面向数据流滑动窗口的自适应直方图发布算法

王修君，莫磊，郑啸，高云全

引用本文

王修君，莫磊，郑啸，高云全. 面向数据流滑动窗口的自适应直方图发布算法[J]. 计算机科学, 2022, 49(10): 344-352.

WANG Xiu-jun, MO Lei, ZHENG Xiao, GAO Yun-quan. Adaptive Histogram Publishing Algorithm for Sliding Window of Data Stream[J]. Computer Science, 2022, 49(10): 344-352.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210700242

面向数据流滑动窗口的自适应直方图发布算法

王修君[1,2,3]　莫　磊[1,3]　郑　啸[1,2,3]　高云全[1,2,3]
1 安徽工业大学计算机科学与技术学院　安徽 马鞍山 243032
2 合肥综合性国家科学中心人工智能研究院　合肥 230091
3 安徽省工业互联网智能应用与安全工程实验室　安徽 马鞍山 243032

摘　要　差分隐私技术作为一种有效的隐私保护机制,已被广泛应用在诸多领域。目前已有的静态数据集和动态数据集上的直方图发布方法在处理数据流滑动窗口模型时,往往只能够通过对数据直方图信息添加统一噪声的形式来实现数据保护,这导致了它们在实际应用中存在数据可用性低、时间复杂度高等问题。针对这些问题,文中通过将数据流近似计数技术综合到差分隐私保护算法中,进而提出了一种面向数据流滑动窗口模型的自适应直方图发布方法 APS(Adaptive Histogram Publishing Method for Sliding Window)。APS 算法首先利用数据流近似计数方法来预测下一时刻滑动窗口内数据的分布信息;然后通过比较估计值与真实值之间的差异来选取合适的发布值;最后对排序后的直方图区间进行聚类处理,并优化其桶内数据的误差。理论分析显示,APS 算法能够在减少隐私预算的同时,有效地提高数据的可用性和缩短运行时间。在两种不同的真实数据集上的实验结果也验证了 APS 算法在数据可用性和运行时间上显著优于现有的基于分组的直方图发布算法。

关键词:差分隐私;滑动窗口;数据流;直方图发布;近似误差;拉普拉斯误差
中图法分类号　TP309

Adaptive Histogram Publishing Algorithm for Sliding Window of Data Stream

WANG Xiu-jun[1,2,3],MO Lei[1,3],ZHENG Xiao[1,2,3] and GAO Yun-quan[1,2,3]
1 School of Computer Science and Technology,Anhui University of Technology,Maanshan,Anhui 243032,China
2 Institute for Artificial Intelligence,Hefei Comprehensive National Science Center,Hefei 230091,China
3 Anhui Engineering Laboratory for Intelligent Applications and Security ofIndustrial Internet,Maanshan,Anhui 243032,China

Abstract　As one of the most effective privacy protection mechanisms,differential privacy has been widely used in many fields. The existing histogram publishing methods for either static data set or dynamic data set mainly protect the privacy of sliding windows in data streams by adding unified noise. This leads to low data availability,high time complexity and weak privacy protection in their practical applications. In this paper,we tackle this problem by integrating the approximate counting techniques into the differential privacy and proposing an adaptive histogram publishing method for sliding window(APS). Firstly,the proposed APS predicates the distributional information of the sliding windows in the data stream by using an approximate counting method. Secondly,it computes an appropriate value suitable for publishing by checking the difference between estimated values and actual values. Finally,it reduces statistical errors within each interval by clustering. Theoretical analysis shows that the APS algorithm can effectively improve data availability and reduce running time while reducing the privacy budget. Experimental results on two different real data sets also verify the superiority of APS algorithm over existing grouping-based histogram publishing algorithms in terms of data availability and running time.

Keywords　Differential privacy,Sliding window,Data stream,Histogram publishing,Approximation error,Laplacian error

到稿日期:2021-07-25　返修日期:2021-12-08
基金项目:安徽高校协同创新项目(GXXT-2020-012);国家自然科学基金(61402008,61702006,61672038);安徽省重点研发与开发计划面上攻关项目(202004a05020009,201904a05020071);安徽省自然科学基金(2108085MF218);安徽高校自然科学研究项目(KJ2020A0249,KJ2020A0250);安徽普通高校重点实验室开放基金(CS2020-06)
This work was supported by the University Synergy Innovation Program of Anhui Province(GXXT-2020-012),National Natural Science Foundation of China(61402008,61702006,61672038),Provincial Key Research and Development Program of Anhui Province(202004a05020009,201904a05020071),Natural Science Foundation of Anhui Province(2108085MF218),University Natural Science Research Project of Anhui Province(KJ2020A0249,KJ2020A0250) and Open Fund of Key Laboratory of Anhui Higher Education Institutes(CS2020-06).
通信作者:王修君(xjwang@ahut.edu.cn)

1 引言

随着信息技术高速发展，众多的应用程序往往需要动态发布其已经生成或获取的数据，例如：1）在用户浏览网站时，通过获取用户的浏览信息来分析用户喜好[1]；2）在实时交通信息系统中对当前的交通状况进行分析，有助于对目的地的规划以及流量状态的预测[2]；3）医疗卫生系统需要实时发布患者的医疗数据[3]。如果我们对以上数据不加以保护而直接进行发布，则会引发数据的安全性问题。

作为一种目前最有效的数据隐私保护的标准和方法，差分隐私[4-7]已经被广泛地应用到疾病监测[8]、实时交通监控[9]、互联网流量分析[10]等诸多领域。

从本质来说，差分隐私保护敏感数据的原理为：通过对数据添加随机噪声来达到保护敏感数据的目的。注意，一方面，加入随机噪声后，当用户查询数据库中的数据时，他们获得的查询结果中就会包含这些随机添加的噪声，这使得他们无法通过倒推查询结果来获得数据库中的敏感数据信息；另一方面，这些加入的噪声也降低了原始数据的精度，从而影响了数据的效用（如查询结果过于失真）。因此，在使用差分隐私算法时，为了达到合乎要求的数据保护级别的目的和保证数据的效用，需要严格控制加噪的规模，即如果添加过多的噪声，敏感数据的保护级别会增强，但数据的效用会降低；相反，如果加入过少的噪声，敏感数据的效用会增强，但数据的保护级别会降低。

直方图作为报告数据分布和统计分析的核心数据描述方法，是数据发布的重要形式[11-12]。简单来说，直方图中的桶频率表示不同的属性值区间的计数，由直方图表示的数据通常支持多种关键查询，如范围计数查询、聚合查询等。但是，通过直方图发布数据存在隐私泄露的问题，导致该问题的原因可由下面的例子来说明。

表1列出了一个关于患病直方图发布的例子，在表1中可以看到关于患者是否患糖尿病的敏感个人记录，图1给出了依据表1中的数据记录而构建的关于糖尿病人数目的直方图。此时，如果直接发布图1所示的直方图，糖尿病人的隐私信息（敏感信息）就会被泄露。比如，如果攻击者想要知道病人 Lisa 患糖尿病的情况，并且假设该攻击者具有很强的背景知识，即知道 Lisa 的年龄和属于同一个年龄区间的其他人的患病信息。此时，攻击者便可以简单地通过如下过程来推断出病人 Lisa 的患病信息。

表1 一个敏感数据集

Table 1 A sensitive dataset

编号	姓名	年龄	患病
1	Lisa	38	Y
2	Emm	55	N
3	Jessie	81	Y
4	Abby	31	Y
5	Anne	45	Y
6	Colin	69	Y
7	Kelly	22	Y
…	…	…	…

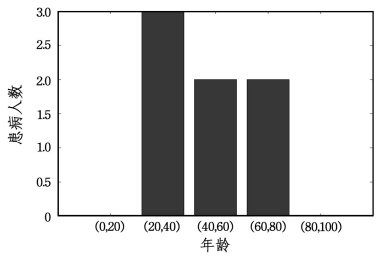


图1 患糖尿病直方图

Fig. 1 Histogram of diabetic patients

Step1 攻击者已知 Lisa 的年龄是38岁，但不知道她是否患糖尿病；

Step2 攻击者已知直方图中属于[20，40]年龄段内除 Lisa 之外其他人的患病信息（已知 Kelly 和 Abby 患糖尿病）；

Step3 由于直方图中属于[20，40]年龄段的人数为3，攻击者可以得出 Lisa 也患了糖尿病。

从上述分析过程可以看出：在发布与数据流对应的直方图时，如何对其进行有效的隐私保护处理是一个重要的问题。一般来说，使用差分隐私技术可以达到隐私保护的要求，即在发布的直方图中加入噪声，使得攻击者在即使拥有很强的数据集合背景知识的情况下，也无法推断出原始数据流中是否存在某一条特定记录。更多的关于差分隐私可以有效保护直方图隐私信息的事例请参考文献[8-24]。

目前，虽然存在着众多的针对静态数据集的直方图发布方法[13-16]，但这些方法在处理数据流时都需要缓存大量的历史数据而几乎无法快速地处理数据流（运行时间长）。另一方面，虽然现在已经有一些专门针对数据流滑动窗口的直方图发布方法[21-25]，但是这些方法存在以下问题：

（1）现有算法未考虑数据流近似统计方法和数据隐私保护问题内在的相关性，未利用这个特性来设计解决方案。具体地，数据流近似统计方法[26-27]一般可以快速地返回某个真实值 x 的近似估计值 $\tilde{x}$。由于 $\tilde{x}\neq x$，$\tilde{x}$ 也可以看作是 x 的一种隐私保护，而差分隐私的基本保护机制也是通过给真实值 x 加上噪声变量 ε 的方式来保护 x，因此两者之间存在密切的相关性。

（2）现有算法的运行时间长，空间开销较大。更加具体地说，现有差分隐私算法仍然依赖于时刻缓存整个窗口数据再加噪发布的设计策略，这使得这些算法需要消耗大量内存空间来存储历史数据，以保证可以在任意时刻创建直方图。同时，由于这些算法在每个时刻生成的直方图都是通过对所有窗内数据逐一加噪而获得的，因此使得它们的时间开销也较长。

这两个问题导致了目前已有的算法在处理数据流滑动窗口模型时存在发布数据可用性低和时空开销大的问题。

本文利用滑动窗口近似统计计数方法和优化的在线加噪策略来解决这两个问题。本文的具体贡献如下：

（1）提出了一种面向数据流滑动窗口模型的自适应直方图发布算法 APS。该算法包含3个关键步骤：1）利用滑动窗口近似统计技术来获得下一时刻的数据流直方图估计信息；2）比较近似值和真实值之间的差异，优化选取合适的直方图发布值；3）利用聚类算法来重新优化直方图的分组，从而降低桶内数据的误差。通过该算法达到了降低隐私预算和增加

数据可用性的目的。

(2)从理论上证明了基于滑动窗口的区间估计算法的近似误差不超过阈值$\frac{W}{2k}$,且证明了 APS 算法的空间开销显著小于现有的基于分组的直方图发布方法。

(3)在两个不同真实数据集上进行实验,将本文方法与现有的差分隐私直方图发布方法进行度量比对。实验结果显示,与现有的直方图发布方法相比,APS 算法显著地提高了数据的可用性,缩短了运行时间。

本文第 2 节介绍了相关工作;第 3 节提供了数据流处理模型、差分隐私以及直方图的相关知识;第 4 节介绍了一种数据流环境下的基于滑动窗口自适应直方图发布方法以及相关理论;第 5 节分析了所提方法的隐私性和可用性;第 6 节给出了实验设置与结果分析;最后总结全文。

2 相关工作

目前基于差分隐私的直方图发布方法可以分为两类:静态数据上的发布方法和动态数据上的发布方法。

静态数据上的发布方法主要有:文献[13]提出了 StructureFirst 算法,该算法通过对生成的噪音直方图中的相邻和值相近的频数进行合并,来降低查询结果中的噪声量;文献[14]提出了 P-HPartition 方法来压缩直方图,该方法利用真实数据集的固有冗余性,来解决将噪声添加到直方图计数中时精度不高的问题;文献[15]提出了一种精确直方图发布算法 DiffHR,该方法首先利用 Metroplis-Hasting 算法联合指数机制对直方图区间数进行排序,并依据贪心聚类的思想对处理数据分组,这使得这组数据拥有较好的可用性;文献[16]提出了隐私预算自适应分配方法 APB,该方法根据隐私预算权重分配的模型进行优化计算总误差最小分配比例,并对结果采用贪心分组,该算法均衡了噪声误差和重构误差,提高了数据可用性。

动态数据上的发布方法主要有:文献[17]提出了一种基于差分隐私的连续计数器,它在每个时间的步长来输出所看到的 1 的值,保护了数据流连续发布的隐私信息;文献[18]提出了一种分布式时间序列数据的差分隐私聚合算法,该算法采用了离散傅里叶变换的方法对查询结果进行扰动;文献[21]提出了一种针对二维空间数据流的隐私保护发布算法 PTDSS-SW,该算法能够以较低空间开销来达到隐私保护的效果;文献[22]提出了 SHP 算法,首先将滑动窗口中的桶计数分成不同的分组,然后根据数据采样结果的不同来自适应分配隐私参数,该方法能降低整体隐私预算的效果;文献[23]提出了数据流直方图发布算法 ASDP-HPA,结合自回归集成移动平均模型和动态滑动窗口的方法,来保证数据安全性和数据可用性;文献[24]提出了一种基于 Kulback-Leibler 散度的直方图发布算法,该算法利用 KL 相似性度量的方法,降低了噪音,提升了数据可用性。

总体来说,现有的直方图发布算法在处理我们的问题会出现以下问题:1)已有的静态数据无法实时处理数据流;2)已有的数据流算法未考虑利用直方图发布和滑动窗口近似统计之间的相关性;3)已有的数据流算法仅考虑依据当前窗口内数据进行简单计数,没有考虑算法的时空效率。为了解决以上问题,提出了面向数据流滑动窗口模型自适应直方图发布算法 APS。APS 算法综合了近似统计算法与直方图发布算法,利用自适用发布方法发布合适的噪音,在降低隐私预算的同时,减小了时空的消耗。

3 理论基础与相关定义

本节对数据流处理模型和差分隐私技术以及直方图的概念进行了介绍。

3.1 数据流处理模型

数据流作为一种实时动态数据模型,已被广泛地应用到众多数据处理应用中[28-30]。一个数据流被定义为一个无线长度的数据序列,在每个时刻都会有新数据到达。因此,在数据流模型中,一般认为数据的总量远超可用内存容量[31-32]。

目前,现有的处理数据流的模型[33]主要分为快照模型(Snapshot Model)、界标窗口模型(Landmark Model)和滑动窗口模型(Sliding Window Model)等。

(1)快照模型:两个已知时间戳之间的所有数据进行处理统计。

(2)界标模型:对开始时间戳到当前时间戳之间的所有数据进行处理,并对这些数据进行数据统计。

(3)滑动窗口模型: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。

本文考虑使用滑动窗口模型来处理数据流,原因是滑动窗口模型能够对历史数据以及近期数据进行很好的处理,并且考虑到内存空间的使用和磁盘的访问数据量,通常数据的时效性随时间衰减,即越近的数据越重要。

3.2 差分隐私技术

差分隐私技术将独立的拉普拉斯噪音添加到发布数据来实现隐私保护的目的。具体地,该技术通过对每一个需要发布的数据加入符合用户预设隐私预算的噪音来扰动原始真实数据,从而实现保护原始真实数据的隐私信息的目的。

为了在数据流上使用差分隐私技术,本文给出如下定义。

定义 1(近邻关系) 给定两个数据流 D 和 D',如果两者之间最多只有一个有差别的记录,那么 D 和 D' 是邻居数据流[4]。

定义 2(ε-差分隐私) 给定数据流 D 和 D'(近邻数据流),设定一个符合差分隐私的算法 A,如果 A 在 D 和 D' 上的任何输出结果 $O\subseteq range(A)$ 满足以下不等式(1),则 A 表示基于用户的 ε-差分隐私[4]。

$$Pr[A(D)=O]\leqslant \exp(\varepsilon)\times Pr[A(D')=O] \tag{1}$$

由此不等式可知,$range(A)$ 表示算法 A 的取值范围,$A(D)$表示以 D 为算法 A 的输入时得到的输出,$Pr[\cdot]$表示输出 $A(D)=O$ 的概率,并且通过其来控制隐私保护的效果。从不等式可以看出,隐私预算 ε 值越小,则 $A(D)=O$ 和 $A(D')=O$ 之间的概率值越接近,说明算法 A 对个人隐私保护得越好。

定义 3(全局敏感度) 对于任意一个函数 $f:D\rightarrow R^d$ 来说,函数 f 的全局敏感度定义为:

$$\Delta f=\max_{D,D'}\| f(D)-f(D')\| \tag{2}$$

其中,R 表示映射的实数空间,d 表示函数 f 的查询维度。

文献[20]利用经典的拉普拉斯分布来产生随机噪声值,并将其添加到原始真实值上来得到加噪值。此方式满足了

差分隐私保护的典型要求，该典型要求如定理1所示。

定理1 对于任何一个函数 $f:D\to R^d$，如果算法 A 的结果满足式(3)，那么说明 A 满足 ε-差分隐私[23]。

$$A(D)=f(D)+\langle Lap_1\left(\frac{\Delta f}{\varepsilon}\right),\cdots,Lap_d\left(\frac{\Delta f}{\varepsilon}\right)\rangle \tag{3}$$

Laplace 分布的概率密度函数为：

$$g(x)=\frac{1}{2b}e^{-\frac{|x|}{b}} \tag{4}$$

随着 $b(b>0)$ 增大，添加的噪声增大，隐私保护程度增强。一般情况下，$b=\frac{\Delta f}{\varepsilon}$，则隐私预算 ε 越小，b 越大，隐私保护程度越好。如果 b 过大，则添加的噪声较大，从而导致发布数据的可用性明显降低。

除上述定义之外，我们还需要使用以下两个属性。

属性1(序列组合属性) 给定数据集 D 和 n 个随机算法 $\langle A_1,A_2,\cdots,A_n\rangle$，$A_i(1\leqslant i\leqslant n)$ 满足 ε_i-差分隐私，则 $\langle A_1,\cdots,A_n\rangle$ 在 D 上的序列组合满足 ε-差分隐私，且满足 $\varepsilon=\sum_{i=1}^{n}\varepsilon_i$[34]。

属性2(并行组合属性) 给定数据流 D 并将其划分成 n 个不相交的子集 $D=\{D_1,D_2,\cdots,D_n\}$，如果算法 A 满足 ε-差分隐私，则算法 A 在 $\{D_1,D_2,\cdots,D_n\}$ 中也能进行操作，结果满足 ε-差分隐私[34]。

本文通过属性1和属性2证明了本文提出的APS算法的隐私性，详情请见5.1节。

3.3 直方图的概念

直方图是许多研究领域的基本工具，包括数据分析、计算机视觉等。给定数据库中的一组数据，即 $d=\{x_1,\cdots,x_N\}$，假设属性 P 上这些元组的值分别为 $\{x_{1,P},\cdots,x_{N,P}\}$，则属性 P 的直方图由一组等宽的桶组成，即 $H=\{H_1,H_2,\cdots,H_n\}$，其中每个桶 H_i 与区间范围相关联，并分配统计值作为区间范围内的价值。为简单起见，我们用 H_i 表示对当前区间中所对应的区间计数值。通常，桶的范围不会相互重叠，它们的并集应覆盖 P 中的所有值，这种直方图能够实时回答属性 P 上的一系列范围查询。

4 数据流环境下的基于滑动窗口自适应直方图发布方法

本节主要介绍APS算法的概述以及该算法的具体实现细节，并在本节的最后说明了近似统计误差与拉普拉斯噪音分布的标准差之间的关系。本文中使用的符号的定义如表2所列。

表2 符号定义

Table 2 Symbol definition

符号	定义
D_i	第 i 个数据的值
b_t	b 中最老的块的索引值
y_j	第 j 位的1的计数值
b	大小为 k 的数组
m	所在块内偏移量
B	b 中所有位的总和
$\widetilde{H}_i$	所在区间内的加噪值
$\hat{H}_i$	所在区间内的估计值
$\bar{H}_{w_i}$	所在窗口内的所有区间加噪值

4.1 APS算法概述

APS算法是利用滑动窗口近似统计方法对下一时刻的区间统计值进行估计，若估计值与真实值之间的差异小于文中给定的阈值，则将发布估计值，否则发布噪音值；利用贪心聚类算法将排序后的直方图区间进行分组，并通过优化分组的误差下限来降低每组桶之间的误差。

4.2 数据流环境下的基于滑动窗口自适应直方图发布方法(APS)

当滑动窗口随着时刻变化时，APS算法会在每一时刻发布一个差分隐私直方图。APS算法的描述如算法1所示。

算法1 APS算法

输入：数据流 D，滑动窗口 W，隐私参数 ε，窗口内的分组数 M

输出：隐私处理后的直方图 $\bar{H}_{w_i}$

1. $\varepsilon_1=\alpha\varepsilon,\varepsilon_2=\varepsilon(1-\alpha)$
2. 使用 ANM 算法判断出噪音值 $\widetilde{H}_i$
3. 将 $\widetilde{H}_i$ 组成当前窗口直方图统计数据 $\widetilde{H}_{w_i}$
4. $\widetilde{H}_{w_i}=\mathrm{sort}(\widetilde{H}_{w_i})$
5. $C=\mathrm{Clustering}(\widetilde{H}_{w_i})$
6. for 每一个 $C_i\in C$
7. $\quad\bar{C}_i=\sum_{\widetilde{H}\in C_i}H_j/|C_i|$
8. for 每个 $H_j\in H_{w_i}(H_j\in C_i)$
9. $\quad\bar{H}_j=\bar{C}_i+\mathrm{Lap}(\frac{1}{\varepsilon_2})/|C_i|$
10. end for
11. end for
12. if $\bar{H}_i\leqslant 0$:
13. $\quad\bar{H}_i=0$
14. end if
15. return $\bar{H}_{w_i}=\{\bar{H}_1,\bar{H}_2,\cdots,\bar{H}_M\}$

对当前滑动窗口内的直方图进行的操作如下：

(1)通过 ANM，输出 $\widetilde{H}_i$(见算法1中的第2行)。

(2)将噪音值插入当前直方图统计数据中(见算法1中的第3行)。

(3)首先利用排序算法对滑动窗口内的统计数据进行排序；并计算每一个频数与其他频数之间合并的误差以及不合并的误差。检查直方图频数合并和不合并误差大小并进行区间分组(见算法1中的第4—5行)。

(4)使用每个分组平均值取代在每个分组中的频数值。利用分组中的区间个数，动态分配隐私参数进行加噪(见算法1中的第6—11行)。

(5)通过非负性约束(减少噪音误差所产生的误差)，过滤掉不符合条件的数值，最后返回当前 i 时刻的噪声直方图(见算法1中的第12—15行)。

4.3 基于滑动窗口近似估计的自适应加噪方法(ANM)

为了减少拉普拉斯误差与节省隐私预算，利用在当前窗口的数据进行近似估计，使用估计值来取代加噪值的方法，在近似统计算法的基础上，提出了基于滑动窗口近似估计的自适应加噪方法 ANM 算法(Adaptive Noise Adding Method for Sliding Window Approximate Estimation)，用于提高数据的可用性。

算法2 ANM算法

输入：数据流 D，窗口大小 W，数据流的长度 L，前一时刻的统计量

信息 m, b_t, B, y

输出：输出当前值 $\widetilde{H}_i$

1. for 当前窗口区间数据：
2. if $D_t \in M_i$: /* 判断 D_t 是否属于 M_i 区间 */
3. $D_i=1$
4. else
5. $D_i=0$
6. end if
7. if $m=\frac{W}{k}-1$; /* 判断是否为子块的最后一位 */
8. $B=B-b_t$
9. if $y+D_i \geqslant \frac{W}{k}$:
10. $b_t=1$; $y=y-\frac{W}{k}+D_i$;
11. else
12. $b_t=0$; $y=y+D_i$;
13. $B=B+b_t$; $m=0$; $t=(t+1) \bmod k$;
14. else
15. $y=y+D_i$; $m=m+1$;
16. $\hat{H}_i=\frac{W}{k}\times B+y-\frac{W}{2k}-m\times b_t$; /* 区间估计值 */
17. end for
18. for $H_i \in [\hat{H}_i-\frac{W}{2k}, \hat{H}_i+\frac{W}{2k}]$
19. if $|H_i-\hat{H}_i| \leqslant \frac{\sqrt{2}}{\varepsilon_1}$ then
20. $count_1=count_1+1$; $R=R+H_i$;
21. else
22. $count_2=count_2+1$
23. end if
24. end for
25. if $count_1 > count_2$ then
26. $\widetilde{H}_i=rand(R)+Lap(\frac{1}{\varepsilon_1})$;
27. else
28. $\widetilde{H}_i=\hat{H}_i+Lap(\frac{1}{\varepsilon_1})$;
29. end if
30. if $\widetilde{H}_i \leqslant 0$
31. $\widetilde{H}_i=0$ /* 非负性约束 */
32. end if

对于当前滑动窗口数据流每一个元素进入滑动窗口的自适应加噪过程如下：

(1) 对于当前数据进行区间判定，判定是否在当前的某一个区间中，并进行标记(见算法 2 中的第 2－6 行)。

(2)利用区间分块的原理近似统计当前窗口内的数据，并获得近似统计值(见算法 2 中的第 7－16 行)。

(3)利用近似阈值与加噪阈值之间的差异，获得计数值 $count_1$ 和计数值 $count_2$(见算法 2 中的第 18－24 行)。

(4)根据计数值之间的大小判定添加噪音(见算法 2 中第 25－29 行)。

(5)通过非负性约束(减少近似统计所产生的误差)，过滤掉不符合条件的数值，最后得到 $\widetilde{H}_i$(见算法 2 中的第 30－32 行)。

由于算法 2 利用了近似统计的方法，其统计频数与真实频数存在着一定的误差，根据算法 2 容易得出以下结论。

结论 1 对于 $H_i \in H$，每个区间的结果为 $|\hat{H}_i-H_i| \leqslant \frac{W}{2k}$。

证明：假设当前窗口所到达的位置为 $m+W$，其中 D_W 表示分块的最后一位，且 $m<\frac{W}{k}$。处理完 $m+W$ 位后，我们再考虑 b_t。

考虑数据流入第一个子块情形下，通过算法 2 可知：

$$|\hat{H}_i-H_i|=y_0+\sum_{j=1}^{m}D_j-m\times b_t-\frac{W}{2k}$$

我们考虑以下两种情况：

(1)当 $b_t=1$ 时，我们所考虑的是 y 大于 $\frac{W}{k}$ 的情况，可以得到不等式：$y_0+\sum_{j=1}^{\frac{W}{k}}D_j \geqslant \frac{W}{k}$ 和 $y_0+\sum_{j=1}^{m}D_j \geqslant \frac{W}{k}-\sum_{j=m+1}^{\frac{W}{k}}D_j$，进而可以得出：

1) $|\hat{H}_i-H_i|=y_0+\sum_{j=1}^{m}D_j-m\times b_i-\frac{W}{2k}$

$\geqslant \frac{W}{k}-\sum_{j=m+1}^{\frac{W}{k}}D_j-m-\frac{W}{2k}$

$\geqslant \frac{W}{k}-(\sum_{j=m+1}^{\frac{W}{k}}1)-m-\frac{W}{2k} \geqslant -\frac{W}{2k}$

$|\hat{H}_i-H_i| \geqslant -\frac{W}{2k}$

2) $|\hat{H}_i-H_i|=y_0+\sum_{j=1}^{m}D_j-m-\frac{W}{2k} \leqslant \frac{W}{k}-\frac{W}{2k}=\frac{W}{2k}$

$|\hat{H}_i-H_i| \leqslant \frac{W}{2k}$

由 1)和 2)的结果可知，当 y 大于 $\frac{W}{k}$ 时有：

$$|\hat{H}_i-H_i| \leqslant \frac{W}{2k}$$

(2)当 $b_t=0$ 时，我们所考虑的是 y 小于 $\frac{W}{k}$ 的情况，可以得到不等式：$y_0+\sum_{j=1}^{\frac{W}{k}}D_j \leqslant \frac{W}{k}-1$ 和 $y_0+\sum_{j=1}^{m}D_j \leqslant \frac{W}{k}-\sum_{j=m+1}^{\frac{W}{k}}D_j-1$

1) $|\hat{H}_i-H_i|=y_0+\sum_{j=1}^{m}D_j-\frac{W}{2k} \geqslant y_0-\frac{W}{2k} \geqslant -\frac{W}{2k}$ | $\hat{H}_i-H_i| \geqslant -\frac{W}{2k}$

2) $|\hat{H}_i-H_i|=y_0+\sum_{j=1}^{m}D_j-\frac{W}{2k} \leqslant \frac{W}{k}-\sum_{j=m+1}^{\frac{W}{k}}D_j-\frac{W}{2k}-1$ |

$\hat{H}_i-H_i| \leqslant \frac{W}{2k}-1$

由 1)和 2)的结果可知，当 y 小于 $\frac{W}{k}$ 时有：

$$|\hat{H}_i-H_i| \leqslant \frac{W}{2k}$$

通过(1)和(2)的结果可知，由于 $\hat{H}_i$ 可以使用非负性约束，使得 $\hat{H}_i$ 一直大于 0，因此导致 $|\hat{H}_i-H_i| \leqslant \frac{W}{2k}$ 一直成立。

定理 2 ANM 算法的内存需要为 $k+2\log W+O(1)$-bit 内存。

证明：本文算法中 y 需要使用 $\left\lceil 2+\log\left(\frac{W}{2k}\right)\right\rceil$-bit 的内存，$m$ 使用了 $\left\lceil 1+\log\left(\frac{W}{2k}\right)\right\rceil$-bit 的内存，$b$ 使用了 k-bit 的内存。此外，i 需要 $\lceil \log k \rceil$-bit，B 需要另一个 $\lceil \log(k+1) \rceil$-bit。总体而言，ANM 算法所需的 bit 数为：$k+\left\lceil 2+\log\left(\frac{W}{2k}\right)\right\rceil+\left\lceil 1+\log\left(\frac{W}{2k}\right)\right\rceil+\lceil \log k \rceil+\lceil \log(k+1) \rceil \leqslant k+2\log W+O(1)$。

4.4 APS 算法的空间复杂度

定理 3 APS 算法需要 $Mk+2M\log W+O(1)$ 位内存。

证明：APS 算法的空间复杂度由两个部分共同决定，第一部分为基于滑动窗口近似估计的自适应加噪方法，第二部分为贪婪聚类的分组方法，其中 APS 算法的空间复杂度主要由基于滑动窗口近似估计的自适应加噪方法中的数据结构决定。由定理 1 可知：ANM 算法需要的内存为 $k+2\log W$。对于一个直方图来说有 M 个区间，APS 算法所需要的空间为 $Mk+2M\log W+O(1)$。

5 隐私性和可用性

算法的隐私性用于验证算法是否满足 ε -差分隐私的概念和性质，可用性用于衡量直方图最终的发布误差。本节验证了 APS 算法满足 ε-差分隐私，并分析了该算法的发布误差。

5.1 隐私性分析

定理 4 APS 算法满足 ε-差分隐私。

证明：算法 1 中的第 2 行使用拉普拉斯机制来计算每个直方图区间的噪声计数，第 9 行满足 ε_2-差分隐私。使用属性 2 来解释算法 1 中的第 9 行，使用拉普拉斯机制来计算每个直方图区间的噪声均值。由于每个直方图的区间大小是已知的，对于每一个区间所建立的直方图数据，都需要知道区间噪音，总误差为所有区间的直方图的噪音总和。

算法 2 中的第 28 行满足 ε_1-差分隐私。使用属性 1 来解释，我们了解到算法 1 中的第 9 行和算法 2 中的第 28 行一起满足差分隐私定义——$(\varepsilon_1+\varepsilon_2)$-差分隐私，即说明 APS 算法满足 ε-差分隐私。

5.2 可用性分析

算法的可用性是通过直方图发布所产生的误差来进行计算的，这是衡量差分隐私算法性能的另一标准。

本文中的误差为：

$$Err(H_{W_i})=E\left(\sum_{H_i\in W_i}(H_i-\bar{H}_i)^2\right)$$

本文中总误差由 3 个误差构成：1）近似算法的误差；2）直方图的重构误差；3）噪声误差。其中，近似算法的误差主要取决于滑动窗口大小 W 和窗口单元个数 k，直方图重构误差取决于分组的大小，噪声误差取决于分配隐私预算 ε 的大小。

下文通过将 APS 算法和差分隐私直方图发布算法进行比较，来分析它们的可用性。直方图发布算法的误差如表 3 所列。

表 3 直方图发布算法总误差

Table 3 Total error of histogram publishing algorithm

算法	总误差
SF[13]	$\sum_{H_j\in C_i}(H_j-\bar{C}_i)^2+2/\varepsilon_2^2$
PHP[14]	$\sum_{H_j\in C_i}(H_j-\bar{C}_i)^2+2/\lvert C_i\rvert\varepsilon_2^2$
DiffHR[15]	$\sum_{H_j\in C_i}(H_j-\bar{C}_i)^2+2/\lvert C_i\rvert\varepsilon_2^2$
APB[16]	$\sum_{H_j\in C_i}(H_j-\bar{C}_i)^2+2/\lvert C_i\rvert\varepsilon_2^2$
APS	$\sum_{H_j\in C_i}(H_j-\bar{C}_i)^2+2/\lvert C_i\rvert\varepsilon_2^2$

SF 算法将隐私预算分为 ε_1 和 ε_2 两部分，第一部分隐私预算 ε_1 用于指数机制随机选取分组边界，第二部分隐私预算 ε_2 用于拉普拉斯机制对分组进行拉普拉斯加噪。PHP 算法同样将隐私预算分为两部分，首先通过指数机制找出每次的二分割点，然后对分组后的直方图进行拉普拉斯加噪。APS 算法、DiffHR 算法和 APB 算法一样，都是先用 ε_1 对原始直方图的分组结构进行保护，再用 ε_2 对分组后的直方图添加噪音。很明显，如果所有算法用于拉普拉斯加噪的隐私预算都相同且等于隐私预算的一半，那么 APB 算法误差的第二项比 SF 要低。

考虑分组误差第一项，因为 APS 算法、APB 算法、DiffHR 算法采用先排序后分组的策略，可以将全局的计数相同的桶都放到同一个分组中，很大程度上降低了分组引入的重构误差，所以 APS 算法的重构误差比 SF 算法和 PHP 算法低，故 APB 算法的可用性高于 SF 和 PHP。APS 算法、APB 算法以及 DiffHR 算法误差的不同主要取决于排序前分组误差的不同，由于 APS 算法通过预测值来代替加噪值，降低了分组前的误差，而 APB 算法和 DiffHR 算法只是采用了不同的隐私预算分配参数，进而造成了更多的噪音误差，因此 APS 算法的可用性高于 APB 算法和 DiffHR 算法。

6 实验设置与结果分析

6.1 实验设置

实验环境为 AMD Ryzen 5 2600 CPU 3.4 GHz，16 GB 内存，Windows 10 操作系统。

本文采用了两个真实的数据集 UK Accidents 和 Taxi 进行验证。UK Accidents 抽取了从 2005－2014 年英国人身伤害道路事故情况的详细道路安全数据，每条记录包含伤亡年龄、性别、伤亡严重程度、伤亡社会阶层、伤亡类型等属性，其中年龄区间为 $[0,100]$[1)]；Taxi 抽取了有关 2019 年 1 年纽约黄色出租车详细行程数据集，每条记录包含接送日期/时间、接送地点、行程距离、费率类型和司机报告的乘客计数等属性，其中乘客计数区间为 $[0,6]$[2)]。

对于这两个真实数据集，我们分别用 Matlab2014b 编程语言实现了 APS 算法、DiffHR 算法[20]以及 APB 算法[21]，并将发布的直方图的数据可用性与这两种算法进行了比较，本文使用均方误差（MSE）对算法的可用性和发布数据的准确

1) https://www.kaggle.com/benoit72/uk-accidents-10-years-history-with-many-variables?select=Casualties0514.csv

2) https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

性进行度量。请注意,均方误差越小,已发布数据的数据可用性越高。

6.2 APS算法的隐私预算分配设置实验

为了测试本文算法在隐私预算的不同分配比例下对算法的可用性产生的影响,对于3种不同数据集、固定滑动窗口和分块的大小,我们将隐私预算分配比率设置为0.1,0.3,0.5,0.7和0.9,并获得隐私预算不同分配比例下的均方误差,如图2所示。

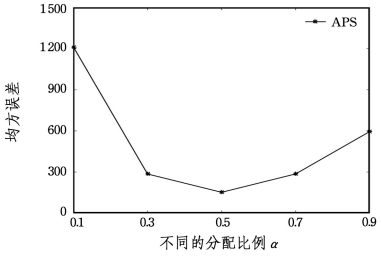


图2 不同分配隐私预算比例下的均方误差

Fig. 2 Mean square error with different allocation ratios of privacy budget

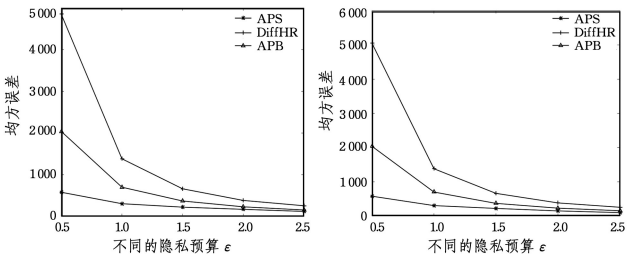
图2中,在英国车祸数据集下的均方误差首先随着隐私预算分配比率 α 的增加而增加;然后,在 $\alpha=0.5$ 时,均方误差到达最低点;最后,均方误差随着隐私预算分配比率 α 的增加而增加。产生这样的原因有以下两点:1)自适应加噪方法是在近似统计值与噪音值之间选取一个合适的值,因此该算法在提升数据的运行效率的同时,能够增加发布数据的可用性;2)贪心分组算法中的隐私预算主要用于添加拉普拉斯噪声。随着隐私预算的增加,数据中添加的噪声相应减少,从而使得最终的数据误差减小。其中隐私预算分配比率 α 为0.5时,均方误差最为合适。因此,在后文的实验中,本文算法中的隐私预算分配比例 α 设置为0.5。

6.3 发布数据的可用性比较

本节在不同的隐私预算 ε、不同的滑动窗口 W 和不同的窗口分块个数 k 的条件下,在英国车祸数据集上测试APS算法、DiffHR算法、APB算法的均方误差。为了确保实验结果的准确性,我们将这3种算法分别在两种不同真实数据集上重复运行30次。

(1)隐私预算 ε 与均方误差

我们设置窗口大小 W 为1000,k 为1000,我们分别在不同隐私预算 ε 下使用APS算法与DiffHR算法和APB算法来测试已发布数据的均方误差,最终结果如图3所示。



(a)British car accident data set (b)New York Taxi Data set

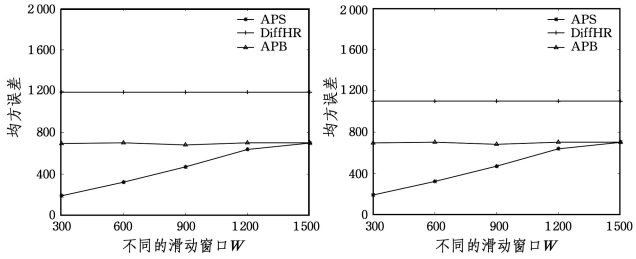
图3 隐私预算与均方误差的曲线

Fig. 3 Curves of privacy budget and mean square error

图3给出了在两种不同数据集的情况下,APS算法、DiffHR算法和APB算法的隐私预算与均方误差的关系。由图3可知,隐私预算提高,致使数据具有更少的均方误差,数据可用性变高。本文算法通过自适应的加噪方式(预测值与真实值之间的差值同阈值进行加噪)导致误差更低,使得APS算法的可用性好于APB算法和DiffHR算法。

(2)滑动窗口 W 与均方误差

我们分别在不同滑动窗口 W 下使用APS算法与DiffHR算法和APB算法来测试已发布数据的均方误差,最终结果如图4所示。



(a)British car accident data set (b)New York Taxi Data set

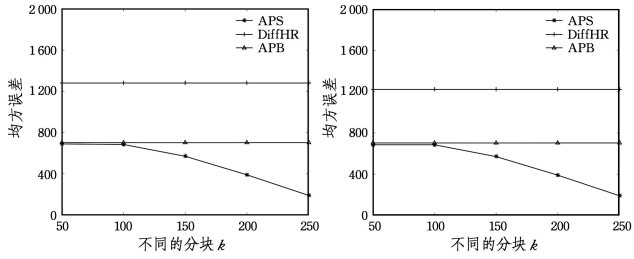
图4 窗口大小与均方误差的曲线

Fig. 4 Curves of window size and mean square error

图4给出了在两种不同数据集的情况下,APS算法、DiffHR算法和APB算法的滑动窗口大小与均方误差的关系。由图4可知,APS算法的均方误差增大是因为滑动窗口大小增加。随着滑动窗口增大,近似统计误差随着滑动窗口的增大而增加,近似统计值的噪音大于噪音统计值,从而导致总误差增加,均方误差增加。APS算法均方误差明显少于APB算法以及DiffHR算法,原因是APS算法采用滑动窗口近似统计值来代替噪音值,进而避免了在滑动窗口中直接添加噪音所带来的结果。当滑动窗口到达一定值时,APS算法和APB算法拥有相同的均方误差。

(3)分块大小 k 与均方误差

我们分别在不同的分块大小 k 下使用APS算法与DiffHR算法和APB算法来测试已发布数据的均方误差,最终结果如图5所示。



(a)British car accident data set (b)New York Taxi Data set

图5 分块大小与均方误差的曲线

Fig. 5 Curves of block size and mean square error

图5给出了在两种不同数据集的情况下,APS算法、APB算法和DiffHR算法的分块大小与均方误差之间的关系。由图5可知,3种方法中只有APS算法的均方误差是随着分块 k 的增加而降低的。其原因是,分块 k 增大,会导致近似统计值变化变小,从而导致总误差降低。当滑动分块窗口

个数过小时，APS 算法和 APB 算法拥有相同的均方误差。APB 算法和 DiffHR 算法的均方误差与分块大小没有直接关系，因为没有使用到滑动窗口近似统计误差。APS 算法的均方误差明显低于 APB 算法和 DiffHR 算法，这是因为 APS 算法利用滑动窗口近似统计值与加噪值进行比较，进而避免了在滑动窗口中直接添加噪音而带来更大的误差。

6.4 算法总运行时间分析

本节在不同滑动窗口 W 条件下以及不同滑动窗口分块个数 k 的条件下在英国车祸数据集上测试 APS 算法、DiffHR 算法和 APB 算法的总运行时间。

(1)滑动窗口 W 与总运行时间

固定滑动窗口分块 $k=200$，分别在不同的滑动窗口 W 下，利用 APS 算法、DiffHR 算法和 APB 算法来测试已发布数据的总时间，最终结果如图 6 所示。

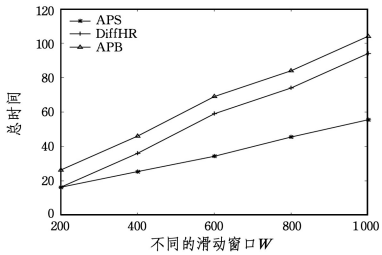


图 6 滑动窗口与总时间的曲线

Fig. 6 Curves of window size and total time

图 6 给出了在纽约黄色出租车行程数据集的情况下，APS 算法、DiffHR 算法和 APB 算法的滑动窗口大小与总运行时间之间的关系。由图 6 可知，APS 算法、DiffHR 算法和 APB 算法的总运行时间随着滑动窗口大小 W 的增加而增加，这是因为随着滑动窗口的增加，3 种算法所处理的数据会越来越多，从而导致总运行时间增加。其中，APB 算法的总运行时间高于 APS 算法和 DiffHR 算法，这是因为 APB 算法使用了隐私预算分配权重的优化模型，导致总运行时间增加。当滑动窗口大小 W 和滑动窗口分块 k 相等时，APS 算法需要缓存当前窗口内的所有数据，从而使得 APS 算法和 DiffHR 算法拥有相同的运行时间。随着滑动窗口大小 W 的增加，APS 算法使用近似统计算法提升了算法效率，从而导致 APS 算法的总运行时间短于 DiffHR 算法。

(2)滑动窗口分块个数 k 与总运行时间

固定滑动窗口大小为 1 000，分别在不同的分块个数 k 下，利用 APS 算法、DiffHR 算法、APB 算法来测试已发布数据的总时间，最终结果如图 7 所示。

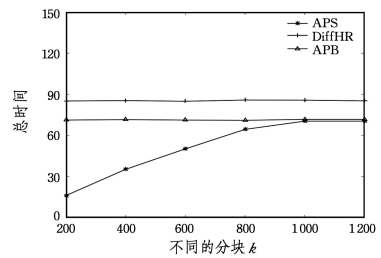


图 7 分块个数与总时间的曲线

Fig. 7 Curves of window block and total time

图 7 给出了在纽约黄色出租车行程数据集的情况下，APS 算法、DiffHR 算法和 APB 算法的滑动窗口分块大小 k 与总运行时间的关系。由图 7 可知，DiffHR 算法和 APB 算法的总运行时间是不同的恒定时间，与 k 无关，这是因为两种算法在每一时刻都需要缓存整个滑动窗口内的数据。APB 算法使用了隐私预算分配权重的优化模型，导致总运行时间增加。APS 算法的总运行时间随着滑动窗口分块 k 的增加而增加，这是因为 APS 算法采用近似统计的思想，可以快速处理数据，进而使得时间开销小于 DiffHR 算法和 APB 算法。当滑动窗口大小 W 小于等于分块个数 k 时，ANM 算法使用了精确计数，从而导致 APS 算法和 DiffHR 算法具有相同的总运行时间。

结束语 本文研究了静态数据和动态数据的隐私保护直方图发布，并依据贪心分组策略和基于滑动窗口的近似统计思想，提出了一种基于滑动窗口自适应直方图发布方法。APS 算法首先利用滑动窗口区间估计算法对下一时间戳的滑动窗口内的区间数据进行预测；然后采用自适应加噪方法，在噪音值与近似误差值之间选取一个合适的值来进行添加；最后对当前时刻的直方图数据采用优化分组的误差下限方式来降低每组桶之间的误差。我们从理论上证明了 APS 算法比现有算法使用了更小的空间。我们在两种不同的真实数据集上进行了实验，结果表明 APS 算法可以在提升运行效率的同时，减小噪声误差对数据的影响，提高发布数据的可用性。

本文未来要考虑两个方向上的研究：1)如何设计针对衰减窗口模型的数据流直方图发布方法；2)如何在分布式场景中，设计针对滑动窗口模型的数据流直方图发布方法。

参 考 文 献

[1] FAN L, BONOMI L, XIONG L, et al. Monitoring web browsing behavior with differential privacy[C]// Proceedings of the 23rd International Conference on World Wide Web. 2014: 177-188.

[2] CELA A, JURIK T, HAMOUCHE R, et al. Energy optimal real-time navigation system[J]. IEEE Intelligent Transportation Systems Magazine, 2014, 6(3): 66-79.

[3] ALMADANI B, SAEED B, ALROUBAIY A. Healthcare systems integration using real time publish subscribe(RTPS) middleware[J]. Computers & Electrical Engineering, 2016, 50(C): 67-78.

[4] DWORK C. Differential privacy [C]// Proceedings of the 33rd International Conference on Automata, Languages and Programming-Volume Part II. Springer-Verlag, 2006: 1-12.

[5] LUO Z, WU D J, ADELI E, et al. Scalable Differential Privacy With Sparse Network Finetuning [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 5059-5068.

[6] YANG W, SUN Y E, HUANG H, et al. Persistent transportation traffic volume estimation with differential privacy[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(1): 213-231.

[7] FICEK J, WANG W, CHEN H, et al. Differential privacy in health research: A scoping review[J]. Journal of the American Medical Informatics Association, 2021, 28(10): 2269-2276.

[8] SUN Z, WANG Y, SHU M, et al. Differential privacy for data and model publishing of medical data[J]. IEEE Access, 2019, 7: 152103-152114.

[9] FAN L, XIONG L, SUNDERAM V. Differentially private multi-dimensional time series release for traffic monitoring[C]// IFIP Annual Conference on Data and Applications Security and Privacy. Berlin: Springer, 2013: 33-48.

[10] LIU J Q, ZHANG C, FANG Y. Epic: A differential privacy framework to defend smart homes against internet traffic analysis[J]. IEEE Internet of Things Journal, 2018, 5(2): 1206-1217.

[11] LI H R, XIONG L, JIANG X Q. Differentially private histogram and synthetic data publication [M] // Medical Data Privacy Handbook. Cham: Springer, 2015: 35-58.

[12] QU J J, CAI Y, XIA H K. A survey of differential privacy protection research for dynamic data release[J]. Journal of Beijing University of Information Science and Technology (Natural Science Edition), 2019, 34(6): 30-36.

[13] XU J, ZHANG Z, XIAO X, et al. Differentially private histogram publication[J]. The VLDB Journal, 2013, 22(6): 797-822.

[14] ACS G, CASTELLUCCIA C, CHEN R. Differentially private histogram publishing through lossy compression[C]// IEEE the 12th International Conference on Data Mining. IEEE, 2012: 1-10.

[15] ZHANG X J, SHAO C, MENG X F. Accurate histogram release under differential privacy [J]. Journal of Computer Research and Development, 2016, 53(5): 1106-1117.

[16] TANG H X, YANG G, BAI Y L. Histogram publishing algorithm based on adaptive differential privacy budget allocation strategy[J]. Application Research of Computers, 2020, 37(7): 1952-1957, 1963.

[17] CHAN T H H, SHI E, SONG D. Private and continual release of statistics[J]. ACM Transactions on Information and System Security (TISSEC), 2011, 14(3): 1-24.

[18] RASTOGI V, NATH S. Differentially private aggregation of distributed time-series with transformation and encryption [C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. 2010: 735-746.

[19] LI H F, LEE S Y, SHAN M K. Online mining (recently) maximal frequent itemsets over data streams[C]// The 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications (RIDE-SDMA'05). IEEE, 2005: 11-18.

[20] NGUYEN H L, WOON Y K, NG W K. A survey on data stream clustering and classification [J]. Knowledge and Information Systems, 2015, 45(3): 535-569.

[21] LIN F P, WU Y J, WANG Y L, et al. Differentially private statistical publication for two-dimensional data stream [J]. Journal of Computer Applications, 2015, 35(1): 88-92.

[22] ZHANG X J, MENG X F. Streaming histogram publication method with differential privacy [J]. Journal of Software, 2016, 27(2): 381-393.

[23] LI Y, LI S. Research on Differential Private Streaming Histogram Publication Algorithm[C]// 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, 2018: 598-603.

[24] GAO R C, MA X B. Dynamic data histogram publishing based on differential privacy[C]// 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). IEEE, 2018: 737-743.

[25] WU X, TONG N, YE Z, et al. Histogram Publishing Algorithm Based on Sampling Sorting and Greedy Clustering[C]// International Conference on Blockchain and Trustworthy Systems. Springer, Singapore, 2019: 81-91.

[26] SCOTT D W. Histogram[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(1): 44-48.

[27] CHEN Z, ZHANG A. A survey of approximate quantile computation on large-scale data [J]. IEEE Access, 2020, 8: 34585-34597.

[28] YANG L, CAO J, YUAN Y, et al. A framework for partitioning and execution of data stream applications in mobile cloud computing [J]. ACM SIGMETRICS Performance Evaluation Review, 2013, 40(4): 23-32.

[29] LI Y, ORGERIE A C, RODERO I, et al. End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT [J]. Future Generation Computer Systems, 2018, 87: 667-678.

[30] TANG Y, GEDIK B. Autopipelining for data stream processing [J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 24(12): 2344-2354.

[31] KOLAJO T, DARAMOLA O, ADEBIYI A. Big data stream analysis: a systematic literature review[J]. Journal of Big Data, 2019, 6(1): 1-30.

[32] GAROFALAKIS M, GEHRKE J, RASTOGI R. Data Stream Management: Processing High-Speed Data Streams [J/OL]. http://www.springer.com/?SGWID=0-102-1297-72039114-0.

[33] CHAN T H H, LI M, SHI E, et al. Differentially private continual monitoring of heavy hitters from distributed streams[C]// International Symposium on Privacy Enhancing Technologies Symposium. Berlin: Springer, 2012: 140-159.

[34] LI C, MIKLAU G, HAY M, et al. The matrix mechanism: optimizing linear counting queries under differential privacy[J]. The VLDB Journal, 2015, 24(6): 757-781.

WANG Xiu-jun, born in 1983, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include RFID system tag management and data stream processing.

(责任编辑:喻藜)