计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440481
61(10): 2433−2447, 2024

基于采样的数据流差分隐私快速发布算法

王修君[1,2] 莫 磊[3] 郑 啸[1,2] 卫琳娜[1,2] 董 俊[4] 刘 志[5] 郭龙坤[3]

[1](安徽工业大学计算机科学与技术学院 安徽马鞍山 243032)

[2](安徽省工业互联网智能应用与安全工程研究中心(安徽工业大学) 安徽马鞍山 243032)

[3](福州大学数学与统计学院 福州 350108)

[4](中国科学院合肥物质科学研究院智能机械研究所 合肥 230031)

[5](电气通信大学 日本东京 163-8001)

Sampling Based Fast Publishing Algorithm with Differential Privacy for Data Stream

Wang Xiujun[1,2], Mo Lei[3], Zheng Xiao[1,2], Wei Linna[1,2], Dong Jun[4], Liu Zhi[5], and Guo Longkun[3]

[1](*School of Computer Science and Technology*, *Anhui University of Technology*, *Ma'anshan*, *Anhui* 243032)

[2](*Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet* (*Anhui University of Technology*), *Ma'anshan*, *Anhui* 243032)

[3](*School of Mathematics*, *Fuzhou University*, *Fuzhou* 350108)

[4](*Institute of Intelligent Machines* , *Hefei Institute of Physical Science*, *Chinese Academy of Sciences*, *Hefei* 230031)

[5](*The University of Electro-Communications*, *Tokyo*, *Japan* 163-8001)

Abstract Many cloud native database applications need to handle massive data streams. To analyze group trend information in these data streams in real time without compromising individual user privacy, these applications require the capability to quickly create differentially private histograms for the most recent dataset at any given moment. However, existing histogram publishing methods lack efficient data structures, making it difficult to rapidly extract key information to ensure real-time data usability. To address this issue, we deeply analyze the relationship between data sampling and privacy protection, and propose a sampling based fast publishing algorithm with differential privacy for data stream (SPF). SPF introduces an efficient data stream sampling sketch structure (EDS) for the first time, which samples and statistically estimates data within a sliding window and filters out unreasonable data, enabling rapid extraction of key information. Then, we demonstrate that the approximations output by the EDS structure are theoretically equivalent to adding differential privacy noise to the true values. Finally, to meet the privacy protection strength provided by the user while reflecting the true situation of the original data stream, an adaptive noise addition algorithm based on efficient data stream sampling is proposed. According to the relationship between the user-provided privacy protection strength and the privacy protection strength provided by the EDS structure, the algorithm adaptively generates the final publishable histogram through privacy allocation. Experiments show that compared with existing algorithms, SPF significantly reduces time and space overhead while maintaining the same data usability.

Key words cloud native database; sliding window; data stream; differential privacy; data sampling; data publication

收稿日期: 2024-05-31; **修回日期**: 2024-07-22

基金项目: 国家自然科学基金项目(62172003, 12271098, 61772005); 安徽省自然科学基金项目(2108085MF218, 2108085MF217); 安徽省高校自然科学研究项目(2022AH040052); 马鞍山市科技创新项目(2021a120009)

This work was supported by the National Natural Science Foundation of China (62172003, 12271098, 61772005), the Natural Science Foundation of Anhui Province (2108085MF218, 2108085MF217), the Natural Science Research Project of Anhui Educational Committee (2022AH040052), and the Science and Technology Innovation Program of Ma'anshan (2021a120009).

通信作者: 郭龙坤(lkguo@fzu.edu.cn)

摘 要 基于云原生数据库的许多应用场景需要处理海量的数据流.为了实时分析数据流中的群体趋势信息而又不泄露单个用户的隐私,这些应用需要在每个时刻都可以为数据流中的最近数据集快速创建可以安全发布的差分隐私直方图.然而,现有的直方图发布方法因缺乏高效数据结构,导致无法快速提取关键信息以确保数据的实时可用性.为解决此问题,深入分析数据采样与隐私保护之间的关系,提出基于采样的数据流差分隐私快速发布算法SPF(sampling based fast publishing algorithm with differential privacy for data stream).SPF首创高效数据流采样草图结构(efficient data stream sampling sketch structure, EDS),EDS对滑动窗口内数据进行采样统计估计,并过滤不合理数据,实现了对关键信息的快速提取.然后,证明EDS结构输出的近似值理论上等效于对真实值添加差分隐私噪声.最后,为了满足用户所提供的隐私保护强度,并且避免正确反映原始数据流的真实情况,提出了一种基于高效数据流采样的自适应加噪算法.根据用户的隐私保护强度和EDS结构所提供的隐私保护强度之间的关系,通过隐私分配的方式自适应生成最终可发布直方图.实验证明,相较于现有算法,SPF在保持相同数据可用性的前提下显著降低了时间和空间开销.

关键词 云原生数据库;滑动窗口;数据流;差分隐私;数据采样;数据发布

中图法分类号 TP391

随着信息技术的迅猛发展,现如今正面临着越来越多基于云原生数据库的实时数据流分析应用场景[1-2].例如,在网络流量分析领域,云原生数据库可以存储和处理大量用户浏览行为数据流,以支持网站工作人员进行实时分析和商品推荐[3-5].在车辆交通监测领域,云原生数据库能够存储车辆位置和移动数据流,并为实时交通信息系统提供支持,从而帮助优化交通流量和路线规划[6-8].在金融交易监控领域,云原生数据库能够处理大量的交易数据流,支持实时监测和风险管理决策[9-11].

在这些云原生数据库的应用场景中,快速生成和发布基于最新数据的直方图信息,以反映数据流中的最近群体趋势是至关重要的.这是因为随着数据处理量的激增和实时性的不断提高,当前的应用程序更加注重最近的数据而非历史数据,因为最近的数据更有价值,更能反映当前数据的变化趋势.例如,苹果公司在其云数据库中仅保留最近3个月的用户数据,超过此时间范围的数据往往被视为过时,从推荐的角度来看,任何超过3个月的数据都可能被认为是过时的[12-13].进一步而言,数据流在线直方图发布算法作为云原生数据库的关键组成部分,对实现数据流分析的实时性和准确性至关重要.因此,研究和优化这些算法不仅能够提升云原生数据库的性能和效率,还将直接影响到实际应用的数据分析能力和决策结果.

本文研究致力于解决一个具有挑战性的问题:如何在不断涌入新数据的无限数据流中快速而安全地生成数据直方图并进行发布.具体而言,本文关注于在任意给定一个滑动窗口大小w和一个隐私预算ε(隐私保护强度参数)的情况下,如何实时生成并发布差分隐私直方图的问题.该问题的关键要求是,需要在每个时刻都能够快速生成一个实时的差分隐私直方图,同时确保该直方图既可以有效地反映当前滑动窗口(数据流中最新的w条数据)内数据的群体趋势信息,又可以保护这w条数据中每个用户的隐私信息.本文以数据流中每个时刻的滑动窗口快速生成差分隐私直方图为研究目标有2个原因:首先,差分隐私是针对最严格的攻击者模型而建立的隐私保护技术之一,具有广泛的应用前景,尤其是在云原生数据库等领域[1-2,14-17];其次,差分隐私建立在严格的数学理论基础上,并允许用户根据实际的应用场景设置合适的量化隐私保护强度[18-23].此外,本研究对于云原生数据库中的数据流处理具有重要的意义.在云环境下,对数据流进行实时分析和处理是至关重要的,而保护用户隐私则是不可或缺的要求.因此,解决在不断涌入的数据流中实时生成差分隐私直方图的问题,将为云原生数据库等领域的数据管理和隐私保护提供有力的技术支持[1-2,14-23].

目前,对于差分隐私直方图发布算法,无论是在静态数据还是动态数据方面,都有大量关于数据发布的研究工作[24-35].然而,现有文献中提出的方法并未充分适用于数据流场景和滑动窗口模型.注意:一般来说,对于数据流场景,人们一般要求所设计的算法具有在线性,即算法需要较低的时间和空间开销[36-37];对于滑动窗口模型,人们只关心数据流中最近的w条数据.

具体来说，这些已有的算法在发布数据流滑动窗口差分隐私直方图时，存在3个主要缺点：

1)对窗口内最近元素的关注不足. 最近元素是指在数据流处理中，滑动窗口内最近w条数据. 现有的直方图发布方法主要关注整体数据的统计特性，而不是特别关注滑动窗口内最近w的数据，且未考虑到数据流的特性，缺乏专门用于快速统计数据流滑动窗口中直方图的方法.

2)数据实时缓存需求导致高存储开销. 现有数据流直方图发布方法通常需要实时缓存整个窗口中的所有数据，即每个时刻都需要完整存储当前的最近w条数据. 这种存储方式导致较高的存储开销，并且缺乏对噪音的有效量化.

3)直接统计和对数据加噪带来的高时间开销. 现有的直方图发布方法直接对当前窗口内的全部数据进行统计，以生成准确的直方图，随后通过添加噪音的方式生成可发布的直方图. 然而，这种方法在每次生成可发布的直方图时都需要扫描当前窗口内的所有数据，因而带来相对较高的时间开销.

针对上述问题，提出了一种基于采样的数据流差分隐私快速发布算法(sampling based fast publishing algorithm with differential privacy for data stream, SPF)，以实现对滑动窗口数据的高效发布，并在保障隐私的同时降低时间和空间开销，如图1所示. 该算法巧妙地将实时发布算法与滑动窗口采样相结合，使其适用于数据流的直方图数据发布. 通过设计一种新颖的内存高效的数据结构——高效数据流采样草图结构(efficient data stream sampling sketch structure, EDS)，并利用数据流滑动窗口采样机制与差分隐私保护机制之间的内在关系，实现了自适应的加噪，生成符合隐私保护要求的直方图.

本文的主要贡献包括3个方面：

1)新颖的内存高效的数据流采样草图结构. SPF算法采用了滑动窗口模型来获取当前数据流中最新的数据，并提出了一种高效数据流采样草图结构EDS. 该结构在每个时刻对当前滑动窗口内的数据进行采样统计估计，通过非负约束过滤掉不符合要求的估计数据，从而实现对数据流中最近元素(每个时刻滑动窗口中的w条数据)的关键直方图信息的高效提取. 该草图结构可以提供完全符合差分隐私定义的噪声值，即证明了EDS输出的近似值从理论上等效于真实值加上一个满足(ε,δ)-差分隐私的噪声值. 该草图结构适用于滑动窗口模型连续发布直方图，并能够在使用较低的空间开销的情况下估计当前滑动窗口的统计计数值，并提供严格可控的数据保护能力.

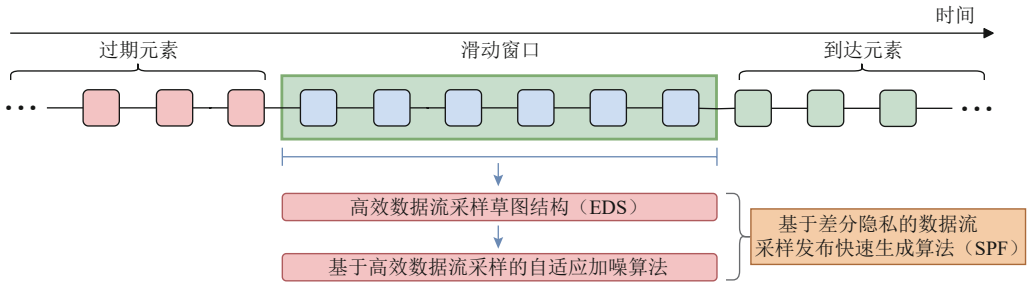


Fig. 1 Sampling based fast publishing algorithm with differential privacy for data stream

图1 基于采样的数据流差分隐私快速发布算法

2)基于高效的数据流采样的自适应加噪算法. EDS算法的输出值是加噪后的近似值，尽管该值在一定程度上反映了原始数据流的真实情况，但仍可能暴露部分敏感信息. 为了解决这个问题，提出了一种基于高效数据流采样草图结构的自适应加噪算法. 该算法根据隐私保护强度与用户需求的隐私保护强度之间的差异来自适应添加噪音，满足了用户所需的隐私保护强度. 该算法提高了差分隐私直方图生成速度，并保证了相同的隐私保护强度.

3)隐私性和时空复杂度分析. 本文首先对SPF算法的隐私性进行了理论分析，使其满足差分隐私定义. 然后，对SPF算法的时间和空间复杂度进行了理论分析，以全面了解其性能. 最后，通过在真实数据集的实验分析，结果表明SPF算法可以使用较少的内存开销来快速实时生成直方图数据，同时发布的数据仍然提供了相同的可用性.

1 相关工作

目前现有的基于差分隐私的直方图发布方法可以分为2类：1)静态数据的直方图发布方法；2)动态数据的直方图发布方法.

1.1 静态数据的直方图发布方法

最近的许多工作都致力于静态数据的直方图发布方法[24-27]. 静态数据的直方图发布方法是基于一个固定的数据集进行分析和发布的. 这意味着数据在发布之前已经全部收集完毕, 不再发生变化. Xu 等人[20]提出了 SF(structure first)算法, 其核心思想在于通过启发式压缩相似频率的间隔, 以减少查询误差. 经过理论分析和实验证明, 这一方法取得了良好的效果. Zhang 等人[25]提出了一种聚类方案 AHP (accurate histogram publication), 该方案可以均衡聚类引起的近似误差和拉普拉斯噪声注入引起的拉普拉斯误差, 提高直方图发布的准确性. 张啸剑等人[26]提出了一种精确的直方图发布算法 DiffHR(differentially private histogram release), 使用 Metropolis-Hasting 算法并联合指数排序方法对直方图中的每个区间数据进行排序, 然后根据贪心聚类的思想对排序数据进行分组, 提高直方图数据的可用性. 唐海霞等人[27]提出了隐私预算自适应分配方法 APB (adaptive privacy budget), 通过优化隐私预算权重分配模型, 计算出使得总误差最小的隐私预算权重分配比例, 并通过贪心思想进行分组, 均衡噪声误差和重构误差, 提高数据的可用性.

1.2 动态数据的直方图发布方法

最近的一些研究工作提出了动态数据的直方图发布方法[28-35]. 动态数据直方图发布是针对不断更新的数据流进行的, 这意味着数据是实时获取和处理的. 林富鹏等人[28]提出了一种定长二维数据流滑动窗口差分隐私统计发布算法 PTDSS-SW, 针对二维空间数据流, 以较低空间开销实现隐私保护. 张啸剑等人[29]提出了 SHP(streaming histogram publication method with differential privacy), 将滑动窗口中的桶计数分组, 并根据数据采样结果自适应分配隐私参数, 降低整体的隐私预算. Sun 等人[30]将 Fenwick 树和矩阵优化结合在一起, 提出了一种完整的差分私有实时流数据发布算法 RTP_DMM, 在保证查询质量的同时有效提高了查询效率. Wu 等人[31]提出了一种以 Kullback-Leibler 散度为测度的直方图发布算法 HPSSGC(histogram publishing algorithm based on sampling sorting and greedy clustering). 该算法使用 KL 散度计算相邻数据之间的变化量, 并在发布的数据中加入不同的噪声值, 以减少基于 KL 散度计算的不同值的噪声误差. Cao 等人[34]提出了一种元算法, 该算法可以使用现有的一次性 Top-k 方法作为 DP 算法子程序, 从流中连续释放私有直方图. 2023 年, Lebeda 等人[35]提出了一种基于经典 Misra Gries 草图结构的在线数据发布方法 MGP[38,39], 有效地平衡了数据的隐私性和实用性. 分析和实验结果证实, 与以前的直方图发布方法相比, 该方法可以提高处理效率和数据效用.

综上所述, 不论是针对静态数据还是动态数据, 现有的数据发布算法在处理数据流滑动窗口模型时都面临 3 个问题:

1)现有的动态数据直方图在实时数据流环境中具有潜在的应用价值, 但其在数据处理延迟和空间消耗方面的挑战限制了其实际应用;

2)现有的数据流算法在数据发布过程中, 未能充分考虑利用差分隐私和滑动窗口采样之间的数据内在相关性;

3)现有的数据流直方图发布算法在加噪过程中仅直接对统计数据添加噪音, 未能充分满足用户对隐私保护强度的要求.

本文针对现有工作存在的问题, 提出了一种基于差分隐私的数据流采样发布快速生成算法(SPF), 如图 1 所示. SPF 算法首先提出了内存高效的数据流采样草图结构, 它能够对当前滑动窗口内的数据在每个时间戳进行采样估计, 并且该草图结构能够提供严格可控程度的数据保护能力. 然后, 本文提出了一种基于高效数据流采样的自适应加噪算法, 该算法提高了直方图生成的速度, 减少了运行时间, 并提供了满足用户所需要的隐私保护强度.

2 定义与模型

在本节中, 主要介绍数据流差分隐私相关的定义与模型. 在表 1 中总结了本文所有常用的符号.

Table 1 Commonly Used Symbols

表 1 常用符号

符号	描述
e_t	在数据流DS中当前时间点t的元素
W	滑动窗口
w	滑动窗口的大小
S^+	采样中间集合
s^+	采样中间集合大小
S	采样集合
s	采样集合的大小
b	随机性增强因子
r	随机因子
l	直方图区间个数
ε	隐私预算
β	隐私预算分配比例

2.1 数据流和滑动窗口模型

数据流被定义为一个可数无限的元素序列，并能够用于表示随时间推移可用的数据元素. 数据流被广泛应用于众多数据处理应用中，包括环境监测应用中的传感器读数、金融应用中的股票报价或计算机监测中的网络数据等.

定义 1. 数据流. 给定一个无限数据流集 $DS=\{e_1, e_2, \cdots, e_t, \cdots\}(t \geqslant 0)$表示某个时间到达系统时所对应的元素.

滑动窗口模型是指在每一时刻处理最近的滑动窗口内的数据，并对该窗口内的所有数据进行统计计数处理. 滑动窗口模型能够很好地处理历史数据和近期数据. 滑动窗口模型定义如定义 2 所示.

定义 2. 滑动窗口模型. 给定一个无限数据流集 $DS=\{e_1, e_2, \cdots, e_t, \cdots\}(t \geqslant 0)$，滑动窗口大小 w，当前时间戳 t 的滑动窗口包含 $[t-w, t]$ 之间的 w 个元素.

2.2 差分隐私

如果一个机制的结果不受单个用户的删除或添加的显著影响，那么该机制就是差分隐私的. 因此，攻击者无法获取任何用户的信息，而不管该用户在原始数据库中是否存在.

定义 3. 相邻的数据流. 给定 2 个数据流 D 和 D'，如果 D 和 D' 之间最多有 1 个不同的元素，则表示 D 和 D' 是相邻的数据流.

定义 4. 全局敏感度. 对于任意函数 f，定义函数 f 的全局敏感度为：

$$\Delta f = \max \| f(D) - f(D') \|_2.$$

定义 5. (ε, δ)-差分隐私. 给定 2 个相邻的数据流 D 和 D'，当且仅当对任意输出集 $O \subseteq Range(A)$ 存在以下不等式时，算法 G 达到差分隐私保护要求.

$$Pr[G(D)=O] \leqslant \mathrm{e}^{\varepsilon} \times Pr[G(D')=O] + \delta,$$

其中，$Range$ 表示算法 G 的可能输出的集合，ε 表示隐私预算，δ 表示松弛项. 请注意，概率 Pr 来自于算法 G 的内部随机性. 显然，隐私保护强度随着隐私预算 ε 的增加而增加.

定义 6. 高斯噪声机制. 高斯噪声机制是从高斯分布中提取的噪声，其方差根据敏感度和隐私参数进行校准. 对于任意 $\delta \in (0,1)$，$\varepsilon \in (0,1)$，其机制定义为：

$$M_{\text{gauss}}(D, f, \varepsilon, \delta) = f(D) + N\left(u, \sigma^2\right),$$

其中，$u=0$，$\sigma^2 = \dfrac{2\ln\left(\dfrac{1.25}{\delta}(\Delta f)^2\right)}{\varepsilon^2}$. 高斯噪声机制提供 (ε, δ)-差分隐私.

定义 7. 差分隐私串行组合性质. 令 M_i 提供 ε_i-差分隐私. $M_i(D)$ 的序列提供 $\left(\sum_i \varepsilon_i\right)$-差分隐私，其中 M_i 表示一个满足 ε_i-差分隐私算法，而 $M_i(D)$ 表示所有的算法作用于同一个数据集 D，那么这些算法构成的集合满足 $\left(\sum_i \varepsilon_i\right)$-差分隐私.

3 基于差分隐私的数据流采样发布快速生成算法（SPF）

3.1 高效数据流采样草图结构

SPF 算法采用了滑动窗口模型来获取当前数据流中最新的数据，并提出了一种高效数据流采样草图结构（EDS），如算法 1 所示. 为了更好地描述相关算法，现对相关定义进行阐述.

1）采样中间集合. 实时数据流中通过滑动窗口模型的这些数据中随机抽取的采样集合大小为 s^+ 的数据集合.

2）采样集合. 从采样中间集合中随机抽取的采样集合大小为 s 的数据集合，用于近似代表整个数据流的特征.

3）过期数据. 不再处于当前滑动窗口内的数据，这些数据在新的窗口内不再具有代表性，应被移除或忽略.

算法 1. EDS.

输入：时间 t，滑动窗口大小 w，采样中间集合 S^+，采样集合 S，采样中间集合大小 s^+，采样集合大小 s，随机性增强因子 b；

输出：在每一个时刻的采样集合 S 的元素.

① 初始化 w，S^+，S，s^+，s，b；
② for $t \in [1, s^+]$ do
③ $S_1^+(t)=t$，$S_2^+(t)=e_t$；
④ for $t \in (1, s)$ do
⑤ 将 S^+ 中的数据赋值给 S 中的数据；
⑥ end for
⑦ for $t \in (s, s^+)$ do
⑧ $r_1 = randi([1, t])$；
⑨ if $r_1 < s^+$ then
⑩ $r_2 = randi([1, s^+])$；
⑪ $S_1^+(r_2)=t$，$S_2^+(r_2)=e_t$；
⑫ end if
⑬ $S = Delete(S^+, \max(0, s^+ - s))$；
⑭ end for
⑮ end for
⑯ for $t \in (s^+, w)$ do

⑰ $r_1 = randi([1,t])$;
⑱ if $r_1 < s^+$ then
⑲ $r_2 = randi([1,s^+])$;
⑳ $S_1^+(r_2) = t,\ S_2^+(r_2) = e_t$;
㉑ end if
㉒ S=Delete(S^+, $s^+ - s = b$);
㉓ end for
㉔ for $t > w$ do
㉕ $r_1 = randi([1,t])$;
㉖ if $r_1 < s^+$ then
㉗ $r_2 = randi([1,s^+])$;
㉘ $S_1^+(r_2) = t,\ S_2^+(r_2) = e_t$;
㉙ end if
㉚ S=*Delete*(S^+, $s^+ - s = b$);
㉛ end for

算法 1 的具体步骤如下所示:

1)对于当前时刻 $t\in[1,s^+]$内,把每一个时刻的时间戳和计数值都记录下来.其中当前时刻 $t\in[1,s]$,采样集合的数据与采样中间集合的数据一致;当前时刻 $t\in[s,s^+]$,从采样中间集合S^+中随机删除$t-s$个元素作为采样集合(行②~⑭).

2)对于当前时刻 $t\in[s^+,w]$,首先检查采样中间集合中是否有过期数据,如果存在过期元素,则加入最新的数据,如果没有过期元素,则以随机的概率 $\frac{s+b}{t}$ 进行替换.然后,从采样中间集合S^+中随机删除b个元素作为采样集合(行⑮~㉑).

3)对于当前时刻$t>w$,首先检查采样中间集合中是否有过期数据,如果存在过期元素,则加入最新的数据(即当前采样集合中有随机的概率 $\frac{s+b}{w}$ 的过期元素)[①];如果没有过期元素,则以随机的概率 $\frac{s+b}{w}$ 进行替换.然后,从采样中间集合S^+中随机删除b个元素作为采样集合(行㉒~㉘).

高效数据流采样草图结构能够在每个时间戳对当前滑动窗口内的数据进行采样统计估计,实现了对数据流中关键直方图信息的高效提取.该草图结构可以提供完全符合差分隐私定义的噪声值——证明了 EDS 结构输出的近似值从理论上等效于真实值加上一个满足(ε,δ)-差分隐私的噪声值.该草图结构适用于滑动窗口模型连续发布直方图,并能够在使用较低的空间开销时估计当前滑动窗口的计数值.该 EDS 结构提供严格可控程度的数据保护能力.

3.2 基于高效数据流采样的自适应加噪算法

本文提出了一种基于高效数据流采样的自适应加噪算法,如算法 2 所示.

算法 2. 基于高效数据流采样的自适应加噪算法.

输入:时间t,滑动窗口大小w,采样集合S,采样集合大小s,隐私预算分配比例β,EDS 采样集对应的区间计数η_i,隐私预算ε;

输出:发布噪音直方图$\tilde{H}_{W_t}$.

① 分配隐私预算:$\varepsilon_1=\beta\varepsilon$, $\varepsilon_2=(1-\beta)\varepsilon$;

② 计算区间的阈值误差:$error_{\min}=\frac{2\ln(1.25/\delta)}{\varepsilon_1^2}+\frac{2\ln(1.25/\delta)}{\varepsilon_2^2}$;

③ 计算隐私预算分配参数β;

④ 采样集合区间计数:$\{\eta_1,\eta_2,\cdots,\eta_l\}$;

⑤ 计算当前时间t的滑动窗口区间计数:$H_{W_t}=\{\eta_1,\eta_2,\cdots,\eta_l\}\times\frac{w}{s}$;

⑥ 发布噪音直方图:$\tilde{H}_{W_t}=H_{W_t}+Gaussian\left(\frac{\Delta f}{\varepsilon_2}\right)$.

算法 2 的具体步骤如下所示:

1)分配 2 段隐私预算的比例(行①).

2)根据基于分组的高斯噪声来计算差分隐私分配参数β(行②~③).

3)根据高效数据流采样的计算结果,计算区间结果(行④~⑤).

4)添加第 2 部分噪声值(行⑥).

基于高效数据流采样的自适应加噪算法通过衡量隐私保护强度与用户需求的隐私保护强度之间差异来自适应添加噪声,满足用户所需的隐私保护强度.该算法在提高数据流差分隐私直方图生成速度的同时,保证了满足用户要求隐私预算下的相似的隐私保护强度.

4 理论分析

4.1 EDS 算法的理论基础

本节中,我们深入探讨了 EDS 结构的理论基础,特别是关注了该算法中若干关键引理和定理的作用

① 当前采样集合中有随机的概率为 $\frac{x+b}{x}$ 的过期元素的具体概率分析:Ⅰ.针对滑动窗口内 w 个元素中选择 $s+b$ 个元素的组合数 C_{s+b}^{w};Ⅱ.针对每个时刻只有 1 个元素过期,w 个元素中选择 $s+b-1$ 个元素的组合数 C_{s+b-1}^{w-1};Ⅲ.有 1 个元素过期时,选择过期元素的概率 $C_{s+b-1}^{w-1}/C_{s+b}^{w}=\frac{s+b}{w}$.所以对于是否有过期元素,这一步骤都具有相同的随机性.

与意义.

EDS 算法的理论基础主要围绕 EDS 结构的均匀性证明(定理 1)展开,这些理论构件为 SPF 算法的设计逻辑、操作流程以及性能分析提供了坚实的数学保障.

引理 1. 在任一时刻 t, 采样集合 S^+ 中的数据为 EDS 结构从当前滑动窗口中均匀随机选取的 $|s^+|$ 条数据.

证明.

1)当 $t\in[1,s+b]$ 时, 引理 1 显然成立, 即在 $t\in[1,s+b]$ 时, $Pr(S^+=\{e_1,e_2,\cdots,e_t\})=1$;

2)当 $t\in[s+b+1,w]$ 时, 对于 $t=s+b+1$ 来说, 基于算法 1 可知 e_{s+b+1} 将以 $\dfrac{s+b}{s+b+1}$ 的概率替换原来的采样中间集合 S^+ 中的一条记录, 从而在 $t=s+b+1$ 的时刻可知:

$$Pr(S^+=\{e_{s+b+1},e_2,\cdots,e_t\})=Pr(S^+=\{e_1,e_{s+b+1},\cdots,e_t\})=\cdots=Pr(S^+=\{e_1,e_2,\cdots,e_{s+b+1}\})=\frac{1}{s+b+1}=\frac{1}{\binom{s+b+1}{s+b}},\tag{1}$$

由此, 对于元素 e_{s+b+1} 来说, 它替换采样集合 S^+ 中的记录的概率为 $\dfrac{1}{s+b+1}$, 并且 e_{s+b+1} 没有替换的概率: $Pr(S^+=\{e_1,e_2,\cdots,e_t\})=\dfrac{s+b}{s+b+1}$, 从而 $t=s+b+1$ 时, 引理 1 也成立.

3)利用相似的分析方法可得, 当 $t>w$ 时, 引理 1 成立.

根据以上 3 点分析, 可以知道在任一时刻 t, 采样集合 S^+ 中的数据为从当前滑动窗口中均匀随机选取的 $|s^+|$ 条数据, 所以引理 1 成立. 证毕.

定理 1. 在任一时刻 t 来说, 令 W 为滑动窗口, w 为滑动窗口大小, 则 S 等于 W 的任何一个 S-子集的概率等于 $\dfrac{1}{C_s^w}$.

证明. 对于任一时刻 t, 对于 W 来说分成 2 种情况:

1) $W=\{e_1,e_2,\cdots,e_t\},\ (t<w)$

对于当前时刻 $t<w$ 来说, 可以知道当前滑动窗口中的元素为 $W=\{e_1,e_2,\cdots,e_t\}$. 由引理 1 可知, 采样集合 S^+ 中的 s^+ 条数据是从 W 中随机选取的 s^+ 条数据. 由此, 可知采样集合概率满足:

$$Pr(S^+=\text{Any }S^+\text{−subset in }W)=\frac{1}{C_{s^+}^w}.\tag{2}$$

另外, 由于算法 1 将 S^+ 中的 s^+-s 条数据随机删除. 由此, 由算法 1 可知采样集合概率满足:

$$Pr(S=\text{Any }S-\text{subset in }W)=\frac{1}{C_{s^+}^w}\frac{1}{C_{s^+-s}^{s^+}}\times C_{s^+-s}^{w-s}=\frac{1}{C_s^w}.\tag{3}$$

2) $W=\{e_{t-w+1},e_{t-w+2},\cdots,e_t\},\ (t\geqslant w)$

对于当前时刻 $t\geqslant w$ 来说, 可以知道当前滑动窗口中的元素为 $W=\{e_{t-w+1},e_{t-w+2},\cdots,e_t\}$. 同样, 由引理 1 可知, 采样集合 S^+ 中的 $s^+=s+b$ 条数据是从 W 中随机选取的 $s+b$ 条数据, 从而

$$Pr(S^+=\text{Any }(s+b)-\text{subset in }W)=\frac{1}{C_{s^+}^w}.$$

另外, 由算法 1 随机删除采样集合 S^+ 中 b 条数据来生成采样集合 S. 由此, 知道采样集合概率:

$$Pr(S=\text{Any }S-\text{subset in }W)=\frac{1}{C_{s+b}^w}\frac{1}{C_s^{s+b}}\times C_b^{w-s}=\frac{1}{C_s^w}.\tag{4}$$

根据情况 1 和情况 2 所满足的采样集合概率, 即式(3)(4)可以知道定理 1 成立.

在所设计的数据结构 EDS 中, 实际上使用了采样集合 s 中的区间个数 $\hat{\eta}$, 并使用 $\hat{\eta}$ 来估计真实计数值 η, 其中 $\hat{\eta}\times\dfrac{w}{s}$ 表示真实计数值 η 的估计结果. 证毕.

定理 2. 数据结构 EDS 中的 $E\left(\hat{\eta}\times\dfrac{w}{s}\right)=\eta$.

证明. 对于数据结构 EDS 来说, 可知 $E(\hat{\eta})=\dfrac{s\eta}{w}$. 根据引理 1 可知数据结构 EDS 对于每个数据点被选择的概率是相等的. 因此其估计通常是无偏估计的. 可知 $E\left(\hat{\eta}\times\dfrac{w}{s}\right)=\dfrac{s\eta}{w}\times\dfrac{w}{s}=\eta$. 所以数据结构 EDS 中的 $E\left(\hat{\eta}\times\dfrac{w}{s}\right)=\eta$. 证毕.

定理 3. 通过数据结构 EDS 所生成的采样计数值与当前窗口内的真实计数值之间的方差为

$$\frac{\eta(w-\eta)(w-\mathrm{s})}{(w-1)s}.$$

证明. 对于任何给定的区间间隔 I 来说, 当前滑动窗口中的 w 个元素可以表示为二进制位字符串. 那么可以说明当 $e_i\in I_i$, 则对应的位为 1, 否则为 0, 判断该元素是否满足当前区间.

在不失一般性的情况下, 假设对于某个直方图间隔 I, 当前滑动窗口中属于当前区间的数量为 η, 不属于当前区间的数量为 $w-\eta$, 其中 η 表示当前滑动窗口中的真实计数值.

$\hat{\eta}$ 满足分布:

$$P(\hat{\eta}=m|\eta)=\frac{\mathrm{C}_m^{\eta}\mathrm{C}_{s-m}^{w-\eta}}{\mathrm{C}_s^w},\tag{5}$$

其中, m 表示当前滑动窗口内区间的估计计数结果, w 表示滑动窗口大小, s 表示采样集合 S 的长度.

为了证明数据结构 EDS 所生成的采样计数值与当前窗口内的真实计数值之间的方差, 公式推导过程如下所示:

首先, η 的方差 $D(\eta)=0$, 这是因为当滑动窗口大

小给定时, η是一个常数值, 而不是一个随机变量.

$$D\left(\hat{\eta}\times\frac{w}{s}-\eta\right)=D\left(\hat{\eta}\times\frac{w}{s}\right)=\left(\frac{w}{s}\right)^2D(\hat{\eta})=\left(\frac{w}{s}\right)^2\frac{s\frac{\eta}{w}\left(1-\frac{\eta}{w}\right)(w-s)}{w-1}=\frac{w^2}{s^2}\frac{s\frac{\eta}{w}\left(1-\frac{\eta}{w}\right)(w-s)}{w-1}=\frac{\eta(w-\eta)}{w-1}\frac{w-s}{s}. \tag{6}$$

根据式(6), 可以得到$D\left(\hat{\eta}\times\frac{w}{s}-\eta\right)=\frac{\eta(w-\eta)}{w-1}\frac{w-s}{s}$. 当滑动窗口大小给定时, 式(6)中的$\frac{\eta(w-\eta)}{w-1}$是一个常数值. 证毕.

为了确定方程的单调性, 现在只需要确定$\frac{w-s}{s}$函数的单调性, 因此对$f(s)=\frac{w-s}{s}$进行求导运算:

$$f'(s)=\frac{-s-(w-s)}{s^2}=\frac{-s-w+s}{s^2}=-\frac{w}{s^2}. \tag{7}$$

因为式(7)结果小于0, 可知式(6)随着采样集合的变大而方差逐渐减小. 当滑动窗口大小w等于采样集合S的长度s时, 说明对滑动窗口内所有数据都进行了采集, 估计值与真实值相同, 所以数据结构EDS的方差为0.

令$\hat{\eta}$表示属于某个区间I的数据个数, 数据结构EDS从最近的w个元素中随机选取s个元素并统计满足当前区间的个数, 下面证明$\hat{\eta}$近似服从正态分布.

引理 2. $\hat{\eta}$近似服从正态分布$N\left(\frac{s\eta}{w},\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\cdot\frac{w-s}{w-1}\right)$, 其中$\eta$表示当前窗口中属于区间$I$的计数个数, s表示采样集合的大小, w为滑动窗口大小.

证明. 对于区间I, 当前滑动窗口中的w个元素可以转化为w个数, 其中0表示对应的数据不属于I, 1表示对应的数据属于I; 由于数据结构EDS是从w个元素中均匀选择s个元素, 因此从式(1)可知$\hat{\eta}$等于m的概率.

由于数据流上的滑动窗口大小w一般会设置一个较大的值, 这是因为对于一个较小的w, 内存可以有条件地保存所有的窗口记录, 因此不需要进行特殊的内存设计, 以实现高效的数据流算法[18,19].

根据正态分布可得到方程:

$$\frac{C_m^{\eta}C_{s-m}^{w-\eta}}{C_s^w}\approx\frac{1}{\sqrt{2\Pi}h}e^{-\frac{1}{2}\left(\frac{m-\frac{s\eta}{w}}{h}\right)^2}. \tag{8}$$

$$h=\sqrt{\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}}$$

根据式(5)(8)显然可知: 数据结构EDS中的$\hat{\eta}$近似服从正态分布$N\left(\frac{s\eta}{w},\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$. 证毕.

定理 4. 数据结构EDS近似服从正态分布$N\left(0,\frac{w^2}{s^2}\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$.

证明. 根据引理2可知, $\hat{\eta}$近似服从正态分布$N\left(\frac{s\eta}{w},\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$. 那么, $\left(\frac{w}{s}\hat{\eta}-\eta\right)$近似服从正态分布$N\left(0,\frac{w^2}{s^2}\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$, 数据结构EDS近似服从正态分布$N\left(0,\frac{w^2}{s^2}\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$. 证毕.

4.2 SPF算法的隐私性

定理 5. SPF算法满足(ε,δ)-差分隐私定义.

证明. SPF算法主要是由2部分组成: 高效数据流采样草图结构和基于高效数据流采样的自适应加噪算法. 这2部分都需要满足差分隐私算法.

1)情况1. 采样集合大小小于等于采样集合最大大小时.

① 高效数据流采样草图结构

EDS直接反映出滑动窗口内数据结果, 采样集合实际上是包含了窗口内的真实数据, 则EDS对应的隐私预算$\varepsilon_1=0$, 不需要添加高斯噪声.

② 基于高效数据流采样的自适应加噪算法

基于高效数据流采样的自适应加噪算法, 会根据当前隐私预算$\varepsilon_2=\varepsilon$直接添加相应的高斯噪声.

根据定义7, 该隐私预算将满足所有区间的隐私保护要求, 那么SPF算法满足$(0+\varepsilon,\delta)$-差分隐私, 即SPF算法满足(ε,δ)-差分隐私定义.

2)情况2. 采样集合大小大于采样集合最大值时.

① 高效数据流采样草图结构

根据定理4可知, 给定一个滑动窗口大小w和一个采样集合s, EDS的$\left(\frac{w}{s}\hat{\eta}-\eta\right)$近似服从正态分布$N\left(0,\frac{w^2}{s^2}\frac{s\eta}{w}\left(1-\frac{\eta}{w}\right)\frac{w-s}{w-1}\right)$. 加入的高斯噪声的标准差$\sigma$通常是$\sigma=\Delta f\frac{\sqrt{2\ln(1.25/\delta)}}{\varepsilon_1}$计算得到. 给定高斯噪声的标准差$\sigma$, 正态分布$N(\mu,\sigma^2)$表示为$N\left(0,\frac{2\ln(1.25/\delta)}{\varepsilon_1^2}\right)$.

根据高斯分布所产生的误差与EDS所产生的误差相等, 得到:

$$\varepsilon_1=\sqrt{\frac{2\ln\left(\frac{1.25}{\delta}\right)s(w-1)}{\eta(w-\eta)(w-s)}}, \tag{9}$$

其中, ε_1的取值主要由窗口大小w和采样集合大小s决定.

由此, 草图结构等价于每个区间都分配与隐私预算为ε_1相同的高斯噪声所量化的噪声.

② 基于高效数据流采样的自适应加噪算法

该算法通过动态调整噪声值，使滑动窗口采样的估计误差与差分隐私噪声值之和能够满足用户的差分隐私需求. 具体来说，算法会根据数据流的特点和用户设定的隐私参数，精确地控制噪声的添加量.

由此，基于高效数据流采样的自适应加噪算法等价于在每个时间区间内分配与隐私预算为ε_2相同的高斯噪声.

根据定义 7，该隐私预算将满足所有区间的隐私保护要求，那么 SPF 算法满足$(\varepsilon_1+\varepsilon_2,\delta)$-差分隐私，即 SPF 算法满足$(\varepsilon,\delta)$-差分隐私定义.

根据情况 1 和情况 2，SPF 算法满足(ε,δ)-差分隐私定义. 证毕.

4.3 SPF 算法的空间代价和时间代价

本节对 SPF 算法的时间和空间代价进行了深入的分析.

定理 6. 基于差分隐私的数据流采样发布快速生成算法 SPF 的空间复杂度为$O(s)$.

证明. SPF 算法由 2 个主要部分组成：高效数据流采样草图结构 EDS 和基于高效数据流采样的自适应加噪算法. 这 2 个算法分别实现了对滑动窗口数据的快速采集和隐私保护.

1) 高效数据流采样草图结构 EDS

EDS 用于对滑动窗口数据进行快速采集，提取关键特征. 这一过程中，EDS 会构建一个包含所有可能的采样中间集合s^+.

采样集合s是从采样中间集合中选取的子集，因此 EDS 的空间开销主要由中间集合s^+的大小决定.

2) 基于高效数据流采样的自适应加噪算法

该算法根据隐私预算比率分配比例来进行加噪，从而保证数据发布的差分隐私特性. 这一步并不会显著增加空间开销，因为加噪是基于采样集合s进行的.

综上所述，通过详细分析 EDS 和基于高效数据流采样的自适应加噪算法的空间开销，可以得出 SPF 算法的空间复杂度为$O(s)$. 证毕.

计算 SPF 算法的时间代价，首先我们需要计算 EDS 所需要的时间开销，然后计算 SPF 算法的时间开销.

定理 7. EDS 消耗的时间复杂度为$O(s)$.

证明. EDS 的时间复杂度主要由时间戳和计数值记录的操作、过期数据的检查和删除操作，以及随机的样本替换操作组成. 为了证明 EDS 的时间复杂度，需要详细分析其每一步的操作及其对应的时间消耗.

1)时间戳和计数值记录的操作. 每一个时刻只有 1 个数据进入$O(1)$.

2)过期数据的检查和删除操作. 样本集合中最多纪录$s+b$条数据记录，并且这种操作最多需要遍历这$s+b$条记录 1 次$O(s)(s \gg b)$.

3 随机的样本替换操作. 不存在过期数据，直接随机替换$O(1)$;

将上述 3 个步骤的时间复杂度综合考虑，EDS 的时间复杂度为$O(s)$. 证毕.

定理 8. 已知采样中间集合的大小为$s+b$，则 SPF 算法消耗的时间开销为$O(s)$.

证明. 对于任意的滑动窗口，SPF 算法的处理过程主要分为 2 个部分：

1)高效数据流采样算法 EDS. 构建一个包含所有可能采样数据的中间集合，这一步的时间复杂度为$O(s)$，由定理 7 可知.

2)基于高效数据流采样的自适应加噪算法. 基于采样集合进行加噪，加噪过程的时间开销通常较小.

通过对这 2 个部分的分析，时间开销最主要的主体在于通过高效数据流采样草图结构 EDS 获得采样集合时所需要的处理时间. 因此，对于 SPF 算法的时间复杂度为$O(s)$. 证毕.

5 实验分析

5.1 实验设置

本实验的硬件环境为 Inter Core i7-13700KF 8-core3.4 GHz, 16 G RAM, 1 TB 硬盘内存存储，软件环境为 Windows 11 操作系统，MatlabR2021b 编程语言.

本文使用了在差分隐私中广泛使用的数据集[21,23]Taxi 和 Traffic 进行实验.

1)Taxi 数据集是一个黄色的出租车旅行记录数据集，包括捕获接送日期/时间、司机报告的乘客数量等，由纽约市出租车和豪华轿车委员会网站提供，选择了 2019 年 1 月至 2019 年 12 月的纽约市黄色出租车出行数据集作为实验数据集.

2)Traffic 数据集是一个详细的道路安全数据集，关于人身伤害的情况、道路事故由英国警方提供，他们收集每一辆汽车碰撞的数据. 选取了从 2005—2014 年的伤亡者档案中包含年龄、性别等信息的数据集作为实验数据集.

对于这 2 个真实数据集，使用了 MatlabR2021b 编程语言实现 SPF 算法、HPSSGC 算法[31]、MetaAlgo 算法[33] 和 MGP 算法[35]. 所有算法的算法复杂度对比

如表 2 所示[①]. 将本文 SPF 算法与其他 3 种算法进行了实验比较, 并对实验结果进行了验证.

Table 2 Comparison of Algorithm Complexity

表 2 算法复杂度对比

算法	时间复杂度	空间复杂度
HPSSGC	$O(w\log w)$	$O(w)$
MetaAlgo	$O(w+l)$	$O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}+w\right)$
MGP	$O(w)$	$O\left(\frac{1}{\varepsilon}+w\right)$
SPF (本文)	$O(s)$	$O(s)$

5.2 性能评估标准

本实验主要比较了平均内存使用量、平均运行时间和数据的准确性. 平均内存使用量是滑动窗口发布一个直方图所需要的内存空间. 平均运行时间是运行算法程序在每一时刻发布数据流数据的直方图所需的平均时间. 数据的准确性是采用均方根误差(root mean square error, *RMSE*)来评价数据的精度.

RMSE 的计算公式为

$$RMSE=\sqrt{\frac{\sum_{i=1}^{l}(H_i-\bar{H}_i)^2}{1}},\tag{10}$$

其中, i表示第i个区间, H_i表示第i个区间间隔查询所对应的真实计数值, $\bar{H}_i$表示当前区间间隔发布的噪声值.

5.3 SPF 算法的隐私预算分配设置

为了测试 SPF 算法中隐私预算在不同的分配比例下对可用性产生怎样的影响. 对于 2 种不同数据集, 固定滑动窗口和分块的大小, 将β设置范围为 (0, 1). 将采用式(1)(2)计算总误差.

对于某一个区间计算来说, 总误差的构成为:

1)在高效数据流采样算法中, 该算法等效于差分隐私中隐私预算为ε_1的噪声;

2)在基于高效数据流采样草图结构的自适应加噪算法中, 在该算法中使用了隐私预算为ε_2的噪声.

总误差的计算公式为:

$$error=\frac{2\ln(1.25/\delta)}{\varepsilon_1^2}+\frac{2\ln(1.25/\delta)}{\varepsilon_2^2}.\tag{11}$$

为了满足数据的可用性, 对于总误差来说, 需要使总误差取最小值, 即考察下面的最小化问题:

$$error_{\min}=2\ln(1.25/\delta)\times\left(\frac{1}{\varepsilon_1^2}+\frac{1}{\varepsilon_2^2}\right).\tag{12}$$

需要将$\varepsilon_1=\beta\varepsilon$, $\varepsilon_2=(1-\beta)\varepsilon$带入$error_{\min}$获得式(13).

$$error_{\min}=\left(\frac{1}{\beta^2}+\frac{1}{(1-\beta)^2}\right)\times\frac{2\ln(1.25/\delta)}{\varepsilon^2}.\tag{13}$$

为了取当前总误差的最小值(达到最好的数据可用性), 只需要取式(13)的$\frac{1}{\beta^2}+\frac{1}{(1-\beta)^2}$的最小值即可. 式(13)可以获得式(14).

$$f(x)=\frac{1}{\beta}+\frac{1}{1-\beta}.\tag{14}$$

首先, 需要对式(14)进行求导, 求导结果如式(15)所示.

$$f'(x)=-\frac{2}{x^3}+\frac{2}{(1-x)^3}.\tag{15}$$

然后, 令$f'(x)=0$, 即$-\frac{2}{x^3}+\frac{2}{(1-x)^3}=0$, 可得

$$1-2x=0.\tag{16}$$

解得式(16)的结果为$x=0.5$. 为了验证$x=0.5$是最小值, $f(x)$的二阶导数当$x=0.5$结果为$192>0$.给定函数$f(x)$可以在$x=0.5$获得最小值. 所以, 最后对应的结果为$x=0.5$.

当隐私预算比例β在 0.1~0.5 之间时, 随着隐私预算比例β的增加, ε_1增加, ε_2减少, 对于总误差来说, 总误差在慢慢减少; 当隐私预算比例β在 0.5~1 之间时, 随着隐私预算比例β的继续增加, ε_1减少, ε_2增加, 对于总误差来说, 总误差在慢慢增加. 因此, 在后面的实验中, SPF 算法中的隐私预算分配比例$\beta=0.5$.

5.4 实验结果分析

本文分析了在 2 种不同的真实数据集下的算法性能.

5.4.1 不同滑动窗口下的算法性能

对于 4 种算法, 改变滑动窗口大小 w, 并在ε=0.1 的条件下测量平均运行时间、平均内存使用量和 *RMSE*, 结果如图 2、图 3 所示.

① 空间复杂度分析: HPSSGC 算法需要缓存整个窗口内的数据, 因此空间复杂度为$O(w)$. 另外, 如果采样进行选择替换的数据集合大小为w和聚类数量为l, 则总的空间复杂度为$O(w+w+l)=O(2w+l)=O(w)$. MetaAlgo 算法空间复杂度涉及隐私预算参数ε和差分隐私参数δ, 以及窗口大小 w, 空间复杂度为$O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}+w\right)$. MGP 算法的空间复杂度主要由隐私预算参数$\varepsilon$和窗口大小 w 决定, 空间复杂度为$O\left(w+\frac{1}{\varepsilon}\right)$. 时间复杂度分析: HPSSGC 算法主要包括数据采样、数据重新排序和贪心聚类, 所有的时间复杂度为$O(w\log w)$. MetaAlgo 算法采用了快速选择算法, 每次滑动窗口操作的时间复杂度主要在于更新直方图, 则时间复杂度为$O(w+l)$. MGP 算法采用了 MG(Misra-Gries)算法, 每次滑动窗口操作的时间复杂度主要在于更新直方图, 则时间复杂度为$O(w)$.

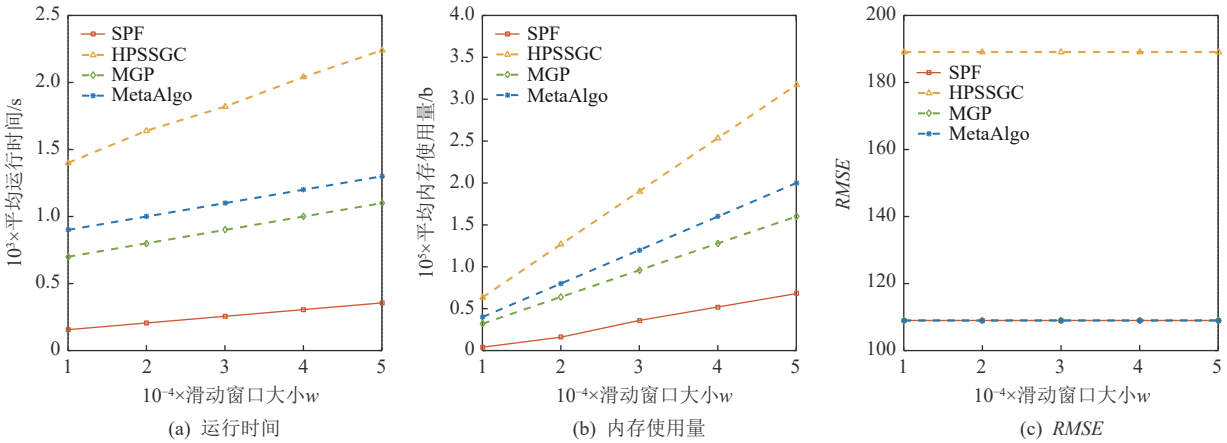


Fig. 2 Performance on Taxi dataset with different sliding window sizes

图 2 不同滑动窗口大小的 Taxi 数据集上的性能

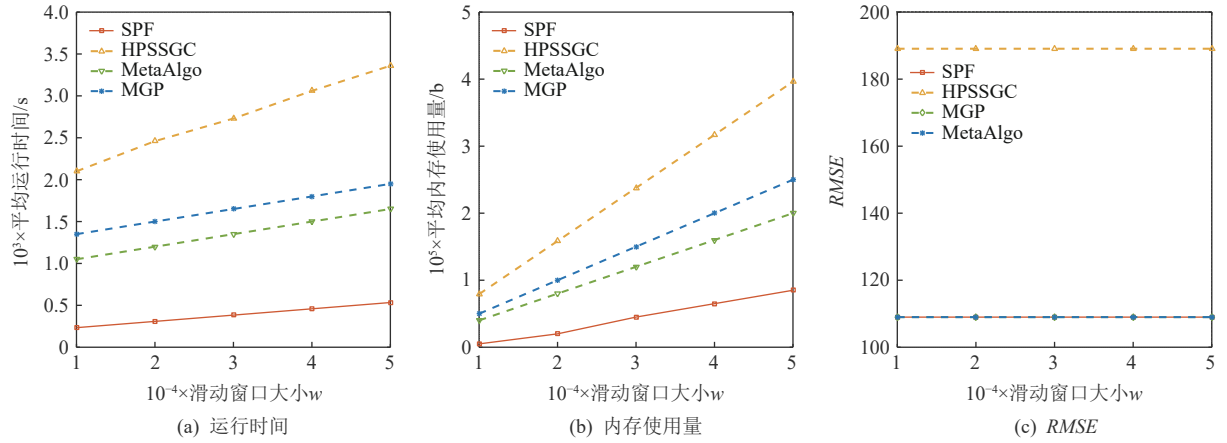


Fig. 3 Performance on Traffic dataset with different sliding window sizes

图 3 不同滑动窗口大小的 Traffic 数据集上的性能

在图 3 中，当 w = 30 000 时，SPF 的平均运行时间，平均内存使用和 RMSE 分别为 2.55×10^{-4} s, 4.5×10^4 bit, 108. 相比之下，HPSSGC, MetaAlgo 和 MGP 需要的平均运行时间分别为 2.72×10^{-3} s, 1.65×10^{-3} s, 1.2×10^{-3} s; HPSSGC, MetaAlgo 和 MGP 需要的平均内存使用量分别为 2.37×10^5 bit, 1.5×10^5 bit, 1.2×10^5 bit; HPSSGC, MetaAlgo 和 MGP 需要的 RMSE 分别为 189, 108, 108.

从图 2 和图 3 中可以得出 2 点结论：

1）即使在滑动窗口大小 w 较小时，SPF 依然比 HPSSGC, MetaAlgo 和 MGP 展现出更快的运行时间和更低的内存使用量. 这一观察结果突显了滑动窗口估计计数草图和基于 EDS 的自适应直方图生成算法的固有效率. 随着滑动窗口的变化，需要采集的样本集合大小会增大，因为为了保证相同的数据精准度，必须采集更多的样本以保持一致.

2）随着滑动窗口大小 w 的增加，所有方法的误差保持不变. 出现这种现象的原因是，在固定的隐私预算情况下，这些方法利用统计数据并随后引入噪声，导致误差水平持续存在. 隐私预算决定了数据流的差分隐私保护程度，而滑动窗口大小的调整对数据效用的影响可以忽略不计.

总之，与其他算法相比，所提出的 SPF 在不同滑动窗口中平均运行时间显著降低了 63%，平均内存使用量显著降低了 50%.

5.4.2 不同隐私预算下的算法性能

在不同隐私预算 ε 下评估了 4 种算法在 2 个真实数据集上的性能，如图 4 和图 5 所示. 固定了 w = 50 000，隐私预算设置 ε 从 0.02 ~0.1 不等.

图 5 中，当 ε = 0.06时，SPF 的平均运行时间、平均内存使用量和 *RMSE* 分别为 2.23×10^{-4} s, 5.75×10^4 b, 181. 相比之下，HPSSGC, MetaAlgo 和 MGP 需要的平均运行时间分别为 2.2×10^{-3} s, 1.2×10^{-3} s, 1.4×10^{-3} s; HPSSGC, MetaAlgo 和 MGP 需要的平均内存使用量

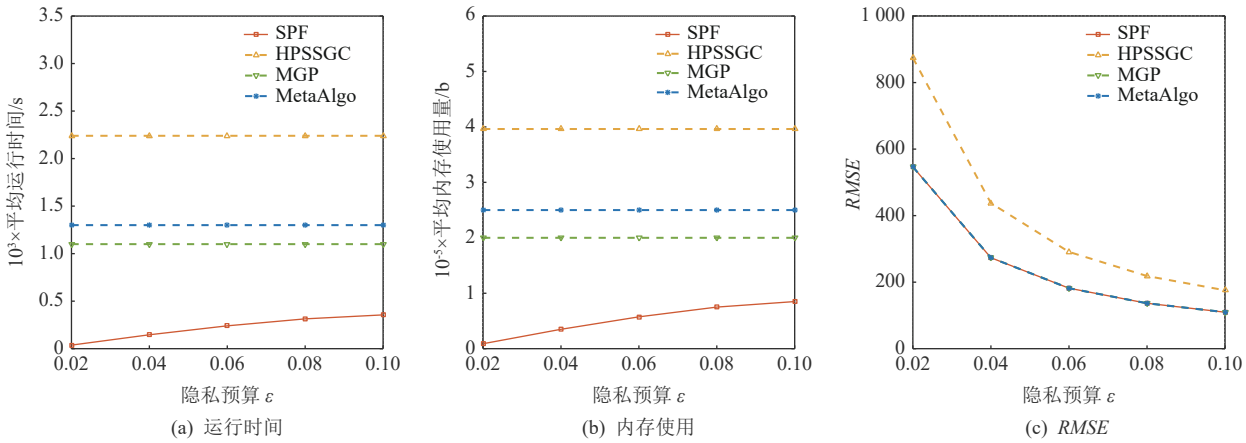


Fig. 4 Performance on Taxi dataset with different privacy budgets

图 4 不同隐私预算的 Taxi 数据集上的性能

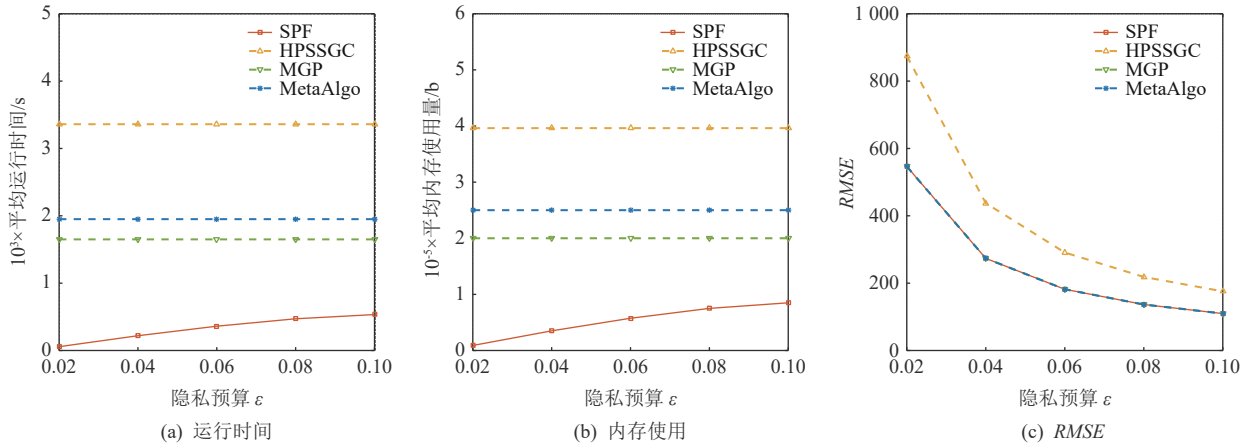


Fig. 5 Performance on Traffic dataset with different privacy budgets

图 5 不同隐私预算的 Traffic 数据集上的性能

分别为 3.95×10^5 b, 2.5×10^5 b, 2×10^5 b; HPSSGC, MetaAlgo 和 MGP 需要的 RMSE 分别为 290, 181, 181.

从图 4 和图 5 中可以得出 3 点结论:

1)更大的隐私预算会导致 SPF 算法的运行时间和内存使用量的增加. 这种现象与滑动窗口估计计数草图和基于 EDS 的自适应直方图生成算法的理论有效性一致. 随着隐私预算的增大, 需要采集的样本集合也会变大, 因为隐私预算越大, 数据值越准确, 因此需要采集的数据更多. 这意味着, 随着隐私预算的增加, 算法需要处理更多的数据和计算, 从而增加了资源消耗.

2)随着隐私预算的增加, *RMSE* 会降低, 表明数据效用更高, 但隐私保护水平降低. 这一观察结果与实用性和隐私性之间的权衡模式一致. 也就是说, 在较高隐私预算下, 尽管数据的准确性提高, 但泄露敏感信息的风险也相应增加.

3)最重要的是, SPF 在运行时间和内存使用量方面始终优于其他方法, 同时保持相似的数据效用. 这显示了 SPF 算法在实际应用中的高效性和竞争力, 使其成为在资源受限环境中进行数据分析的优选方案.

总之, 与其他算法相比, 在不同的隐私预算下, SPF 算法平均运行时间显著减少了 65%, 平均内存使用量显著减少了 69%.

6 总结与展望

本文的研究提出了一种基于差分隐私的数据流采样发布快速生成算法 SPF, 旨在实现差分隐私下的实时数据流发布. 该算法包含 2 个关键创新点: 高效数据流采样算法和基于高效数据流采样的自适应加噪算法. 首先, 为了降低时间和空间开销, 提出一种新颖的内存高效的数据流采样草图结构. 该数据结构可以快速对数据流进行统计, 从而更快地获取统计值, 并且可以满足量化的差分隐私, 保证(ε,δ)-差分

隐私条件，提供严格可控的数据保护能力. 然后，为了满足用户所提供的隐私保护强度，并且避免正确反映原始数据流的真实情况，提出一种基于高效数据流采样的自适应加噪算法. 该算法通过对高效数据流采样算法的误差和差分隐私误差进行量化，并在数据发布过程中添加差分隐私噪声，以达到用户所需的隐私保护强度. 在2个真实数据集上进行了实验验证，结果表明SPF算法能够快速处理数据流，并根据用户要求自适应地添加噪声，以发布满足差分隐私保护要求的直方图数据.

SPF算法对于滑动窗口差分隐私直方图的快速发布很有用，但仍存在一些局限性. 遵循先前的工作假设每个区间结果具有相同的可靠性程度. 然而，在实际情况中，不同区间的结果可能具有不同的隐私属性. 因此，未来将计划扩展SPF算法提供个性化差分隐私的机制.

作者贡献声明：王修君提出并设计了算法方案和实验方案，并进行论文修改；莫磊完成实验部分并撰写论文；郑啸、卫琳娜、董俊、刘志检验算法的性能，提出指导意见，并参与论文修改；郭龙坤把握论文的总体创新性，验证理论分析和实验分析的正确性，并负责论文的最终修订.

参 考 文 献

[1] Dong Haowen, Zhang Chao, Li Guoliang, et al. Survey on cloudnative databases[J]. Journal of Software, 2023, 35(2): 899−926 (in Chinese)
(董昊文, 张超, 李国良, 等. 云原生数据库综述[J]. 软件学报, 2023, 35(2): 899−926)

[2] Zhao Zhanhao, Pan Hexiang, Chen Gang, et al. VeriTxn: Verifiable transactions for cloud-native databases with storage disaggregation[J]. Proceedings of the ACM on Management of Data, 2023, 1(4): 1−27

[3] Papadogiannaki E, Ioannidis S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures[J]. ACM Computing Surveys, 2021, 54(6): 1−35

[4] Shahraki A, Taherkordi A, Haugen Ø. TONTA: Trend-based online network traffic analysis in ad-hoc IoT networks[J]. Computer Networks, 2021, 194: 108125

[5] Shahid M R, Blanc G, Zhang Z, et al. IoT devices recognition through network traffic analysis[C]//Proc of 2018 IEEE Int Conf on Big Data (Big Data). Piscataway, NJ: IEEE, 2018: 5187−5192

[6] Butilă E V, Boboc R G. Urban traffic monitoring and analysis using unmanned aerial vehicles (UAVs): A systematic literature review[J]. Remote Sensing, 2022, 14(3): 620

[7] Jain N K, Saini R K, Mittal P. A review on traffic monitoring system techniques. Soft Computing: Theories and Applications[C]//Proc of SOCTA 2017, 2019: 569−577

[8] Figueiras P, Herga Z, Guerreiro G, et al. Real-time monitoring of road traffic using data stream mining[C]//Proc of 2018 IEEE Int Conf on Engineering, Technology and Innovation (ICE/ITMC). Piscataway, NJ: IEEE, 2018: 1−8

[9] Fang B, Zhang P. Big data in finance[J]. Big Data Concepts, Theories, and Applications, 2016: 391−412

[10] Fikri N, Rida M, Abghour N, et al. An adaptive and real-time based architecture for financial data integration[J]. Journal of Big Data, 2019, 6: 1−25

[11] Thennakoon A, Bhagyani C, Premadasa S, et al. Real-time credit card fraud detection using machine learning[C]//Proc of 2019 9th Int Conf on Cloud Computing, Data Science & Engineering (Confluence). Piscataway, NJ: IEEE, 2019: 488−493

[12] Upadhyay J. Sublinear space private algorithms under the sliding window model[C]//Proc of Int Conf on Machine Learning. New York: PMLR, 2019: 6363−6372

[13] Bassily R, Nissim K, Stemmer U, et al. Practical locally private heavy hitters[J]. The Journal of Machine Learning Research, 2020, 21(1): 535−576

[14] Li F. Cloud-native database systems at Alibaba: Opportunities and challenges[J]. Proceedings of the VLDB Endowment, 2019, 12(12): 2263−2272

[15] Li Haixiang, Li Xiaoyan, Liu Chang, et al. Systematic definition and classification of data anomalies in data base management systems. Journal of Software, 2022, 33(3): 909−930 (in Chinese)
(李海翔, 李晓燕, 刘畅, 等. 数据库管理系统中数据异常体系化定义与分类. 软件学报, 2022, 33(3): 909–930)

[16] Zhao Hongyao, Zhao Zhanhao, Yang Wanqing, et al. Experimental study on concurrency control algorithms in in-memory databases. Journal of Software, 2022, 33(3): 867−890 (in Chinese)
(赵泓尧, 赵展浩, 杨皖晴, 等. 内存数据库并发控制算法的实验研究. 软件学报, 2022, 33(3): 867–890)

[17] Spillner J, Toffetti G, López M R. Cloud-native databases: An application perspective[C]//Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2017. Berlin: Springer, 2018: 102−116

[18] Dwork C. Differential privacy[C]//Proc of Int Colloquium on Automata, Languages, and Programming. Berlin: Springer, 2006: 1−12

[19] Shahin V, Zhang Xinyao, Qiu Dongyu. Analysis and optimization of big-data stream processing[C]//Proc of 2016 IEEE Global Communications Conf (GLOBECOM). Piscataway, NJ: IEEE, 2016. Doi: 10.1109/GLOCOM.2016.7841598

[20] Bar-Yossef Z, Jayram T S, Kumar R, et al. Counting distinct elements in a data stream[C]//Proc of Int Workshop on Randomization and Approximation Techniques in Computer Science. Berlin: Springer, 2002: 1−10

[21] Danassis P, Triastcyn A, Faltings B. A distributed differentially private algorithm for resource allocation in unboundedly large settings[J]. arXiv preprint, arXiv: 2011.07934, 2020

[22] Mazmudar M, Humphries T, Liu J, et al. Cache me if you can: Accuracy-aware inference engine for differentially private data

exploration[J]. arXiv preprint, arXiv: 2211.15732, 2022

[23] Chen Y, Al-Rubaye S, Tsourdos A, et al. Differentially-private federated intrusion detection via knowledge distillation in third-party IoT systems of smart airports[C]//Proc of IEEE Int Conf on Communications (ICC 2023). Piscataway, NJ: IEEE, 2023: 603−608

[24] Xu Jia, Zhang Zhejie, Xiao Xiaokui, et al. Differentially private histogram publication[J]. The VLDB Journal, 2013, 22(6): 797−822

[25] Zhang X, Chen R, Xu J, et al. Towards accurate histogram publication under differential privacy[C]//Proc of the 2014 SIAM Int Conf on Data Mining. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014: 587−595

[26] Zhang Xiaojian, Shao Chao, Meng Xiaofeng. Accurate histogram release under differential privacy[J]. Journal of Computer Research and Development, 2016, 53(5): 1106−1117 (in Chinese)
(张啸剑, 邵超, 孟小峰. 差分隐私下一种精确直方图发布方法[J]. 计算机研究与发展, 2016, 53(5): 1106−1117)

[27] Tang Haixia, Yang Geng, Bai Yunlu. Histogram publishing algorithm based on adaptive privacy budget allocation strategy under differential privacy[J]. Application Research of Computers, 2020, 53(7): 1952−1957, 1963 (in Chinese)
(唐海霞, 杨庚, 白云璐. 自适应差分隐私预算分配策略的直方图发布算法[J]. 计算机应用研究, 2020, 53(7): 1952−1957, 1963)

[28] Lin Fupeng, Wu Yingjie, Wang Yilei, et al. Differentially private statistical publication for two-dimensional data stream[J]. Journal of Computer Applications, 2015, 35(1): 88−92 (in Chinese)
(林富鹏, 吴英杰, 王一蕾, 等. 差分隐私二维数据流统计发布[J]. 计算机应用, 2015, 35(1): 88−92)

[29] Zhang Xiaojian, Meng Xiaofeng. Streaming histogram publication method with differential privacy[J]. Journal of Software, 2016, 27(2): 381−393 (in Chinese)
(张啸剑, 孟小峰. 基于差分隐私的流式直方图发布方法[J]. 软件学报, 2016, 27(2): 381−393)

[30] Sun L, Ge C, Huang X, et al. Differentially private real-time streaming data publication based on sliding window under exponential decay[J]. Computers, Materials and Continua, 2019, 58(1): 61−78

[31] Wu X, Tong N, Ye Z, et al. Histogram publishing algorithm based on sampling sorting and greedy clustering[C]//Proc of Int Conf on Blockchain and Trustworthy Systems. Berlin: Springer, 2020: 81−91

[32] Liu X, Liu H. Data publication based on differential privacy In V2G network[J]. International Journal of Electronics Engineering and Applications, 2021, 9(2): 34−44

[33] Cardoso A R, Rogers R. Differentially private histograms under continual observation: Streaming selection into the unknown[C]//Proc of Int Conf on Artificial Intelligence and Statistics. New York: PMLR, 2022: 2397−2419

[34] Cao X, Cao Y, Pappachan P, et al. Differentially private streaming data release under temporal correlations via post-processing[C]//Proc of IFIP Annual Conf on Data and Applications Security and Privacy. Cham: Springer, 2023: 184−200

[35] Lebeda C J, Tetek J. Better differentially private approximate histograms and heavy hitters using the Misra-Gries sketch[C]//Proc of the 42nd ACM SIGMOD-SIGACT-SIGAI Symp on Principles of Database Systems. New York: ACM, 2023: 79−88

[36] Streeter M, Golovin D. An online algorithm for maximizing submodular functions[J]. Advances in Neural Information Processing Systems, 2008, 21

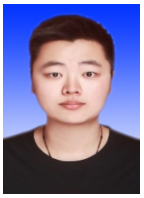
[37] Luo Ziyue, Wu Chuan. An online algorithm for VNF service chain scaling in datacenters[J]. IEEE/ACM Transactions on Networking, 2020, 28(3): 1061−1073

[38] Misra J, Gries D. Finding repeated elements[J]. Science of Computer Programming, 1982, 2(2): 143−152

[39] Chan T H H, Li M, Shi E, et al. Differentially private continual monitoring of heavy hitters from distributed streams[C]//Proc of the 12th Int Symp on Privacy Enhancing Technologies (PETS 2012), Berlin: Springer, 2012: 140−159

Wang Xiujun, born in 1983. PhD, associate professor. His main research interests include data streams and RFID systems.

王修君, 1983 年生. 博士, 副教授. 主要研究方向为数据流、RFID 系统.



Mo Lei, born in 1995. PhD candidate. His main research interest includes data stream differential privacy.

莫　磊, 1995 年生. 博士研究生. 主要研究方向为数据流差分隐私.

Zheng Xiao, born in 1975. PhD, professor. Senior member of CCF. His main research interests include computer network, industrial Internet, cloud computing and service computing, machine learning, and privacy protection.

郑　啸, 1975 年生. 博士, 教授. CCF 高级会员. 主要研究方向为计算机网络、工业互联网、云计算与服务计算、机器学习、隐私保护.

Wei Linna, born in 1984. PhD. Her main research interests include computer networks and wireless networks.

卫琳娜, 1984 年生. 博士. 主要研究方向为计算机网络、无线网络.

Dong Jun, born in 1973. PhD, associate research fellow. Member of CCF. His main research interests include machine vision, information networks, and applications of the agricultural Internet of things.

董　俊, 1973 年生. 博士, 副研究员. CCF 会员. 主要研究方向为机器视觉、信息网络、农业物联网应用.

Liu Zhi, born in 1986. PhD, associate professor. Senior member of IEEE. His main research interests include video streaming, mobile edge computing, and wireless networking.

刘　志, 1986 年生. 博士, 副教授. IEEE 高级会员. 主要研究方向为视频流、移动边缘计算、无线网络.

Guo Longkun, born in 1983. PhD, professor. His main research interests include data science and computer networks.

郭龙坤, 1983 年生. 博士, 教授. 主要研究方向为数据科学、计算机网络.