}\log n \log \frac{1}{\nu}\right)$ 比特空间。

最后，关于后缀兼容预言机的陈述中“此外”部分，可由与我们在引理 3.1 中所用相同的论证推出。□

我们可以将定理 5 应用于多项式衰减模型下的 F_p 矩估计。我们已在第 3 节中描述了用于 F_p 估计的算法（命题 3），因此可以将定理 5 应用于命题 3，从而得到以下结果。

定理 6。 给定一个常数 $p > 2$、一个精度参数 $\varepsilon \in (0, 1)$，以及针对数据流的重击者预言机 O，存在一种单遍算法，可输出多项式衰减模型下矩形 F_p 矩的$(1+\varepsilon)$-近似，且所用空间为 $\widetilde{O}\left(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{2+4/p}}\right)$ 比特。

算法 4. 时间衰减 G 矩估计框架。
输入： 用于$\mathbf{A} \in \mathbb{R}^{k\times n}$ 进行 G 阶矩估计的草图矩阵，后处理函数为 $f(\cdot)$，精度参数为 $\varepsilon \in (0,1)$
令 ν, η, m_ν 按照定义 5 定义
对每个大小为 1 的块 B_i，维护一个使用$\mathbf{A}$ 的线性草图
对于每个时刻 $t \in [m]$ **执行**
$\mathbf{u} \leftarrow \mathbf{0}^k$
对于每个块 B_i **执行**
令$\mathbf{A}\mathbf{v}_i$ 为块 B_i 的线性草图
令 t_i 为块 B_i 中最大的时间步

8: **如果** $t - t_i + 1 \geqslant m_\nu$ **则**9: 删除块 B_i10: **否则，如果**块 B_i 和 B_j 中的所有权重都在 $\sqrt{1+\eta}$ 范围内**则** 11: 合并块 B_i 和 B_j12: **否则** 13: $w_i' \leftarrow \frac{1}{\sqrt{1+\eta}} \cdot w(m - t_i + 1)$14: $\mathbf{u} \leftarrow \mathbf{u} + w_i' \cdot \mathbf{A}\mathbf{v}_i$15: **返回** $f(\mathbf{u})$

现在我们依次给出时间衰减模型的结果。首先考虑多项式衰减模型，其中对于某个固定参数 $s > 0$，有 $w(t) = \frac{1}{t^s}$。对于 F_p 矩估计、矩形矩估计和级联范数，即使坐标最多失真 $(1 + O(\varepsilon))$ 倍， G-矩仍可在 $(1 + \varepsilon)$ 倍因子内保持不变。

引理4.1。对于多项式衰减模型 $w(t) = \frac{1}{t^s}$，只需设置 $\eta = O(\varepsilon)$ 和 $\nu = O\left(\frac{\varepsilon}{pm^{p-1}}\right)$。

证明。我们给出 F_p 问题中 G-矩函数的证明， $G(x) = |x|^p$。级联范数和矩形矩估计的结论可类似推出。

首先注意，对于 $\eta = \frac{\varepsilon}{100p^2} = O(\varepsilon)$，有 $(1+\eta)^p - 1 \leqslant \frac{\varepsilon}{4}$。因此，对于$G(x) = |x|^p$ 以及所有 $x \geqslant 1$，有

$$G((1+\eta)x) - G(x) \leqslant \frac{\varepsilon}{4} G(x),$$

对 $\eta = \frac{\varepsilon}{100p^2} = O(\varepsilon)$ 成立，如所期望。

其次，注意到 $G(1) = 1$，并且由于 $p \geqslant 2$，表达式 $(x+\nu)^p - x^p$ 在 $x = m$ 处达到最大。另一方面，对于 $\nu = \frac{\varepsilon}{100pm^{p-1}}$，我们有 $(m+\nu)^p - m^p \leqslant \frac{\varepsilon}{4}$，因此

$$G(x+\nu) - G(x) \leqslant \frac{\varepsilon}{4} G(1),$$

对所有 $x \in [1, m]$ 成立，如所期望。在这种情况下，我们可以令 $m_\nu = \nu^{-2/s}$，它可能大于也可能不大于 m，但后一种情形并不重要，因为不会存在已被存储$m+1$ 步的块。

□

回顾可知，在标准流模型中， F_p 矩估计可借助如下保证实现:

命题 3. [[AMS99,IW05,AKO11]] 存在一个随机化算法，使得给定$\mathbf{x} \in \mathbb{R}^n$ 作为更新流，可计算出一个 $(1 \pm \varepsilon)$-近似，其对象为 $\|\mathbf{x}\|_p^p$，概率至少为$99/100$，并使用空间 $O(\frac{n^{1-2/p}}{\varepsilon^{2+4/p}} \cdot \log^2 n)$ 空间。

将定理5应用于命题 3，可得到以下用于 F_p 多项式衰减模型矩估计的算法。

定理 7. 给定常数 $p > 2$ 和精度参数 $\varepsilon \in (0、1)$，存在一种单遍算法，可输出多项式衰减模型中 F_p 阶矩的 $(1+\varepsilon)$-近似，所需空间为 $\widetilde{O}\left(\frac{n^{1-2/p}}{\varepsilon^{2+4/p}}\right)$ 比特。

相比之下，采用 [JLL+20] 的方法，我们可以得到以下保证：

定理 8。 给定一个常数 $p>2$、一个精度参数 $\varepsilon\in(0,1)$，以及针对数据流的重击者预言机 O，存在一种单遍算法，可在多项式衰减模型中输出 F_p 阶矩的$(1+\varepsilon)$-近似，并使用 $\widetilde{O}\left(\frac{n^{1/2-1/p}}{\varepsilon^{2+4/p}}\right)$ 比特空间。

类似地，我们可以使用以下线性草图进行矩形 F_p 阶矩估计：

命题 4。 *[TW12]* 给定一个常数 $p>2$ 和一个精度参数 $\varepsilon\in(0,1)$，存在一种单遍算法，使用线性草图并输出流模型中矩形F_p 阶矩的$(1+\varepsilon)$-近似，且使用 $\widetilde{O}\left(\frac{\Delta^{d(1-2/p)}}{\varepsilon^{2+4/p}}\right)$ 比特空间。

将命题 4 应用于定理 5 后，我们的框架可为多项式衰减模型中的矩形 F_p 矩估计提供以下保证。

定理 9. 给定一个常数 $p>2$ 和一个精度参数 $\varepsilon\in(0,1)$，存在一种单遍算法，可输出多项式衰减模型中矩形 F_p 矩的 $(1+\varepsilon)$-近似，其使用 $\widetilde{O}\left(\frac{\Delta^{d(1-2/p)}}{\varepsilon^{2+4/p}}\right)$ 比特空间。

相比之下，采用 [JLL+20] 的方法，可得到以下保证（摘自第 4 节）：

定理 6. 给定一个常数 $p>2$、一个精度参数 $\varepsilon\in(0,1)$，以及一个针对数据流的重击者预言机 O，存在一种单遍算法，可输出多项式衰减模型中矩形 F_p 矩的 $(1+\varepsilon)$-近似，其使用 $\widetilde{O}\left(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{2+4/p}}\right)$ 比特空间。

类似地，考虑指数衰减模型，其中对于某个固定参数 $s\in(0,1]$，我们有 $w(t)=s^t$。对于 F_p 矩估计、矩形矩估计和级联范数，即使坐标被扭曲至多 $(1+O(\varepsilon))$ 倍，G-矩仍保持在 $(1+\varepsilon)$ 倍因子之内。

引理 4.2. 对于指数衰减模型 $w(t)=s^t$，只需令 $\eta=O(\varepsilon)$ 以及 $\nu=O\left(\frac{\varepsilon}{m}\right)$。

证明。 我们再次考虑 F_p 问题的 G-矩函数，$G(x)=|x|^p$，因为级联范数和矩形矩的证明是类似的。首先，对于 $\eta=\frac{\varepsilon}{100p^2}=O(\varepsilon)$，我们有 $(1+\eta)^p-1\leqslant\frac{\varepsilon}{4}$。因此，对于 $G(x)=|x|^p$ 以及所有 ~~$x\geqslant 1$~~，$G((1+\eta)x)-G(x)\leqslant\frac{\varepsilon}{4}G(x)$，符合要求。

其次，回顾 $G(1)=1$。由于 $p\geqslant 2$，数量 $(x+\nu)^p-x^p$ 在 $[1,m]$ 上的最大值于 $x=m$ 处取得。对于 $\nu=\frac{\varepsilon}{100pm}$，我们有 $(m+\nu)^p-m^p\leqslant\frac{\varepsilon}{4}$。因此，对于每个 $x\in[1,m]$，$G(x+\nu)-G(x)\leqslant\frac{\varepsilon}{4}G(x)$。重要的是，这个值意味着我们可以令 $m_\mu=O(\log n/\nu)$。 □

因此，将定理5应用于相关陈述，可得指数衰减模型中 F_p 矩估计的以下结果。

定理10。 给定常数 $p>2$ 和精度参数 $\varepsilon\in(0,1)$，存在一种单遍算法，可输出指数衰减模型中 F_p 矩的 $(1+\varepsilon)$-近似，所需空间为 $\widetilde{O}\left(\frac{n^{1-2/p}}{\varepsilon^{2+4/p}}\right)$ 比特。

定理11。 给定一个常数 $p>2$、一个精度参数 $\varepsilon\in(0,1)$以及针对数据流的重击者预言机 O，存在一种单遍算法，在指数衰减模型中输出F_p矩的$(1+\varepsilon)$-近似，并使用 $\widetilde{O}\left(\frac{n^{1/2-1/p}}{\varepsilon^{2+4/p}}\right)$ 比特空间。

类似地，我们得到了指数衰减模型下矩形 F_p 矩估计的以下结果。

定理 12. 给定一个常数 $p > 2$ 和一个精度参数 $\varepsilon \in (0,\ 1)$，存在一种单遍算法，输出指数衰减模型下矩形 F_p 阶矩的$(1+\varepsilon)$-近似，所需空间为 $\widetilde{O}\left(\frac{\Delta^{d(1-2/p)}}{\varepsilon^{2+4/p}}\right)$ 比特。

定理 13. 给定一个常数 $p > 2$、一个精度参数 $\varepsilon \in (0,\ 1)$，以及数据流的重击者预言机 O，存在一种单遍算法，输出指数衰减模型下矩形 F_p 阶矩的$(1+\varepsilon)$-近似，所需空间为 $\widetilde{O}\left(\frac{\Delta^{d(1/2-1/p)}}{\varepsilon^{2+4/p}}\right)$ 比特。

我们简要说明框架在多项式衰减模型和指数衰减模型中的行为差异。直观而言，在多项式衰减模型中，随着数据流推进，权重需要经过显著更多次更新才会降低一个 $(1+\eta)$ 因子，因此框架会创建对数数量的大块。相比之下，在指数衰减模型中，框架会创建对数数量的小块，但这些块会在 $O(\log n)$ 次更新后被截断。

5 实证评估

5.1 实验设置。

为展示理论结果的实用性，我们比较了非增强型滑动窗口算法及其增强版本。我们实现了 [AMS99] 的范数估计算法（以下简称 AMS），以及 [IW05] 的子采样（SS）算法，用于范数估计和级联范数估计。评估中，我们使用了多种预言机和数据集，具体将在以下各节中介绍。

5.2 预言机与训练

为证明重击者预言机是可行的，我们在实验中使用了几种预言机。在 CAIDA 数据集上的实验中使用了全部三种预言机，而在其他数据集上仅使用了 Count-Sketch 预言机。每种预言机都在数据流前缀上进行训练，并被要求识别在数据流后缀中会成为重元素的项。

Count-Sketch 预言机。 对于第一个预言机，我们实现了广为人知的 Count-Sketch 算法 [CCF04]，用于在数据流中查找重击者。前缀草图化的结果被用作后缀中的重击者。对于合成数据集和 CAIDA 数据集，我们使用长度为 100K 的前缀，将算法重复运行 5 次，使用 300 个哈希桶，并设置 $\varepsilon = 0.1$。对于 AOL 数据集，我们将前缀长度改为 10K，其他参数保持不变。

大语言模型预言机。 对于我们的大语言模型（LLM）预言机，我们向 ChatGPT 和 Google Gemini 提供了相同的 100K CAIDA 前缀，并使用了以下提示词：

给定此流子集，预测未来数据流中出现频率最高的 26 个 IP地址

ChatGPT 和 Google Gemini 预测出的重元素完全相同，因此我们将它们的结果合并为一个大语言模型预言机。由于 Count-Sketch 识别出了 26 个重元素，我们特意要求预测 26 个 IP地址，以便合理比较各预言机的表现。大语言模型与 Count-Sketch 算法在 26 个重击者中有 10 个身份判断一致。

LSTM预言机。 对于我们的 LSTM 预言机，我们使用相同的 100K CAIDA 前缀训练了一个重击者预测器。LSTM 由一个嵌入层、一个单独的 LSTM层和一个全连接输出层组成：嵌入层将全集嵌入到 32 维空间中，LSTM层的嵌入维度为 32、隐藏维度为 64。预测器使用 Adam优化器训练了 50 个周期，损失函数为二元交叉熵（BCE），学习率为 0.001。批量大小设为 64。

5.3 数据集

我们在三个数据集上测试了我们的实现：

(1) 一种合成的偏斜随机整数分布，通过对二项分布进行采样生成，其中 $p = 2 \cdot q/\sqrt{n}$， q 表示所需的重元素数量， $\overline{n}$ 表示不同整数值的数量，即全集大小。(2) CAIDA 数据集[5], , 包含 12 分钟的 IP流量；每分钟约有 30M 个 IP地址。我们使用第一分钟的数据子集测试实现。每个 IP地址都使用 Python 的 ipaddress 库转换为整数值。(3) AOL 数据集 [6], , 包含从 65 万名用户收集的 2000 万条用户查询。每条查询都关联一个匿名用户标识；我们使用这些标识的一个子集测试实现。

5.4 计算设备

我们使用 Python 3.10.18 实现了实验。实验在配备 16GB 随机存取存储器的 Apple MacBook Air M2 上进行。在 CAIDA 上改变窗口大小的实验耗时 1100—1400 分钟。改变样本选择概率的实验耗时 400—600 分钟。AOL 数据集上的运行时间相近。

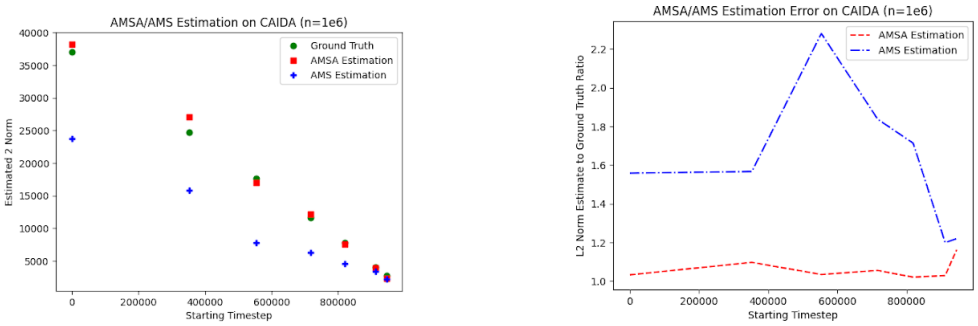
5.5 AMS 与学习增强型 AMS 算法

我们实现了 [AMS99] 的算法，并称之为 AMS，将其作为在 CAIDA 数据集上进行 ℓ_2 范数近似的基线。我们在基线算法中加入了来自预言机的重击者，以比较各算法的性能。为了将流式算法转换为滑动窗口算法，我们跟踪了每种算法的多个实例。每个实例从不同的时间步开始，以对应数据流的不同滑动窗口。当两个实例的 ℓ_2 范数近似值相差不超过两倍时，我们依据 [BO07], 丢弃其中一个实例，并用另一个实例近似被丢弃实例对应的滑动窗口。算法实例最多为 20 个。每个算法实例包含 11 个估计值（使用不同的随机种子获得），因此我们将这些估计值的中位数作为 ℓ_2 范数的估计值。对于哈希函数，我们实现了一种带种子的哈希机制：通过在输入域上计算低次多项式来确定哈希输出，其系数由伪随机数生成器生成。具体而言，我们使用一个种子值初始化 NumPy 的默认随机数生成器

[5] https://www.caida.org/catalog/datasets/passive_dataset

[6] https://www.kaggle.com/datasets/dineshydv/aol-user-session-collection-500k

然后在范围 $[0, p)$ 内采样四个整数系数，其中 p 是一个较大的素数（默认值为 $2^{31}-1$）。请注意，种子设为重复次数。我们将输入流项值强制转换为整数，并代入一个三次多项式；每一项均按p 取模计算，以避免溢出并确保运算处于有限域内。多项式采用增量方式求值，每一步都进行模约简，最后再通过额外的取模运算将结果映射到大小为 2 的输出空间。我们将最终值乘以二再减一，从而将结果映射到 $\{-1,1\}$。



(a) (b)

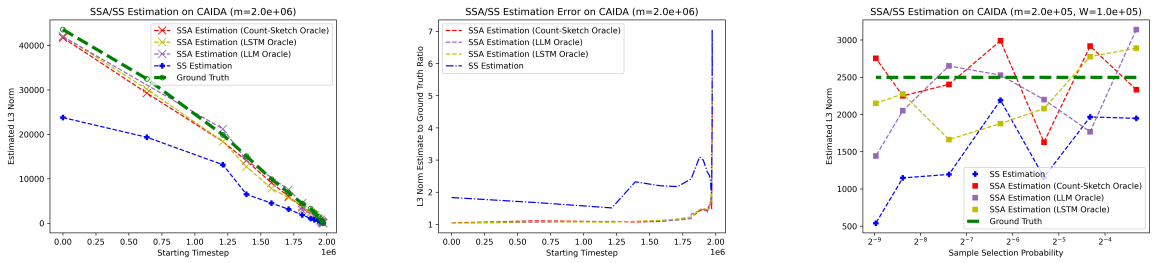
图 2：CAIDA 上的 ℓ_2 范数估计实验。关于符号的说明：图中的变量 n 表示数据流长度（论文其他位置记作 m）。

图 2a 比较了我们的两种算法在多个时间步上的估计结果与实际的 ℓ_2 范数。注意，在图 2a、图 4a 和图 5a 中，n 表示数据流长度（而非全集大小）。x 轴上的“时间步”对应窗口大小：从时间步 0 开始的估计会考虑全部 1M 个流项，即 $W = m$；之后的估计则使用最近的$1M$ –个时间步的值。换言之，$W = m - t_i$，其中 t_i 表示时间步，m 表示数据流长度。如图 2a 所示，在所有选取的窗口大小下，AMSA 的估计值都比 AMS 的估计值更接近真实值，这表明增强算法始终能给出更准确的估计。图 2b 进一步验证了这一点，该图绘制了多个窗口大小下估计值与真实值的比率。如图所示，在所有窗口大小下，AMSA 的估计值与真实值的比率都在 1.2 以内，而 AMS 的比率则在约 1.25 至 2.3 之间变化。此外，AMSA 的估计值与真实值之间的偏差相对稳定，而 AMS 的估计值似乎随着窗口大小减小而逐渐接近真实值，这表明增强算法在各种窗口大小下都更加精确。图 2b 中，AMSA 的误差曲线在所有窗口大小下都较为平坦，尤其与 AMS 的误差线相比，更印证了 AMSA 的精度。

5.6 SS 与学习增强型 SS 算法

对于高阶范数估计，我们实现了 [IW05] 的选择性子采样（SS）算法。与之前一样，我们创建了该算法的增强版本，并在 CAIDA、AOL 和合成数据集上将其性能与基线进行比较。我们使用相同的直方图机制，为两种算法创建了滑动窗口版本。我们将两种算法各运行 15 次：每个算法实例在每个时间步都针对同一窗口保存 3 组估计值，每组包含 5 个估计值（使用不同的随机种子获得）。我们先分别计算每组 5 个估计值的均值，

然后再计算剩余 3 个值的均值，从而得到我们的范数估计值。与之前一样，我们将算法实例数上限设为 20。对于哈希函数，我们将一个随机种子和流项值作为输入，将二者拼接成规范字符串表示，并对这个复合输入计算 SHA-256 加密哈希。请注意，随机种子是通过将流项值与重复次数相加得到的。随后，将生成的 256 位摘要解释为一个大整数，并用它初始化 Python 伪随机数生成器的本地实例。接着，在区间 $[0, 1)$ 内生成一个均匀随机变量。如果哈希值小于我们的样本选择概率（我们将其设为 $q_{ssa} = \frac{1}{100}$ 和 $q_{ss} = \frac{1}{10}$），就对该流项进行采样，从而确定桶计数。

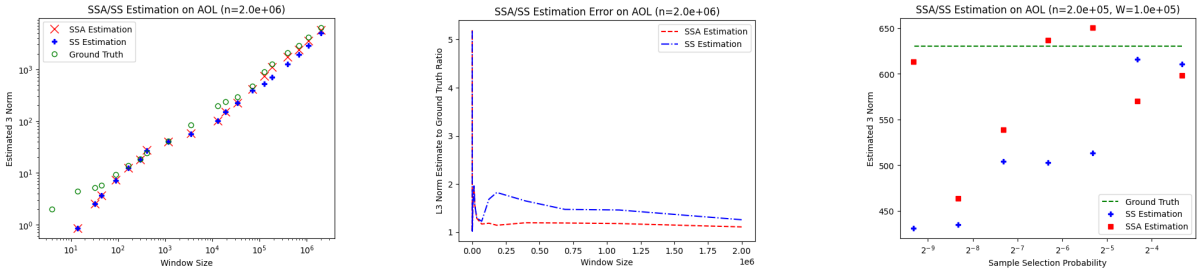


(a) (b) (c)

图 3：使用多个预言机在 CAIDA 上进行 ℓ_3 估计的实验

图 3a 比较了 SSA 和 SS 在各种窗口大小下的估计结果与实际 ℓ_3 范数。三种预言机都提供了有用的增强，使 SSA 在所有窗口大小下估计出的范数都比基线 SS 算法更接近真实值。与 AMSA 类似，SSA 的误差曲线比 SS 的误差曲线更接近 1，也更平坦，这表明 SSA 比未经增强的对应算法更准确、精度更高。

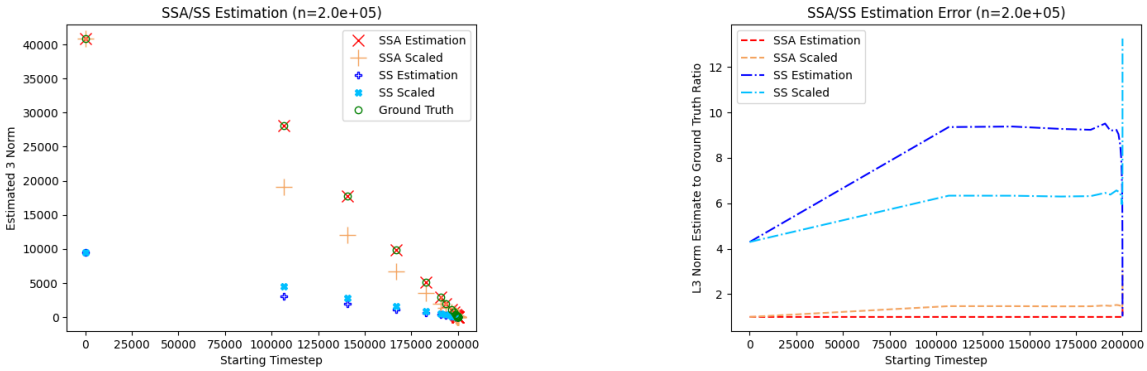
图 3c 比较了在不同样本选择概率下，SSA 和 SS 对实际 ℓ_3 范数的估计结果。同样，三种预言机都有助于增强基线算法，使其在所有选择概率下都能给出更接近真实值的估计。较高的选择概率大致意味着更高的内存使用量，因为必须在估计样本中存储更多元素。SSA 的估计在内存使用量极低时表现尤为出色；在内存使用量较高时，其表现也大幅优于 SS。这意味着，即使基线算法的估计样本规模较大，增强算法仍然有益。



(a) (b) (c)

图 4：在 AOL 上进行 ℓ_3 估计的实验

图 4a 展示了在 AOL 数据集（第二个真实世界数据集）上，不同窗口大小下 SSA 和 SS 对实际 ℓ_3 范数的估计结果。为便于解读，横轴已从时间步转换为窗口大小并采用对数刻度；纵轴也采用对数刻度。结合图 4b 可知，当 $W > 125$、000 时，SSA 比 SS 更准确。不过，鉴于 SS 的误差曲线较为平坦且估计值接近真实值，相较于在 CAIDA 数据集上的表现，SS 似乎能为 AOL 数据集提供更可靠的估计。我们推测，这是因为 AOL 数据集比 CAIDA 数据集更均匀，导致 SSA 相较于 SS 的优势不那么明显。尽管如此，在这种情况下，SSA 仍比 SS 更准确。图 4c 比较了在不同样本选择概率下，SSA 和 SS 对实际 ℓ_3 范数的估计结果。如图所示，当样本选择概率特别低时，SSA 对 ℓ_3 范数的估计比 SS 更准确。随着概率提高，SS 得益于样本规模的增大，最终能够对 ℓ_3 范数给出更好的估计。总体而言，在大多数样本选择概率下，SSA 的估计都更准确，尤其是在概率较低时，这表明它在低空间设置下更有优势。



(a) (b)

图 5：合成数据上的 ℓ_3 估计实验

图 5a 将估计算法 SSA 和 SS 的结果，与第 5 节所述合成生成数据集在多个窗口大小上的实际 ℓ_3范数进行了比较。此外，我们还纳入了“缩放版 SSA”和“缩放版 SS”，它们通过对 $W = m$ （最大窗口大小）的估计值按 $\frac{W}{m}$ 进行缩放而得到，用以估计更小的窗口大小。这些方法旨在构造自然的启发式方法，从而将原始流式算法转化为滑动窗口算法。直观而言，如果分布在数据流上保持不变，仅通过重新缩放来估计更小的窗口应当效果良好。然而，我们的合成数据刻意包含分布偏移，以分析当分布发生变化时，我们的增强算法 SSA 是否能够带来益处。如图 5a 所示，非增强算法 SS 和缩放版 SS 与真实值的偏离显著大于增强算法 SSA 和缩放版 SSA。图 5b 中的误差曲线也支持这一点，表明随着窗口大小缩小，增强算法与非增强算法之间的差距增大，凸显对抗性分布偏移会导致算法失去准确度。在 SSA 与缩放版 SSA 之间，SSA 在各个窗口大小上给出的估计值都更接近真实值。

图6比较了估计算法 SSA 和 SS 在我们合成生成的数据集上，针对多个窗口大小得出的结果与实际 $(k,\ p)$ 级联范数的差异。在所有

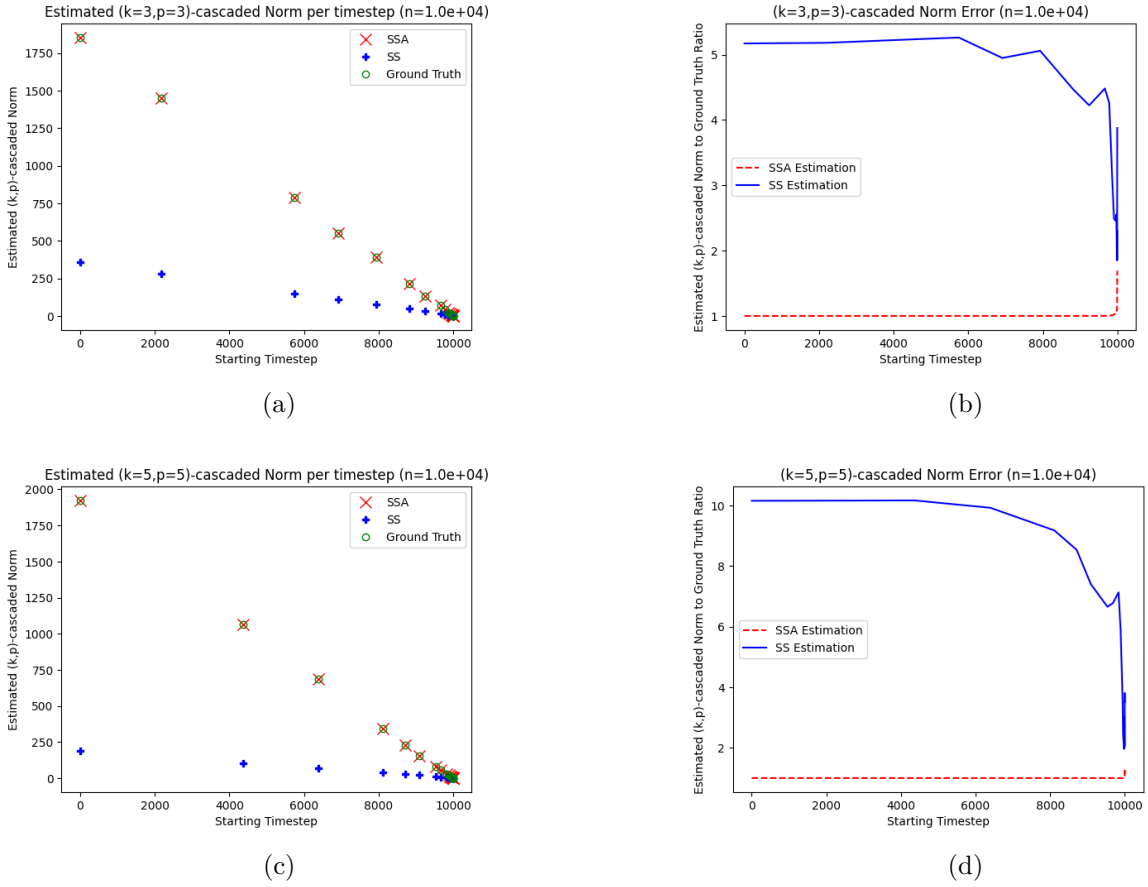


图 6：在合成数据上进行 (k, p)-级联范数估计的实验

图6a 和图6c 所示的窗口大小范围内，增强算法 SSA 的估计质量远高于 SS。如图6b 和图6d 所示，SSA 估计值与真实值的比率在所有窗口大小下几乎保持不变。相反，随着窗口大小增大，SS 的估计效果似乎呈指数级下降。此外，与 SS 对 $(k = 3,\ p = 3)$ 的估计相比，SS 对 $(k = 5,\ p = 5)$ 级联范数的估计误差大了一倍，而 SSA 的表现基本相同。这表明，SSA 在高阶范数上的表现相对稳定，而 SS 的性能下降则更加明显。综合来看，这些图表表明，在基线算法中加入重频项，可为估计不同窗口大小下的 $(k,\ p)$ 级联范数提供有用信息，从而提高估计质量。除了估计质量外，我们还监测了两种算法的内存使用量和运行时间。在 $(k = 5,\ p = 5)$ 级联范数估计中，SSA 使用了 68.86 MB 随机存取存储器，而 SS 使用了 74.63 MB，这与我们预期增强算法占用内存更少相符。对于 $(k = 3,\ p = 3)$ 级联范数估计，趋势也相似：SSA 使用了 112.32 MB 随机存取存储器，而 SS 使用了 117.27 MB。此外，对于 $(k = 5,\ p = 5)$，SSA 的运行时间为 40.3 秒，SS 为 63.5 秒；对于 $(k = 3,\ p = 3)$，SSA 的运行时间为 40.1 秒，SS 为 61.8 秒。在这两种设置下，CountSketch 预言机本身使用了 65.96 MB 随机存取存储器。总的来说，这些结果表明，与 SS 相比，SSA 能以更少的内存提供质量更高的 $(k,\ p)$ 级联范数估计值，且运行速度略快。

致谢

David P. Woodruff 部分获得美国海军研究办公室项目（编号 N000142112647）和西蒙斯研究员奖的资助。Samson Zhou 部分获得美国国家科学基金会 CCF-2335411 项目的资助。Samson Zhou 衷心感谢橡树岭联合大学 (ORAU) Ralph E. Powe初级教员提升奖提供的资助。

参考文献

[AAH19] Eiman Alothali、Hany Alashwal 和 Saad Harous。数据流挖掘技术：综述。*TELKOMNIKA*（电信、计算、电子与控制），17(2):728–737，2019。[AB02] Martin Anthony 和 Peter L. Bartlett。神经网络学习——理论基础。剑桥大学出版社，2002。[ACG+25] Anders Aamand、Justin Y. Chen、Siddharth Gollapudi、Sandeep Silwal 和 Hao Wu。学习增强频繁方向。在第十三届国际学习表征会议（*ICLR 2025*），2025。[AKO11] Alexandr Andoni、Robert Krauthgamer 和 Krzysztof Onak。通过精度采样实现流式算法。在*IEEE*第*52*届计算机科学基础年会（*FOCS 2011*），第363–372页，2011。[AMS99] Noga Alon、Yossi Matias 和 Mario Szegedy。近似频率矩的空间复杂度。计算机与系统科学杂志，58(1):137–147，1999。[App21] 苹果公司。差分隐私概述，2021。[BDM+20] Vladimir Braverman、Petros Drineas、Cameron Musco、Christopher Musco、Jalaj Upadhyay、David P. Woodruff 和 Samson Zhou。在线模型与滑动窗口模型中的近最优线性代数。在第*61*届*IEEE*计算机科学基础年会（*FOCS*），第517–528页，2020。[BDSW24] Vladimir Braverman、Prathamesh Dharangutte、Vihan Shah 和 Chen Wang。学习增强最大独立集。在近似、随机化与组合优化：算法与技术，*APPROX/RANDOM 2024*，第317卷*LIPIcs*，第24:1–24:18页，2024。[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。在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。[BEWZ25] Vladimir Braverman、Jon C. Ergun、Chen Wang 和 Samson Zhou。学习增强层次聚类。在第四十二届国际机器学习会议（*ICML*），*2025*，2025。

[BGL+18] Vladimir Braverman、Elena Grigorescu、Harry Lang、David P. Woodruff 和 Samson Zhou。在滑动窗口上近乎最优地计算不同元素和重元素。在近似、随机化与组合优化：算法与技术，*APPROX/RANDOM*，第7:1–7:22页，2018。[BGL+24] Mark Braverman、Sumegha Garg、Qian Li、Shuo Wang、David P. Woodruff 和 Jiapeng Zhang。用于多遍流及其应用的新型信息复杂度度量。在第*56*届*ACM*计算理论研讨会论文集（*STOC*），第1781–1792页，2024。[BGW+26] Vladimir Braverman、Sumegha Garg、Chen Wang、David P. Woodruff 和 Samson Zhou。滑动窗口模型下信息受限的在线学习。在*ACM-SIAM*离散算法研讨会论文集（*SODA*），2026。[BKSV14] Vladimir Braverman、Jonathan Katzman、Charles Seidell 和 Gregory Vorsanger。使用 o (nˆ(1-2/k)) 位计算大频率矩的最优算法。在近似、随机化与组合优化：算法与技术（*APPROX/RANDOM 2014*），第531–544页，2014。[BLLM15] Vladimir Braverman、Harry Lang、Keith D. Levin 和 Morteza Monemizadeh。在多对数空间中对滑动窗口进行聚类。在第*35*届*IARCS*软件技术与理论计算机科学基础年会（*FSTTCS*），第350–364页，2015。[BLLM16] Vladimir Braverman、Harry Lang、Keith D. Levin 和 Morteza Monemizadeh。滑动窗口上的聚类问题。在第二十七届*ACM-SIAM*离散算法研讨会论文集（*SODA*），第1374–1390页，2016。[BLMZ23] Jeremiah Blocki、Seunghoon Lee、Tamalika Mukherjee 和 Samson Zhou。滑动窗口模型下的差分隐私 l_2-重元素。在第十一届国际学习表征会议（*ICLR*），2023。[BLUZ19] Vladimir Braverman、Harry Lang、Enayat Ullah 和 Samson Zhou。改进时间衰减流的算法。在近似、随机化与组合优化：算法与技术，*APPROX/RANDOM 2019*，第145卷*LIPIcs*，第27:1–27:17页，2019。[BO07] Vladimir Braverman 和 Rafail Ostrovsky。滑动窗口的光滑直方图。在第*48*届*IEEE*计算机科学基础研讨会（*FOCS 2007*），第283–293页，2007。[BO13] Vladimir Braverman 和 Rafail Ostrovsky。使用拾放采样近似大频率矩。在组合优化近似算法国际研讨会，第42–57页，2013。[BVWY18] Vladimir Braverman、Emanuele Viola、David P. Woodruff 和 Lin F. Yang。重新审视随机顺序流中的频率矩估计。在第*45*届国际自动机、语言与编程学术讨论会（*ICALP 2018*），第107卷*LIPIcs*，第25:1–25:14页，2018。

[BWZ21] Vladimir Braverman、Viska Wei 和 Samson Zhou。滑动窗口上的对称范数估计与回归。载于计算与组合——第*27*届国际会议 *COCOON* 论文集，2021年。[BYJKS04] Ziv Bar-Yossef、Thathachar S Jayram、Ravi Kumar 和 D Sivakumar。数据流与通信复杂度的信息统计方法。计算机与系统科学杂志，68(4):702–732，2004年。[CCF04] Moses Charikar、Kevin C. Chen 和 Martin Farach-Colton。查找数据流中的频繁项。理论计算机科学，312(1):3–15，2004年。[CdG+24] Vincent Cohen-Addad、Tommaso d'Orsi、Anupam Gupta、Euiwoong Lee 和 Debmalya Panigrahi。最大割及相关问题的学习增强近似算法。载于神经信息处理系统进展 *38*：*2024*年神经信息处理系统年度会议，2024年。[CEI+22] Justin Y. Chen、Talya Eden、Piotr Indyk、Honghao Lin、Shyam Narayanan、Ronitt Rubinfeld、Sandeep Silwal、Tal Wagner、David P. Woodruff 和 Michael Zhang。利用预测对图流中的三角形与四环进行计数。载于第十届国际学习表征会议，*ICLR 2022*，2022年。[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。载于第十三届国际学习表征会议，*ICLR*，2025年。[CKS03] Amit Chakrabarti、Subhash Khot 和 Xiaodong Sun。集合不相交性多方通信复杂度的近最优下界。载于第*18*届 *IEEE* 计算复杂性年度会议（*Complexity 2003*），第107–117页，2003年。[CKT08] Graham Cormode、Flip Korn 和 Srikanta Tirthapura。数据流上的指数衰减聚合。载于 Gustavo Alonso、José A. Blakeley 和 Arbee L. P. Chen 编，第*24*届国际数据工程会议论文集，*ICDE 2008*，第1379–1381页，2008年。[CM05] Graham Cormode 和 S. Muthukrishnan。一种改进的数据流摘要：Count-Min草图及其应用。算法杂志，55(1):58–75，2005年。[CMS13] Michael S. Crouch、Andrew McGregor 和 Daniel M. Stubbs。滑动窗口模型中的动态图。载于算法——*ESA*——第*21*届欧洲年度研讨会论文集，第337–348页，2013年。[CS03] Edith Cohen 和 Martin Strauss。维护时间衰减流聚合。载于 Frank Neven、Catriel Beeri 和 Tova Milo 编，第二十二届 *ACM SIGACT-SIGMOD-SIGART* 数据库系统原理研讨会论文集，第223–233页。ACM，2003年。[CTX07] Graham Cormode、Srikanta Tirthapura 和 Bojian Xu。用于传感器数据聚合的时间衰减草图。载于 Indranil Gupta 和 Roger Wattenhofer 编，第二十六届 *ACM* 分布式计算原理年度研讨会论文集，*PODC 2007*，第215–224页，2007年。

[CTX09] Graham Cormode、Srikanta Tirthapura 和 Bojian Xu。数据流上的时间衰减相关聚合。统计分析与数据挖掘，2(5-6):294–310，2009年。[CWX21] Xu Chen、Junshan Wang 和 Kunqing Xie。Trafficstream：一种基于图神经网络和持续学习的流式交通流预测框架。载于第三十届国际人工智能联合会议论文集，IJCAI 2021，第3620–3626页。ijcai.org，2021年。[DGIM02] Mayur Datar、Aristides Gionis、Piotr Indyk 和 Rajeev Motwani。维护滑动窗口上的流统计。SIAM计算杂志，31(6):1794–1813，2002年。[DPV25] Yinhao Dong、Pan Peng 和 Ali Vakilian。用于近似最大割的学习增强型流算法。载于第16届理论计算机科学创新会议，ITCS 2025，LIPIcs第325卷，第44:1–44:24页，2025年。[EFS+22] Jon C. Ergun、Zhili Feng、Sandeep Silwal、David P. Woodruff 和 Samson Zhou。学习增强的 k-means聚类。载于第十届国际学习表征会议，ICLR 2022，2022年。[EMMZ22] Alessandro Epasto、Mohammad Mahdian、Vahab S. Mirrokni 和 Peilin Zhong。利用基于分桶的草图改进聚类与覆盖的滑动窗口算法。载于2022年 ACM-SIAM离散算法研讨会论文集，SODA，第3005–3042页，2022年。[Fac21] Facebook。https://www.facebook.com/policy.php，2021年。[FNS+25] Chunkai Fu、Brandon G. Nguyen、Jung Hoon Seo、Ryan S. Zesch 和 Samson Zhou。学习增强搜索数据结构。载于第十三届国际学习表征会议，ICLR 2025，2025年。[GDP16] 欧洲议会和理事会关于在处理个人数据及此类数据自由流动方面保护自然人，并废除指令95/46/EC（《通用数据保护条例》）的条例（EU）2016/679，2016年4月27日，2016年。[GG07] Joao Gama 和 Mohamed Medhat Gaber。从数据流中学习：传感器网络中的处理技术。Springer Science & Business Media，2007年。[GLS+22] Elena Grigorescu、Young-San Lin、Sandeep Silwal、Maoyuan Song 和 Samson Zhou。在线线性规划与半定规划的学习增强算法。载于神经信息处理系统进展 35：2022年神经信息处理系统年度会议，2022年。[Goo25] Google LLC。Google如何保留我们收集的数据。Google，2025年。[GZK05] Mohamed Medhat Gaber、Arkady Zaslavsky 和 Shonali Krishnaswamy。数据流挖掘：综述。ACM SIGMOD记录，34(2):18–26，2005年。[HFH+25] Junyu Huang、Qilong Feng、Ziyun Huang、Zhen Zhang、Jinhui Xu 和 Jianxin Wang。学习增强型k-means问题的新算法。载于第十三届国际学习表征会议（ICLR 2025），2025年。

[HIKV19] Chen-Yu Hsu、Piotr Indyk、Dina Katabi 和 Ali Vakilian。基于学习的频率估计算法。载于第七届国际学习表征会议（ICLR 2019），2019。[INW22] Piotr Indyk、Shyam Narayanan 和 David P. Woodruff。单侧误差的频率估计。载于 2022 年 ACM-SIAM 离散算法研讨会论文集（SODA 2022），第 695–707 页，2022。[ISZ21] Zachary Izzo、Sandeep Silwal 和 Samson Zhou。用于 Wasserstein 重心的降维。载于《神经信息处理系统进展》第 34 卷：2021 年神经信息处理系统年会，第 15582–15594 页，2021。[IW05] Piotr Indyk 和 David P. Woodruff。数据流频率矩的最优近似。载于第 37 届 ACM计算理论研讨会论文集，第 202–208 页，2005。[JLL+20] Tanqiu Jiang、Yi Li、Honghao Lin、Yisong Ruan 和 David P. Woodruff。学习增强的数据流算法。载于第八届国际学习表征会议（ICLR 2020），2020。[JW09] T. S. Jayram 和 David P. Woodruff。级联范数的数据流空间复杂度。载于第 50 届 IEEE计算机科学基础研讨会（FOCS 2009），第 765–774 页，2009。[JWZ22] Rajesh Jayaram、David P. Woodruff 和 Samson Zhou。适用于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。[KNPW11] Daniel M. Kane、Jelani Nelson、Ely Porat 和 David P. Woodruff。以最优空间快速估计数据流中的矩。载于第 43 届 ACM计算理论研讨会论文集（STOC 2011），第 745–754 页，2011。[KP05] Tsvi Kopelowitz 和 Ely Porat。改进多项式时间衰减和带加性误差的时间衰减算法。载于《理论计算机科学》，第九届意大利会议（ICTCS 2005），第 3701 卷，第 309–322 页，2005。[LFKF18] Mario Lucic、Matthew Faulkner、Andreas Krause 和 Dan Feldman。利用核心集大规模训练高斯混合模型。《机器学习研究期刊》，18(160)：1–25，2018。[Li08] Ping Li。利用稳定随机投影估计 l_α ($0 < \alpha \leqslant 2$) 中降维的估计量和尾界。载于第十九届 ACM-SIAM离散算法年会论文集（SODA 2008），第 10–19 页，2008。[LLW22] Honghao Lin、Tian Luo 和 David P. Woodruff。学习增强的二叉搜索树。载于国际机器学习会议（ICML 2022），2022。[LT06a] Lap-Kei Lee 和 H. F. Ting。在滑动窗口上维护重要流统计量。载于第十七届 ACM-SIAM离散算法年会论文集（SODA），第 724–732 页，2006。

[LT06b] Lap-Kei Lee 和 H. F. Ting。一种更简单、更高效的确定性方案，用于在滑动窗口上查找频繁项。载于第二十五届 *ACM SIGACT-SIGMOD-SIGART* 数据库系统原理研讨会论文集，第 290–297 页，2006。[Ope24] OpenAI。https://openai.com/enterprise-privacy/，2024。[PEC21] Marco Pulimeno、Italo Epicoco 和 Massimo Cafaro。分布式挖掘时间衰减重频项。信息科学，545：633–662，2021。[SSM24] Rana Shahout、Ibrahim Sabek 和 Michael Mitzenmacher。滑动窗口中的学习增强频率估计。载于第 *32* 届 *IEEE* 国际网络协议会议（*ICNP 2024*），第 1–6 页，2024。[TW12] Srikanta Tirthapura 和 David P. Woodruff。空间数据流中的矩形高效聚合。载于第 *31* 届 *ACM SIGMOD-SIGACT-SIGART* 数据库系统原理研讨会论文集（*PODS*），第 283–294 页，2012。[Woo04] David P. Woodruff。所有频率矩的最优空间下界。载于第十五届 *ACM-SIAM*离散算法研讨会年会论文集（*SODA 2004*），第 167–175 页，2004。[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。[WZ21a] David P. Woodruff 和 Samson Zhou。数据流上大频率矩估计的分离。载于第 *48* 届国际自动机、语言与编程学术讨论会（*ICALP 2021*），第 198 卷（LIPIcs），第 112:1–112:21 页，2021。[WZ21b] David P. Woodruff 和 Samson Zhou。通过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。[WZZ23] David P. Woodruff、Peilin Zhong 和 Samson Zhou。滑动窗口模型中的近似最优 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。[XWP22] Qingjun Xiao、Haotian Wang 和 Guannan Pan。利用衰减感知布谷鸟过滤器沿踢出路径准确识别时间衰减重频项。载于第 *30* 届 *IEEE/ACM* 国际服务质量研讨会（*IWQoS 2022*），第 1–10 页，2022。[XXC12] Chenliang Xu、Caiming Xiong 和 Jason J Corso。流式分层视频分割。载于《计算机视觉—*ECCV 2012*：第 *12* 届欧洲计算机视觉会议》，第 626–639 页，2012。

A 重击者预言机与学习理论

在本节中，我们将讨论使用可能近似正确（PAC）学习框架实现重击者预言机的理论方面。该框架有助于证明，只要输入实例来自固定的概率分布，就能高效地学习到高质量的预测器。关于为学习增强算法实现预言机的讨论历史悠久，参见例如 [ISZ21、EFS+22、GLS+22、BEWZ25], ；我们将此框架用于重击者预言机。

首先，我们假设存在一个底层分布$\mathcal{D}$，输入数据（$\mathbf{x}$的频率向量）从中采样得到。无论是否使用学习增强预言机，这都是解决频率估计问题的标准设置。只要没有发生泛化失败或分布偏移，预言机的机器学习模型就能表现良好。

接下来，我们的目标是从给定的候选函数族$\mathcal{F}$中高效地导出预测函数 f 。任意预测器 f 的输入都是$\mathbf{x}$的频率向量，输出则是一个向量$\{0,1\}^n$ ，用于指示每个$\mathbf{x}_i$ 是否为重频项。然后，我们引入损失函数 $L\colon f \times G \to R$，用于量化预测器 f 应用于特定输入实例$\mathbf{x}$时的准确度。可以将 L 视为一种函数，用来统计预测结果与实际重频项信息相比有误的部分。

我们的目标 i s 学习函数 $f \in \mathcal{F}$，使其最小化以下目标表达式 :

$$\min_{f\in\mathcal{F}} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} [L(f(\mathbf{x}))]\,. \tag{1}$$

设 f^*表示函数族 $\mathcal{F}$ 中的一个最优函数，使得 $f^* = \operatorname{argmin} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} [L(f(\mathbf{x}))]$是使上述目标最小化的函数。假设对于每个频率向量 $\mathbf{x}$以及每个函数 $f \in \mathcal{F}$，我们都能在时间 $T(n)$ 内同时计算 $f(\mathbf{x})$ 和 $L(f(\mathbf{x}))$，则我们可以利用标准的经验风险最小化方法给出以下结果。

定理 14. 存在一种算法，它使用 $\operatorname{poly}\left(T(n)\ \frac{1}{\varepsilon}\right)$ 个样本，并输出一个函数 $\widehat{f}$，使得以至少 $99/100$ 的概率，我们有

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} \left[L(\hat{f}(\mathbf{x}))\right] \leqslant \min_{f} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}} [L(f(\mathbf{x}))] + \varepsilon.$$

本质上，定理 14 是一个PAC式结果，它给出了为实现以高概率学习到近似最优函数所需样本数量的界。

接下来，我们将更详细地讨论定理 14 的证明。我们首先定义函数类的伪维数，它将 VC 维数的概念推广到具有实值输出的函数。

定义 6（伪维数，例如 [LFKF18] 中的定义 9）。设 X 为一个基集，F 为一组将 X 中元素映射到区间 $[0,1]$ 的函数。考虑一个固定集合 $S = \{x_1, \ldots, x_n\} \subset \mathrm{X}$，以及一组实数 $R = \{r_1, r_2, \cdots\ r_n\}$，其中每个 $r_i \in [0,1]$。任取函数 $f \in \mathrm{F}$，子集 $S_f = \{x_i \in S \mid f(x_i) \geqslant r_i\}$ 称为 S 的诱导子集（由函数 f 和实数值 R 决定）。如果不同诱导子集的数量为 $|\{S_f \mid f \in \mathrm{F}\}| = 2^n$，则称集合 S 及其对应的值 R 被 F 打散。于是，F 的伪维数定义为能够被打散的 X 的最大子集的基数（若不存在这样的最大值，则伪维数为无穷大）。

借助伪维数的概念，我们现在可以为经验风险最小化建立准确度与样本复杂度之间的权衡关系。令$\mathcal{H}$ 为由将$\mathcal{F}$ 中的函数与 L 复合而成的函数类；即$\mathcal{H} := \{L \circ f : f \in \mathcal{F}\}$。此外，通过归一化，我们可以假设 L 的取值范围包含于 $[0,1]$ 的取值范围内。一个广为人知的泛化界如下所示。

命题 5 ([AB02])。 设 D 是定义在问题实例集合 X 上的一个分布，H 是一类函数 $h : X \to [0,1]$，其伪维数为 $d_{\mathcal{H}}$。考虑从 D 中抽取的 t 个独立同分布（*i.i.d.*）样本 $\mathbf{x}_1, \cdots, \mathbf{x}_t$。那么，存在一个普适常数 c_0，使得对于任意 $\varepsilon > 0$，若 $t \geqslant c_0 \cdot \frac{d_{\mathcal{H}}}{\varepsilon^2}$，则对所有 $h \in H$，以至少 $99/100$ 的概率有：

$$\left| \frac{1}{t} \cdot \sum_{i=1}^{t} h(\mathbf{x}_i) - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}} [h(\mathbf{x})] \right| \leqslant \varepsilon.$$

下面的推论可直接由命题 5 应用三角不等式得出。

推论 15。 设 $\mathbf{x}_1, \cdots, \mathbf{x}_t$ 是从 D 中抽取的一组独立样本（频率向量），且 $\hat{h} \in H$ 是使 $\frac{1}{t} \cdot \sum_{i=1}^{t} h(\mathbf{x}_i)$ 最小的函数。若样本数 t 按照命题 *5* 中的规定选取，即 $t \geqslant c_0 \cdot \frac{d_{\mathcal{H}}}{\varepsilon^2}$，则以至少 $99/100$ 的概率，有

$$\mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}} \left[\hat{h}(\mathbf{x}) \right] \leqslant \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}} [h^*(\mathbf{x})] + 2\varepsilon.$$

最后，我们可以利用标准结果将伪维数与 VC 维数联系起来。

引理 A.1 (伪维数与 VC 维数，引理 10 见 [LFKF18])。 对于任意 $h \in H$，令 B_h 为阈值函数的示性函数，即 $B_h(x, y) = sgn(h(x) - y)$。那么，$H$ 的伪维数等于子类 $B_{\mathcal{H}} = \{B_h \mid h \in H\}$ 的 *VC* 维数。

引理 A.2 (定理 8.14 见 [AB02])。 令 $\tau \colon \mathbb{R}^a \times \mathbb{R}^b \to \{0,1\}$，定义类

$$\mathcal{T} = \{x \mapsto \tau(\theta, x) : \theta \in \mathbb{R}^a\}.$$

假设任意函数 τ 都可以由一个算法计算：该算法以数对 $(\theta, x) \in \mathbb{R}^a \times \mathbb{R}^b$ 为输入，并在执行以下操作中的至多 t 项后，输出值 $\tau(\theta, x)$：

- 对实数进行算术运算 $+, -, \times, /$，
- 进行涉及 $>, \geqslant, <, \leqslant, =$ 的比较，并输出比较结果，
- 输出0或1。

则 T 的 *VC* 维数以 $O(a^2t^2 + t^2 a \log a)$ 为上界。

根据引理 A.1 和引理 A.2，我们可以直接界定概念类 $\mathcal{F}$的 VC 维数，进而界定概念类 $\mathcal{F}$的伪维数。这就补齐了证明定理 14 所需的最后一块拼图。

定理 *14* 的证明。 由引理 A.1 和引理 A.2 可知，利用 $a = n$ 和 $t = T(n)$，$\mathcal{F}$的伪维数上界为 $O(n^2 \cdot T^2(n))$。进而，该上界可进一步界定为 poly($T(n)$)。因此，根据推论 15，我们只需要 poly($T(n)$)$/\varepsilon^2$ 个样本。我们假设可以在 $T(n)$时间内计算 $f(\mathbf{x})$ 和 $L(f(\mathbf{x}))$；使用任意多项式时间的经验风险最小化算法，可得总运行时间为 poly($T(n)$ $1/\varepsilon$)，符合预期。 □

需要注意的是，定理 14 是一个适用于学习增强预言机的通用框架。如果所考虑的预言机族中的每个函数都能高效计算，那么定理 14 可以保证，只需多项式数量的样本，就足以学习到一个近乎最优的预言机。