中国科学: 信息科学 2021 年 第 51 卷 第 7 期: 1199–1216
SCIENTIA SINICA Informationis
论文

面向数据流发布的数据自适应隐私保护机制

王腾[1], 杨新宇[1], 任雪斌[1*], 赵俊[2]

1. 西安交通大学计算机科学与技术学院, 西安 710049, 中国
2. School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore
* 通信作者. E-mail: xuebinren@mail.xjtu.edu.cn

收稿日期: 2020–03–31; 修回日期: 2020–05–22; 接受日期: 2020–06–08; 网络出版日期: 2021–06–22

国家自然科学基金 (批准号: 61772410, 61802298, U1811461)、中国博士后基金 (批准号: 2017M623177) 和中央高校基本科研业务费 (批准号: xjj2018237) 资助项目

摘要 群智感知系统中针对数据流的实时发布和深度学习在极大方便人们日常生活的同时, 也严重威胁了参与用户的隐私信息. 现有隐私保护机制在处理动态性强、时空相关性复杂的数据流时, 大都难以实现数据自适应性, 从而导致较低的数据效用性. 因此, 基于 ω- 事件级差分隐私, 本文提出了一种数据自适应的多维数据流隐私保护实时发布机制 AdaPub. 该机制通过集成基于多重哈希的维度划分策略和自适应累积回溯时间聚类策略分别学习数据流的空间和时间相关性, 不需要预定义任何参数, 能够根据数据流的动态变化趋势来自适应地调整隐私参数, 从而保证了隐私保护机制的数据自适应性并有效提高了数据效用性. 此外, 本文进一步提出了一种面向层次数据流发布的隐私保护机制 HierAdaPub, 利用最优隐私预算分配策略来最小化扰动方差以保证数据效用性. 大量仿真实验从不同角度均验证了所提出隐私保护机制能够在提供强隐私保护的同时, 具有较高的数据效用性.

关键词 数据流发布, 数据自适应, 差分隐私, 时空相关性, 数据效用性

1 引言

随着物联网和大数据时代的到来, 群智感知系统 (crowdsensing systems) 利用先进感知技术来全方位地获取和发布物理世界的海量数据流, 从而促进了群智感知应用的发展和普及 [1]. 如图 1 所示, 产生于智能服务应用、Web 点击访问、实时监测等动态环境中的数据流, 经过大数据挖掘和分析处理之后, 能够向用户返回相应的服务与推荐, 从而方便人们的日常生活 [2]. 然而, 数据流发布却给用户带来了前所未有的隐私威胁 [3,4]. 例如, GPS 数据流、交通流量数据流、健康监测数据流等会隐含参与用户的位置信息、日常行为, 甚至是生理特征等信息. 一旦这些数据遭到暴露或滥用, 都会严重威胁用户的隐私信息, 甚至是生命财产安全 [5,6]. 更糟糕的是, 攻击者通过分析数据流的空间和时间关联性, 可

引用格式: 王腾, 杨新宇, 任雪斌, 等. 面向数据流发布的数据自适应隐私保护机制. 中国科学: 信息科学, 2021, 51: 1199–1216, doi: 10.1360/SSI-2020-0076
Wang T, Yang X Y, Ren X B, et al. Data-adaptive privacy-preserving mechanism for data stream publishing in real-time (in Chinese). Sci Sin Inform, 2021, 51: 1199–1216, doi: 10.1360/SSI-2020-0076

© 2021《中国科学》杂志社

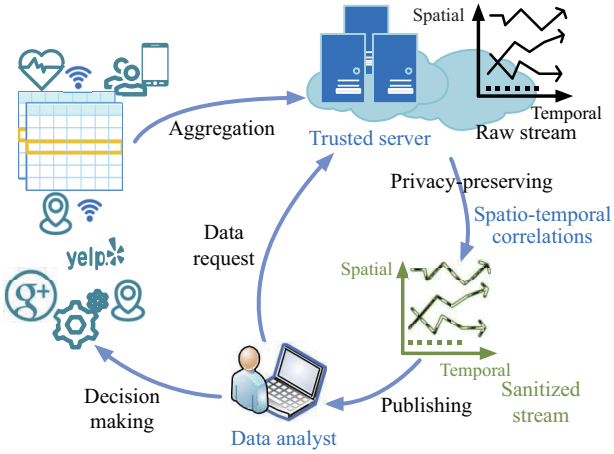


图 1 (网络版彩图) 群智感知系统中感知数据流实时统计发布示意图

Figure 1 (Color online) A general architecture for privacy-preserving stream aggregation and publishing

以获取连续发布数据流中潜在的敏感信息, 且随着时间的增加, 隐私泄露风险也会逐渐累积 [7~9]. 因此, 设计面向数据流实时发布的隐私保护机制已受到众多研究者的广泛关注.

差分隐私 (differential privacy, DP) [10,11] 是目前被广泛接受的一种隐私保护范式. 在数据流发布中应用差分隐私时有两种情况 [12,13]: 事件级 (event-level) 隐私和用户级 (user-level) 隐私. 前者为用户的单独事件提供隐私保证, 而后者能为用户整个时间戳上所有事件提供隐私保证. 此外, 弱隐私保护程度的事件级隐私可以应用于无限流场景中, 而强隐私保护的用户级隐私却只能应用于有限流场景中. 为了实现事件级隐私与用户级隐私之间的良好折衷, Kellaris 等 [14] 提出了 ω- 事件级隐私保护模型, 可以在无限流场景中为连续 ω 时刻内的任意事件提供隐私保证. 基于差分隐私理论, 最直接的做法就是向每个时刻的数据流添加随机噪声来实现隐私保护. 然而, 由于数据流动态性强且时空关联复杂 [15], 直接添加噪声会导致较大的扰动误差. 为此, 一些研究者提出了 FAST [16], RescueDP [17], PeGaSus [18] 等机制, 通过采样滤波、分组、压缩转换、建模等方法提高数据效用性.

然而, 已有隐私保护机制要么缺乏时空相关性学习, 导致容易遭受滤波攻击 [19,20]; 要么难以保证数据自适应性 [15,21], 导致在实际场景中实用性差. 因此, 本文提出了面向感知数据流实时发布的数据自适应隐私保护机制, 具体贡献如下.

(1) 本文提出了一种数据自适应的数据流隐私保护发布机制 AdaPub, 能够以数据自适应的方式来学习数据流的时空相关性并指导隐私保护操作, 从而在满足 ω- 事件级差分隐私的同时提高发布数据流的效用性. 同时, 本文进一步提出了面向层次聚合数据流实时隐私保护发布的数据自适应机制 HierAdaPub, 利用最优隐私分配策略来最小化噪声方差, 保证了发布数据流的效用性.

(2) 本文分别设计了基于多重哈希的维度划分算法 DimParti 和数据自适应的累积回溯时间聚类算法 AdaCluster 来学习多维数据流的空间和时间相关性. 本文所设计算法不需要依赖额外的预定义参数, 能够根据数据流的动态变化趋势自适应地调整和更新阈值参数, 从而在保证数据流时空相关性的前提下, 减少噪声规模并提高数据效用性.

(3) 本文在真实数据流和合成数据流上对所提出机制进行实验验证. 实验结果从不同角度都表明所设计隐私保护机制具有较高的数据效用性. 此外, 不同数据流上的实验结果还证明了所设计机制能够适用于更广泛的数据流, 且针对稀疏性强、波动程度大的数据流时依旧能够维持有效性.

表 1 展示了本文所设计机制 AdaPub 与已有机制在不同角度的对比情况. 可以看到, AdaPub 机

表 1 本文所设计机制与已有机制对比

Table 1 Comparisons of related methods with our proposed mechanism

Algorithm	Privacy level	Dimension	Stream scenario	Learn spatial/temporal correlation	Data-adaptive
FAST [16]	User-level DP	Single	Finite	No/Yes	No
BD/BA [14]	ω-event DP	Multiple	Infinite	No/Yes	–
RescueDP [17]	ω-event DP	Multiple	Infinite	Yes/Yes	No
PeGaSus [18]	Event-level DP	Multiple	Infinite	No/Yes	No
AdaPub	ω-event DP	**Multiple**	**Infinite**	**Yes/Yes**	**Yes**

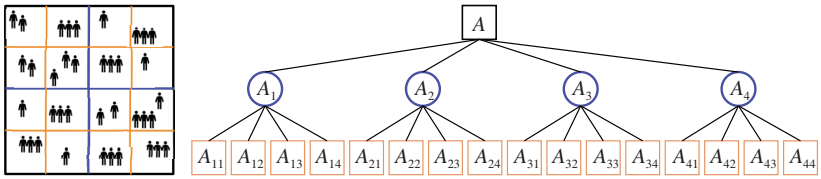


图 2 (网络版彩图) 层次聚合数据流发布示意图

Figure 2 (Color online) An example of hierarchical aggregated stream publishing

制在满足 ω- 事件级差分隐私的前提下, 能够同时学习原始数据流的空间和时间相关性, 并且实现了数据自适应性, 因此会具有更好的实用性.

2 问题描述及基础知识

2.1 问题描述

本文用 D_t^d 表示 t 时刻处于 d 个状态下的 d- 维数据流集合, 其中状态集合表示为 $\mathcal{S}=\{s_1,s_2,\ldots,s_d\}$. 用 U 和 T 分别表示用户和时间集合, 则 D_t^d 中的每一条记录 (u,s,t) 为值域 $U\times\mathcal{S}\times T$ 中的一个原子事件, 表示用户 u 在时刻 t 处于状态 s. 用 $x_t^{s_k}$ 表示在时刻 t 处于状态 s_k 的用户总数, 则 $x_t^{s_k}=|\{(u',s_k{}',t')\in D_t\,|t'=t\wedge s_k{}'=s\}|$. 因此, 时刻 t 的 d- 维感知数据流被表示为 $X_t^d=X_t^{\mathcal{S}}=(x_t^{s_1},x_t^{s_2},\ldots,x_t^{s_d})^{\mathrm{T}}$, d- 维无限感知数据流被表示为 $\boldsymbol{X}^d=\boldsymbol{X}^{\mathcal{S}}=\{X_1^d,X_2^d,\ldots\}$.

此外, 本文进一步研究针对具有层次结构 (hierarchical structure) [18] 的聚合数据流的隐私保护发布问题, 如图 2 所示. 一个聚合 a_i 是状态集合 $\mathcal{S}$ 的子集, 即 $a_i\subseteq\{s_1,s_2,\ldots,s_d\}$. 时刻 t 针对聚合 a_i 的数据流则表示为 $x_t^{a_i}$, 并且满足 $x_t^{a_i}=\sum_{s_i\in a_i}x_t^{s_i}$. 当给定聚合 $A=\{a_1,a_2,\ldots,a_{|A|}\}$ 时, 则 t 时刻的聚合数据流表示为 $X_t^A=(x_t^{a_1},x_t^{a_2},\ldots,x_t^{a_{|A|}})^{\mathrm{T}}$, 并且无限聚合数据流表示为 $\boldsymbol{X}^A=\{X_1^A,X_2^A,\ldots\}$.

综上, 本文的研究问题和目标是: 当给定无限多维数据流 $\boldsymbol{X}^d=\{X_1^d,X_2^d,\cdots\}$ (或层次数据流 $\boldsymbol{X}^A=\{X_1^A,X_2^A,\ldots\}$) 时, 发布满足 ω- 事件级 ϵ- 差分隐私的数据流 $\boldsymbol{R}^d=\{R_1^d,R_2^d,\ldots\}$ ($\boldsymbol{R}^A=\{R_1^A,R_2^A,\ldots\}$), 并保证发布数据流的效用性. 本文利用平均相对误差 (average relative error, ARE) 来衡量原始数据流与隐私保护后的数据流之间的统计距离以体现数据效用性, 定义如下:

$$\mathrm{ARE}(X_t^d,R_t^d)=\frac{1}{d}\sum\nolimits_{i=1}^{d}\frac{\left|x_t^i-r_t^i\right|}{\max\{x_t^i,\delta\}},\tag{1}$$

其中 X_t^d 和 R_t^d 分别表示原始数据流和隐私保护后的数据流, 界限 δ 用来消除零值和极小值的影响.

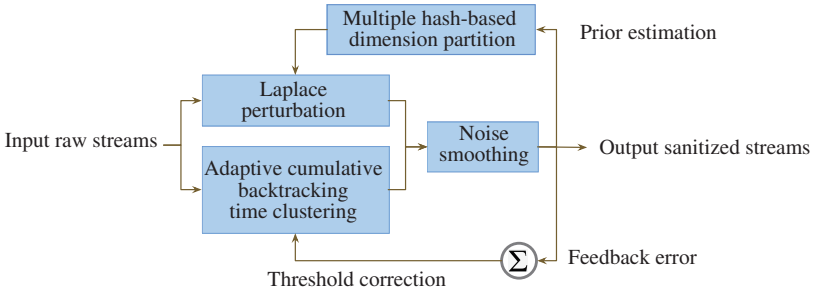


图 3 (网络版彩图) 数据自适应的数据流隐私保护发布机制的框架图

Figure 3 (Color online) The framework of the data-adaptive privacy-preserving mechanism

2.2 差分隐私

用 $\boldsymbol{S}_t = \{D_1, D_2, \ldots, D_t\}$ 表示无限数据流集合 $\boldsymbol{S} = \{D_1, D_2, \ldots\}$ 在 t 时刻的流前缀, 用 D_t 和 D'_t 表示相邻数据集, 则 ω- 事件级 ϵ- 差分隐私 [10, 14] 定义如下.

定义1 (ω- 事件级 ϵ- 差分隐私) 当随机算法 $\mathcal{A}$ 满足 ω- 事件级 ϵ- 差分隐私时, 则对于所有的输出集合 $O \subseteq \mathrm{Range}(\mathcal{A})$, 所有的 ω- 相邻数据流前缀 $\boldsymbol{S}_t$ 和 $\boldsymbol{S}'_t$, 所有的时刻 t, 都满足

$$\Pr[\mathcal{A}(\boldsymbol{S}_t) \in O] \leqslant \mathrm{e}^{\epsilon} \cdot \Pr[\mathcal{A}(\boldsymbol{S}'_t) \in O], \tag{2}$$

其中 Pr 表示概率, ϵ 为隐私预算. 通常, ϵ 越小, 隐私保护程度越高, 但数据效用性越低, 反之亦然.

当给定隐私预算和敏感度之后, 拉普拉斯 (Laplace) 机制 [22] 通过向查询结果添加随机噪声实现差分隐私, 添加的噪声为 $\langle \mathrm{Lap}(\Delta_f/\epsilon) \rangle^d$, 其中敏感度定义为 $\Delta_f = \max_{D_t, D'_t} \|f(D_t) - f(D'_t)\|_1$.

3 面向数据流实时统计发布的数据自适应隐私保护机制

3.1 设计原理及机制概览

图 3 展示了本文提出的数据自适应的隐私保护机制 AdaPub 的基本框架, 主要包括 4 个模块: (1) 基于多重哈希的维度划分; (2) 拉普拉斯扰动; (3) 自适应累积回溯时间聚类; (4) 噪声平滑.

算法 1 展示了 AdaPub 机制的具体工作流程. 首先, AdaPub 机制执行基于多重哈希的维度划分策略 (具体过程见算法 2), 得到划分结果 $\mathcal{P}_t$. 然后, 向 $\mathcal{P}_t$ 中每一个划分 p 中的所有数据流统计值之和添加拉普拉斯噪声, 并计算平均值以得到 p 中每一维度的噪声统计值. 为了实现 ω- 事件级 ϵ_p- 差分隐私, 拉普拉斯噪声的规模为 $\Delta\omega/\epsilon_p$. 由于一条数据记录只影响一个计数值, 因此敏感度 $\Delta = 1$. 当 ω 个时刻内的隐私预算为均匀分配时, 噪声扰动方差的最小值为 $2\omega^2/\epsilon_p^2$, 其中 ϵ_p 是用于拉普拉斯扰动的总预算. 接着, AdaPub 机制执行自适应累积回溯时间聚类算法 (具体过程见算法 3) 来学习数据流的时间相关性. 如算法 1 第 8 行所示, AdaCluster 算法对当前已接收到的数据流所对应的时刻进行聚类, 得到聚类结果 $\mathcal{C}_t^d$. 最后, 基于时间聚类结果 $\mathcal{C}_t^d$, AdaPub 机制对噪声数据流 $\hat{X}_t^d = (\hat{x}_t^1, \hat{x}_t^2, \ldots, \hat{x}_t^d)^{\mathrm{T}}$ 进行中值平滑处理, 并生成最终可以发布的噪声数据流 $R_t^d = (r_t^1, r_t^2, \ldots, r_t^d)^{\mathrm{T}}$.

接下来, 本文详细介绍 AdaPub 机制中的维度划分模块和自适应累积回溯时间聚类模块.

3.2 空间相关性学习: 基于多重哈希的维度划分算法

本小节介绍了一种基于多重哈希的维度划分策略, 能够自适应地将具有相近统计值的维度划分到

Algorithm 1 AdaPub: privately adaptive stream publishing in real-time

Input: $\boldsymbol{X}^d = \{X_1^d, X_2^d, \ldots\}$: d-dimensional infinite stream;
Output: $\boldsymbol{R}^d = \{R_1^d, R_2^d, \ldots\}$: sanitized d-dimensional infinite stream;

for each timestamp t **do**
  Execute multiple hash-based dimension partition: $\mathcal{P}_t = \text{DimParti}(R_{t-1}^d, g)$;
  for each partition p in $\mathcal{P}_t$ **do**
    Compute the sum statistics of stream in p as $X_t^p = \sum_{k \in p} x_t^k$;
    Add Laplace noise: $\hat{X}_t^p \leftarrow X_t^p + \text{Lap}(\Delta\omega/\epsilon_p)$;
    Average to each dimension in p, that is, $\hat{x}_t^k = \hat{X}_t^p / |p|$;
  end for
  Perform adaptive cumulative backtracking time clustering: $\mathcal{C}_t^d \leftarrow \text{AdaCluster}(X_t^d, \epsilon_c, \text{thre}_t^d, \mathcal{C}_{t-1}^d)$;
  $r_t^k = \text{median}\{\hat{x}_j^k \,|j \in \mathcal{C}_t^k\}$ for each dimension $k \in [1, d]$;
  Publish $R_t^d = (r_t^1, r_t^2, \ldots, r_t^d)^{\mathrm{T}}$;
end for

Algorithm 2 DimParti: multiple hash-based dimension partition

Input: The last released stream R_{t-1}^d, space mapping functions $\mathcal{M}$, the number of hash functions g;
Output: Partition results $\mathcal{P}_t = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ of d-dimensional stream;

Obtain the prior estimation: $\bar{X}_t^d = (\bar{x}_t^1, \bar{x}_t^2, \ldots, \bar{x}_t^d)^{\mathrm{T}} = R_{t-1}^d$;
Perform space mapping and obtain d-dimensional g-bit binary vector matrix $V_{d\times g}$, that is, $(\bar{X}_t^d)_{d\times 1} \xrightarrow{\mathcal{M}_{1\times g}} V_{d\times g}$;
Obtain the initial hash table $\mathcal{T}$ by selecting unique value in $V_{d\times g}$;
for each g-bit vector v_i ($i \in [1, d]$) in $V_{d\times g}$ **do**
  Store vector v_i on corresponding bucket of $\mathcal{T}$ based on hash value $\mathcal{H}_g(v_i)$;
  Obtain partition result $\mathcal{P}_t = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ from hash table $\mathcal{T}$;
end for
return $\mathcal{P}_t$;

同一个桶内. 采用多重哈希的核心思想是, 原始数据流空间中相似的数据通过相同的映射 (map) 或投影 (projection) 操作之后, 在新的数据空间中仍然会以很大的概率保持相似, 而本身不相似的数据点被映射到相同桶中的概率则很小. 具体使用的空间映射函数 (space mapping functions) 定义如下.

定义2 (空间映射函数) 给定一组空间映射函数 $\mathcal{M} = \{m_1, m_2, \ldots, m_g\}$, 每个映射函数 m_i ($i \in [1, g]$) 的定义取决于对应的阈值 τ_i ($i \in [1, g]$), 其中阈值 $\tau = \{\tau_1, \tau_2, \ldots, \tau_g\}$ 是在范围 $[0, \text{Range}]$ 上随机选取的服从均匀分布的 g 个随机数. 因此, 任意一个映射函数 $m_i \in \mathcal{M}$ 定义为一个判断输入值 v 是否大于阈值 τ_i 的指示函数, 即 $m_i(v) = \mathbf{1}[v \leqslant \tau_i]$, 其中, $m_i(v)$ 表示对输入值 v 进行映射, Range 表示一组数据流的统计范围, 通常为统计值的最大值.

为了保护隐私和避免再次引入噪声, 本文使用上一时刻的发布值作为当前时刻的先验估计值, 即 $\bar{X}_t^d = R_{t-1}^d = (r_{t-1}^1, r_{t-1}^2, \ldots, r_{t-1}^d)^{\mathrm{T}}$. 当 $t = 1$ 时, 默认将各个维度分别作为一个单独划分, 即 $t = 1$ 时刻的维度划分结果为 $\mathcal{P}_1 = \{p_1, p_2, \ldots, p_d\}$. 故当 $t = 1$ 时, 本质上相当于没有执行基于多重哈希的维度划分算法, 因此不会造成隐私泄露. 算法 2 描述了基于多重哈希的维度划分机制, 主要包括 3 个步骤. (1) 空间映射: 将当前 t 时刻 d- 维数据流的先验估计值 $\bar{X}_t^d$ 中的每一维数据流都映射成 g- 比特二进制向量, 从而生成一个 d- 维二进制矩阵 $V_{d\times g}$. (2) 初始化哈希表: 由于理论上初始哈希表 $\mathcal{T}$ 有 2^g 个桶号, 故本文选择矩阵 $V_{d\times g}$ 中的唯一值作为哈希表中每个桶的桶号, 从而减小哈希表的大小. (3) 基于哈希值进行维度划分: 根据哈希函数族 $\mathcal{H}_g$ 对矩阵 $V_{d\times g}$ 中每一条 g- 比特向量 v_i ($i \in [1, d]$) 进行哈希计算, 基于哈希值 $\mathcal{H}_g(v_i)$ 将向量 v_i 存储到哈希表 $\mathcal{T}$ 对应的桶中, 得到 d- 维感知数据流的

Algorithm 3 AdaCluster: adaptive cumulative backtracking time clustering

Input: d-dimensional stream $X_t^d = (x_t^1, x_t^2, \ldots, x_t^d)^{\mathrm{T}}$ at time t, privacy budget ϵ_c;
Output: Clustering result $\mathcal{C}_t^d$ of d-dimensional stream at time t;

for each dimension $k \in [1, d]$ **do**
  Compute the corrected threshold thre_t^k;
  if $t = 1$ **then**
    Compute $\mathcal{C}_t^k = \{t\}$ and set $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$;
  else
    if $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$ **then**
      Compute deviation value $\mathrm{dev}_t^k = f_{\mathrm{dev}}(X[\mathcal{C}_{t-1}^k \cup \{t\}])$;
      if $\mathrm{dev}_t^k + \mathrm{Lap}(\Delta_{\mathrm{dev}} \cdot \omega/\epsilon_c) < \mathrm{thre}_t^k$ **then**
        Compute $\mathcal{C}_t^k = \mathcal{C}_{t-1}^i \cup \{t\}$ and set $\mathcal{C}_{\mathrm{state}}^k = \mathrm{close}$;
      else
        Compute $\mathcal{C}_t^k = \{t\}$ and set $\mathcal{C}_{\mathrm{state}}^k = \mathrm{close}$;
      end if
    else
      Compute $\mathcal{C}_t^k = \{t\}$ and set $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$;
    end if
  end if
end for
$\mathcal{C}_t^d = \{\mathcal{C}_t^k\}^{\mathrm{T}}, k \in [1, d]$;
return $\mathcal{C}_t^d$;

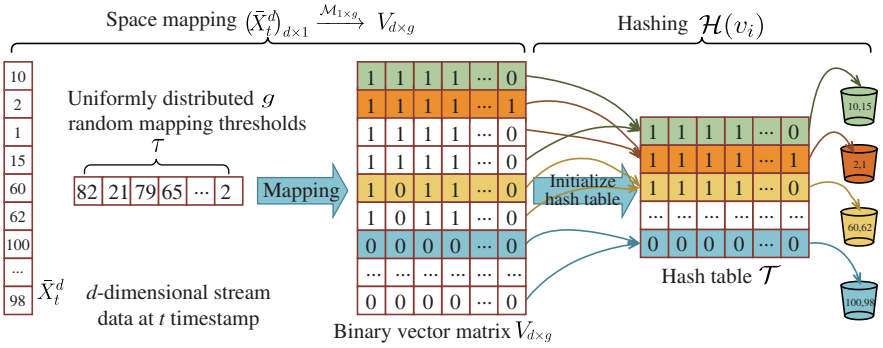


图 4 (网络版彩图) 基于多重哈希的维度划分机制可视化示例图

Figure 4 (Color online) A visualized example of DimParti

维度划分结果 $\mathcal{P}_t = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$.

图 4 展示了基于多重哈希的维度划分的具体过程. t 时刻的 d- 维数据流 $\bar{X}_t^d = (10, 2, \ldots, 98)^{\mathrm{T}}$ 经过空间映射和哈希转换之后, 被划分到不同的桶中. 例如, (10, 15) 在同一桶内, (100, 98) 在同一桶内.

3.3 时间相关性学习: 自适应累积回溯时间聚类算法

本小节介绍了自适应累积回溯时间聚类算法 AdaCluster. 具体地, AdaCluster 算法采用偏差函数来评估聚类结果的合理性. 偏差函数可以测量一组数据与其平均值的均值误差. 用 $\mathcal{C}_t^k$ 表示当前 t 时刻针对第 k ($k \in [1, d]$) 维数据流的时间聚类结果, 则 $\mathcal{C}_t^k$ 中每个时间戳对应的数据流的偏差函数定

义为

$$f_{\mathrm{dev}}(X[\mathcal{C}_t^k]) = \sum\nolimits_{j\in\mathcal{C}_t^k} \left| x_j^k - \frac{\sum_{j\in\mathcal{C}_t^k} x_j^k}{|\mathcal{C}_t^k|} \right|, \tag{3}$$

其中 $X[\mathcal{C}_t^k]$ 表示簇 $\mathcal{C}_t^k$ 中时间戳对应的数据流的集合, $|\mathcal{C}_t^k|$ 表示簇的大小. 一般而言, 当偏差距离 $f_{\mathrm{dev}}(X[\mathcal{C}_t^k])$ 较小时, 则认为 $X[\mathcal{C}_t^k]$ 中数据流接近于均匀分布, 即具有较强的时间相关性.

对于无限 d- 维数据流 $\boldsymbol{X}^d = X_1^d, X_2^d, \ldots$ 而言, 每当时刻 t 的新数据流 X_t^d 到来时, AdaCluster 首先计算当前数据流与上一时刻簇中的数据流之间的偏差距离, 然后将偏差距离与阈值的大小关系作为聚类依据. 算法 3 展示了 AdaCluster 算法的具体过程. 对于数据流的每一维 $k \in [1, d]$, AdaCluster 算法直接将第一时刻的时间戳作为一个簇, 并设置簇状态为 open, 即 $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$. 簇状态为 open 时表示当前簇还可以继续加入新的时间戳. 当新数据流 X_t^d 到来时, 存在以下两种情况.

(1) 当 $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$ 时, 计算当前时刻数据流与上一簇中时间戳对应数据流的偏差距离. 为了实现隐私保护, 对偏差距离添加拉普拉斯噪声, 其中偏差函数的敏感度为 $\Delta_{\mathrm{dev}} = 2$ [18] 且隐私预算为 ϵ_c/ω, 其中 ϵ_c 是用于聚类的总预算. 如果噪声偏差距离小于阈值 thre_t^k, 则当前时刻的簇为上一时刻的簇与当前时刻 t 的并集, 即 $\mathcal{C}_t^k = \mathcal{C}_{t-1}^k \cup \{t\}$, 且设置簇状态为 open. 反之, 则将当前时刻作为一个新的簇, 并设置簇状态为 close, 即 $\mathcal{C}_t^k = \{t\}$, 且 $\mathcal{C}_{\mathrm{state}}^k = \mathrm{close}$.

(2) 当 $\mathcal{C}_{\mathrm{state}}^k = \mathrm{close}$ 时, 直接将当前时刻 t 作为一个新的簇, 并设置当前簇状态为 open, 即 $\mathcal{C}_t^k = \{t\}$, 且 $\mathcal{C}_{\mathrm{state}}^k = \mathrm{open}$ (如算法 3 第 13 和 14 行所示).

为了保证数据自适应性, 本文通过计算 PID 反馈误差 [16] $\Delta\mathrm{err}_t^k$ 并结合隐私预算 ϵ 自适应地更新阈值. 当 PID 误差增加或当隐私预算较小时, 则可以推断此时的扰动误差相对较大. 相应地, 此时应该增大阈值从而增加簇的大小, 从而平滑掉过量的噪声. 因此, 第 k 维数据流在时刻 t 的阈值更新规则为 $\mathrm{thre}_t^k = \max\{1, (\Delta\mathrm{err}_t^k)^2/\epsilon\}$, 其中, 第一时刻的阈值被初始化为 $\mathrm{thre}_t^k = 1/\epsilon$.

4 面向层次数据流实时统计发布的数据自适应隐私保护机制

本节考虑层次数据流的隐私保护发布问题. 针对层次数据流发布时, 最直接的做法就是将 AdaPub 机制应用到层次聚合数据流中. 但是这种做法没有考虑到层次数据流的稀疏性和层次树的结构特性. 一方面, 层次数据流的某些叶子节点的值非常小, 即稀疏性强, 直接向这些叶子节点添加噪声时会导致较大的扰动误差, 破坏层次数据流的一致性. 另一方面, 考虑到层次数据流的高度、宽度 (层次树每层的节点数) 等结构特性, 直接将隐私预算平均分配给每一层是不合理的.

为此, 本文提出数据自适应的隐私保护机制 HierAdaPub, 将层次数据流的每一层看作是多维数据流, 然后应用基于多重哈希的维度划分算法来克服稀疏性带来的影响, 并保证每层节点间的相关性. 此外, 针对层次数据流的结构特性, HierAdaPub 利用最优隐私预算分配策略来最小化扰动误差.

令 μ_i ($i \in [1, H]$) 表示层次数据流第 i 层的权重 (即节点数). 假设用于扰动的总隐私预算为 ϵ_A. 直接应用 AdaPub 机制时, 层次树每层的隐私预算为 ϵ_A/H, 故拉普拉斯扰动的总方差为 $\sum_{i=1}^{H} \mu_i \cdot \frac{H^2}{\epsilon_A^2}$. 由于在分配隐私预算时忽略了层次数据流的结构特性, 因此总方差并不是最优的. 一个最优的隐私预算分配策略应该考虑层次数据流的结构特性. 假设层次树第 i 层分配的隐私预算为 ϵ^i, 因此总方差为 $\mathbb{V} = \frac{\mu_1}{(\epsilon^1)^2} + \frac{\mu_2}{(\epsilon^2)^2} + \cdots + \frac{\mu_H}{(\epsilon^H)^2}$, 且当 $\epsilon^1 = \frac{\epsilon_A}{\sum_i^H (\mu_i/\mu_1)^{\frac{1}{3}}}$ 和 $\epsilon^i = (\mu_i/\mu_1)^{\frac{1}{3}} \cdot \epsilon^1$ 时取得最小值.

算法 4 展示了 HierAdaPub 机制的具体过程. 首先计算层次树每一层的最优隐私预算 ϵ^i. 然后, 将层次树每一层所有节点看作是多维数据流, 并应用 AdaPub 机制对每一层的多维数据流进行处理. 在

Algorithm 4 HierAdaPub: data-adaptive framework of hierarchical streams publishing in real-time

Input: Raw hierarchical streams $\boldsymbol{X}^A = \{X_1^A, X_2^A, \ldots\}$, privacy budget ϵ_A, the weight μ of hierarchical tree;
Output: Sanitized hierarchical streams $\boldsymbol{R}^A = \{R_1^A, R_2^A, \ldots\}$;

```
Compute the optimal privacy budget allocation ε^i, i ∈ [1, H];
for each time t do
    Initialize R_t^A = ∅;
    for each level i ∈ H do
        Obtain μ_i-dimensional stream X_t^{μ_i} of the ith level based on weight μ;
        ε_p^i = ε_i, ε_c^i = ε_c/H;
        Perform R_t^{μ_i} =AdaPub(X_t^{μ_i}, ε_p^i, ε_c^i, ω);
        R_t^A = R_t^A ∪ R_t^{μ_i};
    end for
    Publish R_t^A;
end for
```

每一时刻 t, 初始化隐私保护后的数据流集合 $R_t^A = \emptyset$. 然后基于权重 μ, 得到每一层的 μ_i- 维感知数据流 $X_t^{\mu_i}$. 接着, 直接将 AdaPub 机制应用到 μ_i- 维感知数据流 $X_t^{\mu_i}$ 上, 即 $R_t^{\mu_i} = \text{AdaPub}(X_t^{\mu_i}, \epsilon_p^i, \epsilon_c^i, \omega)$, 其中用于扰动和聚类的隐私预算分别为 $\epsilon_p^i = \epsilon_i$ 和 $\epsilon_c^i = \epsilon_c/H$. 最后, 更新隐私保护后的数据流集合, 即 $R_t^A = R_t^A \cup R_t^{\mu_i}$, 并实时发布隐私保护后的层次数据流 R_t^A.

5 隐私性和复杂性分析

5.1 隐私性分析

定理1 算法 1 中拉普拉斯扰动过程满足 ω- 事件级 ϵ_p- 差分隐私.

证明 根据算法 1 中第 5 行, 每个时刻的隐私预算为 ϵ_p/ω. 根据差分隐私并行组合定理可知, 算法 1 中拉普拉斯扰动过程满足 ϵ_p/ω- 差分隐私. 任意 ω 时刻内都有 $\sum_{t-\omega+1}^{t} \epsilon_{p_t} = \epsilon_p$, 其中 $\epsilon_{p_t} = \epsilon_p/\omega$. 因此, 根据差分隐私顺序组合定理可知, 算法 1 中拉普拉斯扰动过程满足 ω- 事件级 ϵ_p- 差分隐私.

定理2 算法 3 (即 AdaCluster 算法) 满足 ω- 事件级 ϵ_c- 差分隐私.

证明 用 $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_d\}$ 表示算法 3, 且 $\mathcal{A}_i$ 表示对第 i 维数据流应用 AdaCluster 机制. 因此, 当应用 $\mathcal{A}$ 到 d- 维数据流时, 则有

$$\frac{\Pr[\mathcal{A}(D_t) \in O_t]}{\Pr[\mathcal{A}(D'_t) \in O_t]} \leqslant \frac{\Pr[\mathcal{A}_1(D_t) \in O_t]}{\Pr[\mathcal{A}_1(D'_t) \in O_t]} \cdot \cdots \cdot \frac{\Pr[\mathcal{A}_d(D_t) \in O_t]}{\Pr[\mathcal{A}_d(D'_t) \in O_t]}. \tag{4}$$

由于一个用户在每个时刻最多处于一种状态, 因此对 t 时刻的 d 维相邻数据集 D_t 和 D'_t 进行查询时满足

$$\exists i \in [1,d],\ \frac{\Pr[\mathcal{A}_i(D_t) \in O_t]}{\Pr[\mathcal{A}_i(D'_t) \in O_t]} \neq 1 \ \text{ 且 } \forall j \neq i\ (j \in [1,d]),\ \frac{\Pr[\mathcal{A}_j(D_t) \in O_t]}{\Pr[\mathcal{A}_j(D'_t) \in O_t]} = 1. \tag{5}$$

由算法 3 可知, 对每维数据流应用 AdaCluster 机制都满足 ϵ_c/ω- 差分隐私. 由差分隐私定义可知, 对于 t 时刻的相邻数据集 D_t 和 D'_t 以及输出结果 O_t 而言, 针对每个维度的算法 $\mathcal{A}_i$ ($i \in [1,d]$) 都满足

$$\Pr[\mathcal{A}_i(D_t) \in O_t] \leqslant \mathrm{e}^{\epsilon_c/\omega} \cdot \Pr[\mathcal{A}_i(D'_t) \in O_t]. \tag{6}$$

合并式 (4)∼(6) 可得 $\Pr[\mathcal{A}(D_t) \in O_t] \leqslant \mathrm{e}^{\epsilon_c/\omega} \cdot \Pr[\mathcal{A}(D'_t) \in O_t]$. 故算法 3 满足 ϵ_c/ω- 差分隐私.

表 2 实验所用数据集基本信息

Table 2 Datasets used in experiments

Dataset	Length	Dimension	Total count	Average count	Value level	Sparsity	Fluctuation
Flu	427	1	9787	22.92	Median	0.0679	Low
SateFlu	389	51	5599045	282.22	High	0.0364	Median
Uber	212	32	2312991	340.95	High	0.0015	High
OnlineRetail	374	1298	4423432	9.11	Low	0.6841	High
Syn	500	100	24580115	491.60	High	0.0097	High

此外, 算法 3 保证在任意 ω 个时刻都有 $\sum_{t-\omega+1}^{t}\epsilon_{c_t}=\epsilon_c$, 其中 $\epsilon_{c_t}=\epsilon_c/\omega$. 因此, 根据差分隐私顺序组合定理可知, 算法 3 满足 ω- 事件级 ϵ_c- 差分隐私.

定理3 算法 1 (即 AdaPub 机制) 满足 ω- 事件级 ϵ- 差分隐私.

证明 根据算法 1, AdaPub 机制只在第 5∼8 行消耗隐私预算. 根据定理 1 和 2 以及差分隐私顺序组合定理可知, 在任意 ω 时刻内总的隐私预算为 $\epsilon=\epsilon_p+\epsilon_c$. 因此, 基于差分隐私后处理原则 [23] 可知, 算法 1 满足 ω- 事件级 ϵ- 差分隐私.

定理4 算法 4 (即 HierAdaPub 机制) 满足 ω- 事件级 ϵ- 差分隐私.

证明 算法 4 只在第 7 行消耗隐私预算, 其中在任意 ω 时刻内用于扰动和聚类的隐私预算分别是 ϵ_p^i 和 $\epsilon_c^i=\epsilon_c/H$. 基于差分隐私顺序组合定理可知, 任意 ω 时刻内用于扰动和聚类总隐私预算分别是 $\epsilon^1+\epsilon^2+\cdots+\epsilon^H=\epsilon_A$ 和 ϵ_c. 因此, 算法 4 满足 ω- 事件级 ϵ- 差分隐私.

5.2 时空复杂度分析

根据算法 1, 在每个时刻, 拉普拉斯扰动和噪声平滑的时间复杂度分别是 $O(|\mathcal{P}|_m)$ 和 $O(d)$, 其中 $|\mathcal{P}|_m$ 是维度划分结果的集合大小且远小于 d. 此外, DimParti 模块的时间复杂度是 $O(dg)$, 其中 g 是映射函数的个数且小于 d. AdaCluster 模块需要计算大小为 $|\mathcal{C}|_m$ 的回溯簇的偏差距离, 故时间复杂度是 $O(d|\mathcal{C}|_m)$, 其中 $|\mathcal{C}|_m$ 是整个数据流上最大回溯簇的大小. 综上可知, AdaPub 机制的总时间复杂度为 $O(|\mathcal{P}|_m)+O(d)+O(dg)+O(d|\mathcal{C}|_m)=O(d|\mathcal{C}|_m+dg)$. 且 HierAdaPub 机制的时间复杂度为 $O(H\cdot(\mu_m|\mathcal{C}|_m+\mu_m g))$, 其中 H 是层次数据流的高度, μ_m 是层次树各个层中的最大节点数.

对于空间复杂度, 本文所设计隐私保护机制不需要存储整个时间维度上的数据流, 因此降低了空间复杂度. 在任意时刻 t, 算法 1 只需要存储维度划分结果 $\mathcal{P}$、d- 维噪声数据流 $\hat{X}$ 和各个维度的聚类结果 $\mathcal{C}$. 因此, 算法 1 (即 AdaPub 机制) 的总空间复杂度为 $O(|\mathcal{P}|_m)+O(d)+O(d|\mathcal{C}|_m)=O(d|\mathcal{C}|_m)$. 基于上述分析可知, 算法 4 (即 HierAdaPub 机制) 的总空间复杂度为 $O(H\cdot(\mu_m|\mathcal{C}|_m))$.

6 实验评估

本节在如表 2 所示的数据集上对所设计隐私机制进行评估. Flu[1)] 数据流是对由于患流感而死亡的人数的监测统计, StateFlu[1)] 数据流是对美国 51 个地区患流感人数的周统计数据, Uber[2)] 数据流是对 Uber 基站在 212 天内的路线统计数据. OnlineRetail[3)] 是对在线销售商店 1289 个物品的日销售量统计值. Syn 数据集根据函数 $x_t=x_{t-1}+\mathcal{N}(0,20)$ 而产生, 其中 $x_1=500$, $\mathcal{N}$ 表示高斯 (Gauss) 分布.

1) Flu Dataset. http://www.cdc.gov/flu/.

2) Uber Dataset. Kaggle. https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

3) OnlineRetail Dataset. UCI Machine Learning Repository. https://archive.ics.uci.edu/.

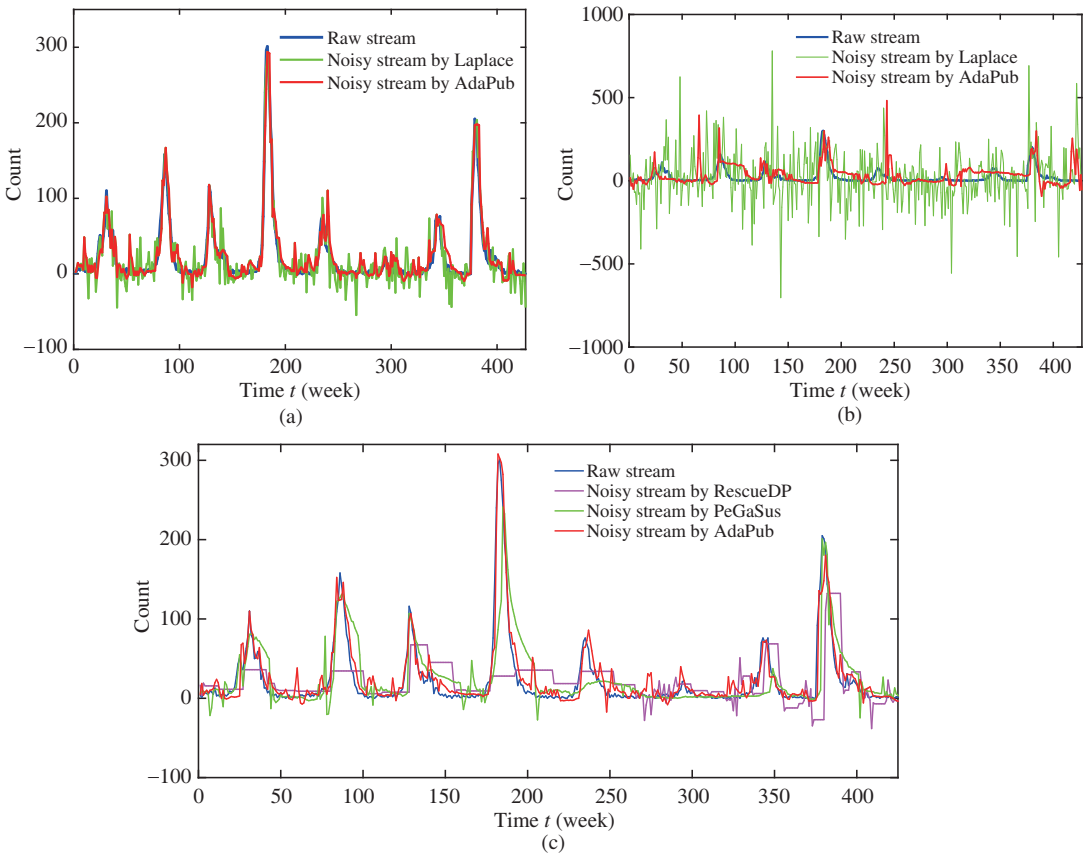


图 5 (网络版彩图) 隐私保护机制在 **Flu** 数据流上的可视化展示 ($\epsilon = 1$)

Figure 5 (Color online) Visualizations of the protection for Flu stream ($\epsilon = 1$). (a) $\omega = 10$; (b) $\omega = 100$; (c) $\omega = 10$

在所有实验中, 本文使用平均相对误差来衡量隐私保护机制的准确性. 默认情况下, 本文设置 $\epsilon_p = 0.8\epsilon$, $\epsilon_c = 0.2\epsilon$ 且 $g = 20$, PID 参数设置为 $(K_p, K_i, K_d) = (0.9, 0.1, 0)$ 和 $\eta = 5$. 此外, 本文设置 δ 为每维数据流计数之和 $\sum_{t=1}^{T} x_t^i$ 的 1%. 其他机制的参数都按照对应的文章进行最优选择.

6.1 数据流隐私保护实时发布的可视化展示

图 5(a) 和 (b) 分别展示了不同隐私保护级别下, AdaPub 机制与 Laplace 机制的可视化效果图. 可以看到, 相较于 Laplace 机制, 即使当隐私保护程度较高时 (即 $\omega = 100$), AdaPub 机制能够有效保留原始数据流的变化趋势. 图 5(c) 展示了 AdaPub 机制相较于 RescueDP 和 PeGaSus 能够更加准确地保留原始数据流的动态变化趋势. 这是因为 RescueDP 和 PeGaSus 在处理数据流时都需要依赖固定的阈值参数, 而固定的阈值参数可能难以适用于动态变化的实际数据流. 相较之下, AdaPub 机制不依赖任何额外参数, 故随时间变化时能够始终保证原始数据流的动态特性.

6.2 隐私保护机制效用性的评估

图 6 展示了 DimParti 机制中参数 g 的大小对效用性的影响, 其中 g 为哈希函数的个数. 可以看到当 $g = 20$ 时, 平均相对误差都几乎达到最小值. 因此, 在后续实验中, 本文默认设置 $g = 20$. 图 7 展示了使用维度划分机制 DimParti 时可以有效降低发布数据流的平均相对误差. 图 8 和 9 分别展示

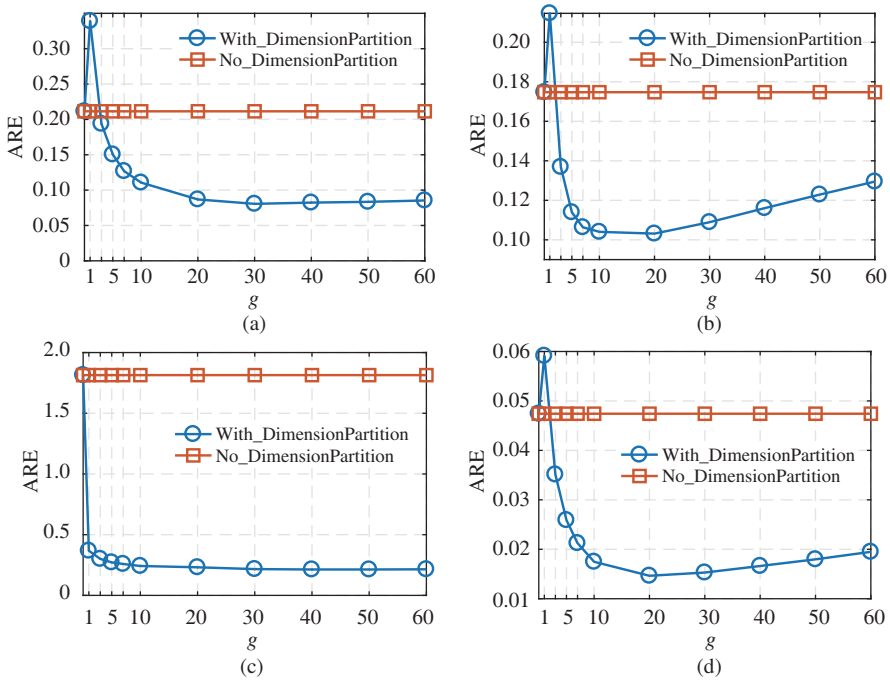


图 6 (网络版彩图) 参数 g 的大小对效用性的影响 ($\epsilon = 1, \omega = 100$)

Figure 6 (Color online) Utility comparisons of the choice of g in DimParti ($\epsilon = 1, \omega = 100$). (a) StateFlu; (b) Uber; (c) OnlineRetail; (d) Syn

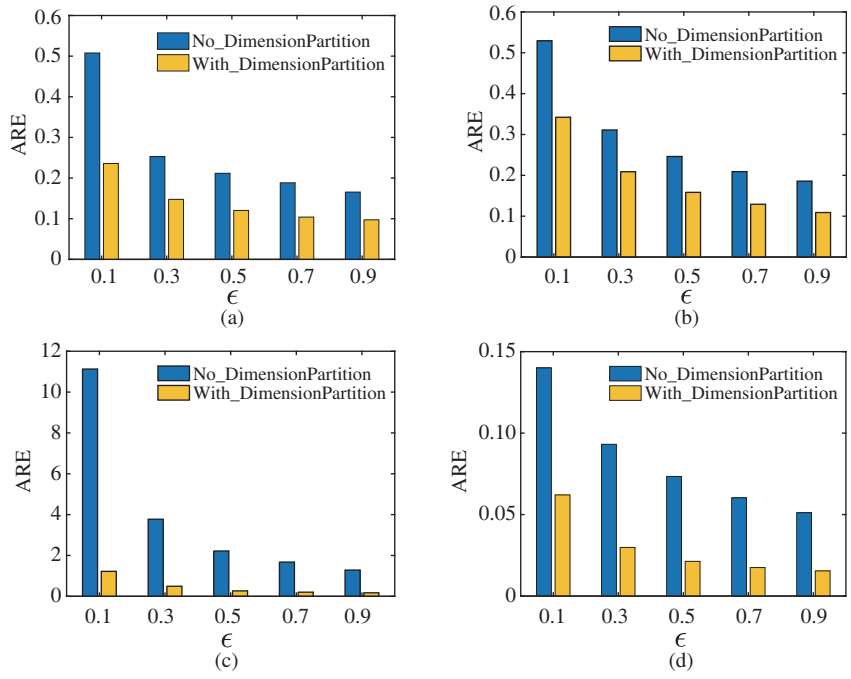


图 7 (网络版彩图) 基于多重哈希的维度划分机制的效用性评估 ($\omega = 100$)

Figure 7 (Color online) Utility evaluations of DimParti when ϵ changes ($\omega = 100$). (a) StateFlu; (b) Uber; (c) OnlineRetail; (d) Syn

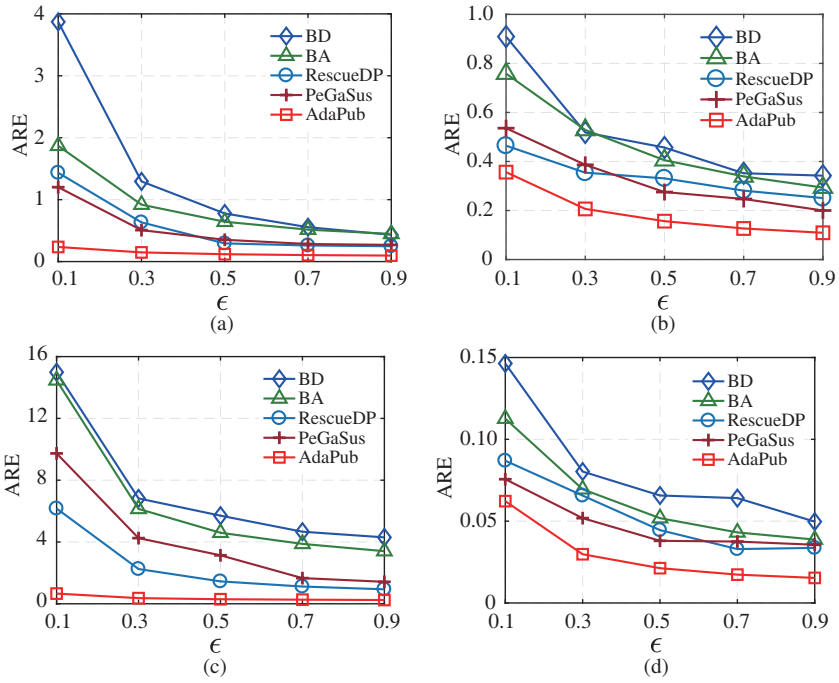


图 8 (网络版彩图) 隐私保护机制效用性随隐私预算 ϵ 的变化情况 ($\omega = 100$)

Figure 8 (Color online) Utility comparisons on stream datasets when ϵ changes ($\omega = 100$). (a) StateFlu; (b) Uber; (c) OnlineRetail; (d) Syn

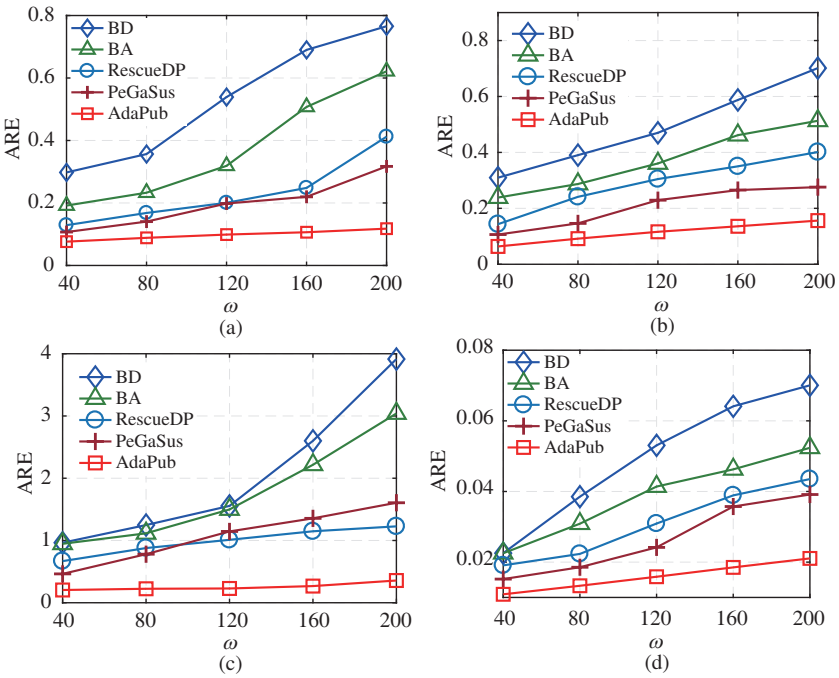


图 9 (网络版彩图) 隐私保护机制效用性随滑动窗口 ω 的变化情况 ($\epsilon = 1$)

Figure 9 (Color online) Utility comparisons on stream datasets when ω changes ($\epsilon = 1$). (a) StateFlu; (b) Uber; (c) OnlineRetail; (d) Syn

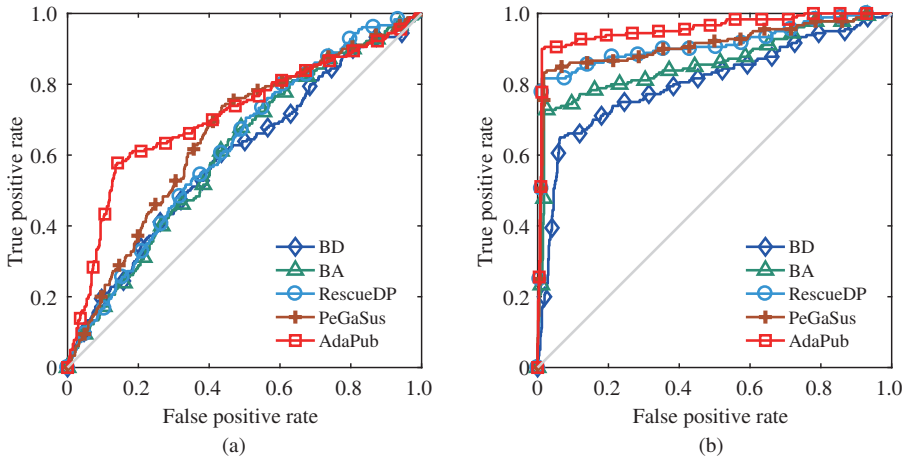


图 10 (网络版彩图) Flu 数据流上升/下降点检测的 ROC 曲线

Figure 10 (Color online) ROC curves for detecting jumping/dropping points on stream Flu ($L = 10, \theta = 50$). (a) $\epsilon = 0.01, \omega = 10$; (b) $\epsilon = 0.1, \omega = 10$

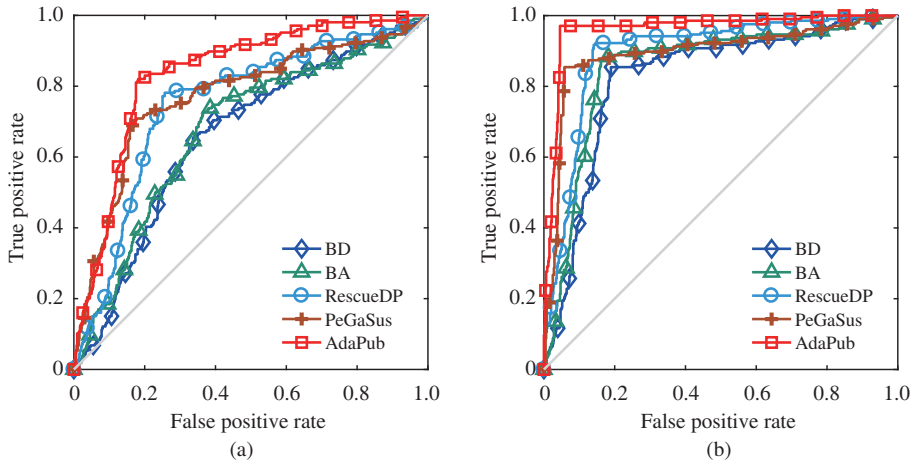


图 11 (网络版彩图) Flu 数据流低信号点检测的 ROC 曲线

Figure 11 (Color online) ROC curves for detecting low signal points on stream Flu ($L = 10, \theta = 300$). (a) $\epsilon = 0.01, \omega = 10$; (b) $\epsilon = 0.1, \omega = 10$

AdaPub 机制和其他机制的平均相对误差随隐私预算和滑动窗口大小的变化情况. 可以看到, AdaPub 机制由于具有数据自适应性, 能够始终维持最小的平均相对误差. 此外, 本文提出的 AdaPub 机制可以有效解决数据稀疏性的问题, 从而有效提高了发布数据流的效用性.

图 10 和 11 分别展示了针对 Flu 数据流上升/下降点 (jumping/dropping points) 检测以及低信号点 (low singal points) 检测的 ROC 曲线, 其中 L 表示检测时间长度, θ 表示检测阈值. 可以观察到, 在不同参数设置下, AdaPub 机制的性能基本都优于其他已有机制, 能够更好地应用到实际场景中.

图 12 展示了相关性分析结果, 相关系数越大表明相关性越强. 可以看到, AdaPub 机制始终优于其他机制, 说明基于 AdaPub 机制发布的数据流与原始数据流具有较强的相关性. 在 Uber 和 Syn 数据流上, 大部分机制都能够维持较高的相关性, 这是因为 Uber 和 Syn 数据流数值级别高且不稀疏. 在面对计数值很小且稀疏性强的数据流 OnlineRetail 时, AdaPub 机制依旧能够保证较好的相关性, 证

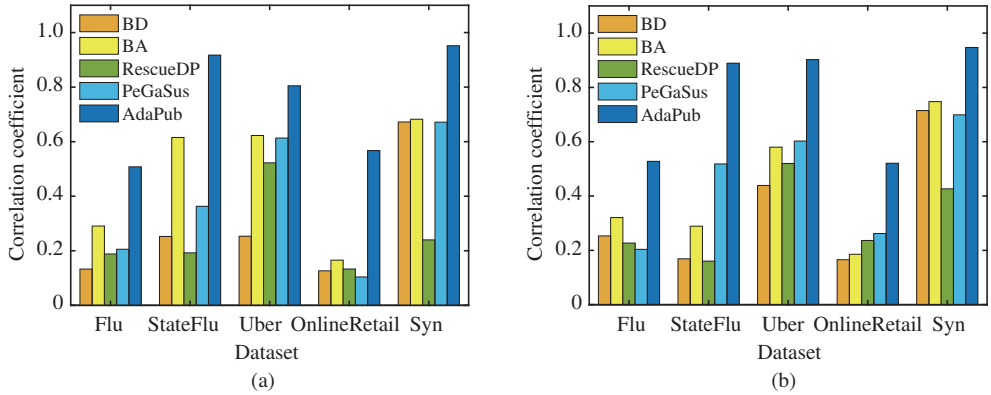


图 12 (网络版彩图) 原始数据流与隐私保护后数据流之间的相关性分析 ($\epsilon = 1, \omega = 100$)

Figure 12 (Color online) Correlation analysis between original stream and released stream ($\epsilon = 1, \omega = 100$). (a) Pearson correlation; (b) Spearman's rank correlation

表 3 具有二叉树结构的层次数据流

Table 3 The hierarchical aggregated streams datasets with binary-tree structure

Dataset	Dimension	# of Aggregation	Height	Weight	Sparsity
SF32	32	31	5	[1, 2, 4, 8, 16]	0.0036
OR1024	1024	1023	10	[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]	0.3715

表 4 具有四叉树结构的层次数据流

Table 4 The hierarchical aggregated streams datasets with quad-tree structure

Dataset	Dimension	# of Aggregation	Height	Weight	Sparsity
SF32	32	21	3	[1, 4, 16]	0.0053
OR1024	1024	341	5	[1, 4, 16, 64, 256]	0.2851

明 AdaPub 机制能够有效处理稀疏数据流.

6.3 层次数据流隐私保护发布机制效用性评估

本小节对 HierAdaPub 机制进行效用性评估. 将 StateFlu 的后 32 维和 OnlineRetail 的后 1024 维分别表示为 SF32 和 OR1024, 并构造成具有二叉树和四叉树结构的层次数据流, 如表 3 和 4 所示. 假设根节点为第一层. 在实验中, 本小节分别与 PeGaSus 机制、AdaPub_NoDimParti 机制和 AdaPub 机制进行对比. 在分配隐私预算时, HierAdaPub 机制采用最优隐私预算分配策略, 而其他 3 个机制都采用均匀隐私预算分配策略. 图 13 和 14 表明了 HierAdaPub 机制在处理具有二叉树和四叉树结构的层次数据流时始终都具有最小的平均相对误差. 图 13(a) 和 14(a) 的平均相对误差在相同隐私保护程度下整体上小于图 13(b) 和 14(b). 这是因为 OR1024 数据流的稀疏性远大于 SF32 数据流. 尽管如此, 本文所设计的 HierAdaPub 机制即使是处理稀疏数据流, 也能够保证较高的数据效用性. 这证明本文提出的 HierAdaPub 机制能够解决数据流稀疏性问题, 具有更好的实用性.

6.4 算法运行时间

表 5 和 6 分别展示了不同隐私保护机制的运行时间, 其中 d 和 $|A|$ 分别表示数据流的维度和层

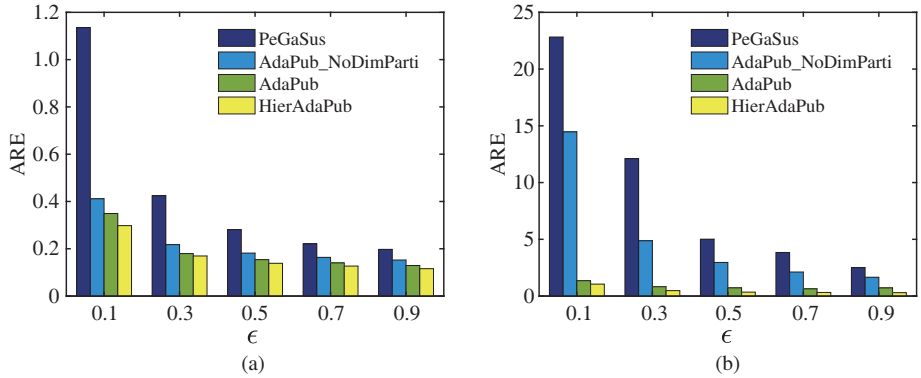


图 13 (网络版彩图) 具有二叉树结构的层次数据流隐私保护随 ϵ 的变化情况 ($\omega = 100$)

Figure 13 (Color online) ARE of hierarchical aggregated streams with binary-tree structure ($\omega = 100$). (a) SF32; (b) OR1024

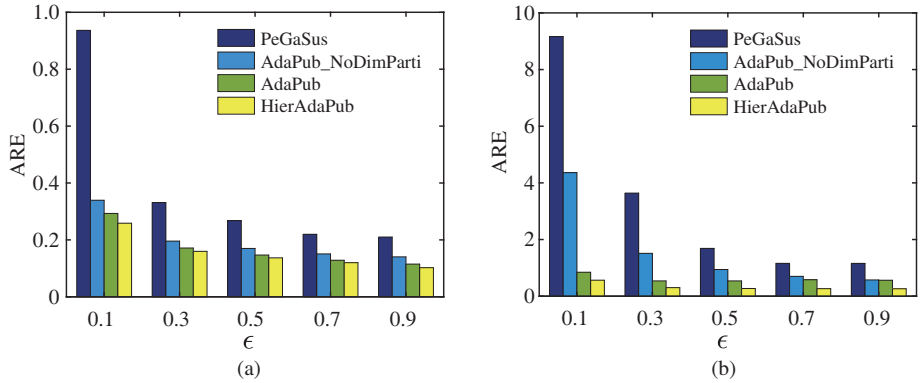


图 14 (网络版彩图) 具有四叉树结构的层次数据流隐私保护随 ϵ 的变化情况 ($\omega = 100$)

Figure 14 (Color online) ARE of hierarchical aggregated streams with quad-tree structure ($\omega = 100$). (a) SF32; (b) OR1024

表 5 针对数据流发布的隐私保护机制的运行时间 (s)

Table 5 Runtime (s) of privacy-preserving mechanisms on different datasets

	BD	BA	RescueDP	PeGaSus	AdaPub
Time complexity	$O(d)$	$O(d)$	$O(d^2)$	$O(d\|\mathcal{C}\|_m)$	$O(d\|\mathcal{C}\|_m + dg)$
Flu ($d = 1$)	1×10^{-5}	1×10^{-5}	1×10^{-3}	6×10^{-5}	6×10^{-5}
StateFlu ($d = 51$)	2×10^{-5}	2×10^{-5}	8×10^{-3}	6×10^{-3}	6×10^{-3}
Uber ($d = 32$)	1×10^{-5}	1×10^{-5}	4×10^{-4}	1×10^{-3}	1×10^{-3}
OnlineRetail ($d = 1298$)	2×10^{-4}	2×10^{-4}	3×10^{-2}	7×10^{-2}	7×10^{-2}
Syn ($d = 100$)	5×10^{-5}	5×10^{-5}	1×10^{-3}	9×10^{-3}	9×10^{-3}

次聚合数据流的节点数. 本文所设计的隐私保护机制 AdaPub 和 HierAdaPub 能够在明显提升发布数据流效用性的前提下维持可接受的运行时间. 基于 5.2 小节的分析与证明, AdaPub 机制的时间消耗主要用于 AdaCluster 算法和 DimParti 算法. 由于实验设置 $g = 20$, 因此 DimParti 的时间消耗实际较小, 主要时间消耗在 AdaCluster 算法上, 故实际时间复杂度为 $O(d|\mathcal{C}|_m + 20d) = O(d|\mathcal{C}|_m)$. 此外, 在

表 6 针对层次聚合数据流发布的隐私保护机制的运行时间 (s)

Table 6 Runtime (s) of privacy-preserving mechanisms on hierarchical streams with binary and quad-tree structures

Algorithm	Time complexity	SF32 (Binary) ($\|A\| = 31$)	OR1024 (Binary) ($\|A\| = 1023$)	SF32 (Quad) ($\|A\| = 21$)	OR1024 (Quad) ($\|A\| = 341$)
PeGaSus	$O(\|\mathcal{C}\|_m \, \|A\|)$	4×10^{-3}	7×10^{-2}	2×10^{-3}	6×10^{-5}
HierAdaPub	$O(H(\mu_m\|\mathcal{C}\|_m + \mu_m g))$	9×10^{-3}	6×10^{-2}	7×10^{-3}	4×10^{-2}

实际场景中, AdaCluster 算法可以很容易地被分布式执行, 因此可以进一步降低算法运行时间.

7 总结

本文提出了群智感知系统中面向多维感知数据流实时发布的数据自适应隐私保护机制, 能够自适应地学习和分析数据流的时空相关性, 不再依赖预定义的固定参数, 能够动态地学习数据流的变化特性, 并自适应地调整和更新隐私参数. 此外, 本文还提出了针对层次数据流实时发布的自适应隐私保护机制, 基于层次数据流的结构特性设计了最优隐私预算分配策略, 从而最小化噪声方差. 大量实验结果均验证了所设计隐私保护机制能够在提供隐私保证的前提下有效提高发布数据流的效用性. 本文的研究内容为群智感知系统中数据流的实时发布提供了有效的隐私保证, 对大数据时代促进群智感知系统的发展和普及具有重要意义.

参考文献

1 Guo B, Wang Z, Yu Z, et al. Mobile crowd sensing and computing. ACM Comput Surv, 2015, 48: 1–31

2 Zhang X, Hamm J, Reiter M K, et al. Statistical privacy for streaming traffic. In: Proceedings of the Network and Distributed System Security Symposium, San Diego, 2019. 1–15

3 Cao Y, Yoshikawa M. Differentially private real-time data publishing over infinite trajectory streams. IEICE Trans Inf Syst, 2016, 99: 163–175

4 Wang J Y, Liu C, Fu X C, et al. Crucial patterns mining with differential privacy over data streams. J Soft, 2019, 30: 158–176 [王金艳, 刘陈, 傅星珵, 等. 差分隐私的数据流关键模式挖掘方法. 软件学报, 2019, 30: 158–176]

5 Li M, Zhu H, Gao Z, et al. All your location are belong to us: breaking mobile social networks for automated user location tracking. In: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, Philadelphia, 2014. 43–52

6 Ji S, Li W, Srivatsa M, et al. General graph data de-anonymization: from mobility traces to social networks. ACM Trans Inf Syst Secur, 2016, 18: 1–29

7 Wu Y J, Ge C, Zhang L Q, et al. An algorithm for differential privacy streaming data publication based on matrix mechanism under exponential decay mode. Sci Sin Inform, 2017, 47: 1493–1509 [吴英杰, 葛晨, 张立群, 等. 指数衰减模式下基于矩阵机制的差分隐私流数据发布算法. 中国科学: 信息科学, 2017, 47: 1493–1509]

8 Cao Y, Yoshikawa M, Xiao Y, et al. Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans Knowl Data Eng, 2019, 31: 1281–1295

9 Cao Y, Yoshikawa M, Xiao Y, et al. Quantifying differential privacy under temporal correlations. In: Proceedings of IEEE International Conference on Data Engineering, San Diego, 2017. 821–832

10 Dwork C, Roth A. The algorithmic foundations of differential privacy. FNT Theor Comput Sci, 2014, 9: 211–407

11 Zhang X J, Meng X F. Differential privacy in data publication and analysis. J Comput, 2014, 4: 197–219 [张啸剑, 孟小峰. 面向数据发布和分析的差分隐私保护. 计算机学报, 2014, 4: 197–219]

12 Dwork C, Naor M, Pitassi T, et al. Differential privacy under continual observation. In: Proceedings of ACM Symposium on Theory of Computing, Cambridge, 2010. 715–724

13 Dwork C. Differential privacy in new settings. In: Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA19), Austin, 2010. 174–183
14 Kellaris G, Papadopoulos S, Xiao X, et al. Differentially private event sequences over infinite streams. Proc VLDB Endow, 2014, 7: 1155–1166
15 Chen R, Shen Y, Jin H. Private analysis of infinite data streams via retroactive grouping. In: Proceedings of ACM International Conference on Information and Knowledge Management, Melbourne, 2015. 1061–1070
16 Fan L Y, Xiong L. An adaptive approach to real-time aggregate monitoring with differential privacy. IEEE Trans Knowl Data Eng, 2014, 26: 2094–2106
17 Wang Q, Zhang Y, Lu X, et al. RescueDP: real-time spatio-temporal crowd-sourced data publishing with differential privacy. In: Proceedings of International Conference on Computer Communications, San Francisco, 2016. 1–9
18 Chen Y, Machanavajjhala A, Hay M, et al. PeGaSus: data-adaptive differentially private stream processing. In: Proceedings of ACM Conference on Computer and Communications Security, Dallas, 2017. 1375–1388
19 Papadimitriou S, Li F, Kollios G, et al. Time series compressibility and privacy. In: Proceedings of Very Large Data Bases, Vienna, 2007. 459–470
20 Wang H, Xu Z. CTS-DP: publishing correlated time-series data via differential privacy. Knowledge-Based Syst, 2017, 122: 167–179
21 Wang T, Yang X Y, Ren X B, et al. Adaptive differentially private data stream publishing in spatio-temporal monitoring of IoT. In: Proceedings of IEEE International Performance Computing and Communications Conference, London, 2019. 1–8
22 Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis. In: Proceedings of Theory of Cryptography Conference, New York, 2006. 265–284
23 Kifer D, Lin B R. Towards an axiomatization of statistical privacy and utility. In: Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Indianapolis, 2010. 147–158

Data-adaptive privacy-preserving mechanism for data stream publishing in real-time

Teng WANG[1], Xinyu YANG[1], Xuebin REN[1*] & Jun ZHAO[2]

1. *School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an* 710049*, China*;
2. *School of Computer Science and Engineering, Nanyang Technological University, Singapore* 639798*, Singapore*
* Corresponding author. E-mail: xuebinren@mail.xjtu.edu.cn

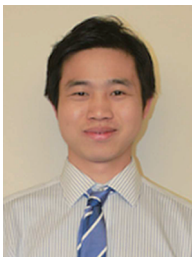
Abstract The real-time publishing and deep exploitation of data streams in crowdsensing systems have significantly facilitated people's daily lives. However, it also seriously compromises the private information of participating users. The existing approaches are non-adaptive to dynamic changes of streams, thus are vulnerable to low data utility. To address such concerns, in this paper, we present AdaPub, a data-adaptive mechanism for infinite multi-dimensional stream real-time publishing under ω-event differential privacy. No longer predefining parameters, AdaPub seamlessly incorporates two modules DimParti and AdaCluster to learn spatial and temporal correlations simultaneously in a data-adaptive manner, thus ensuring data adaptability of privacy-preserving mechanism and greatly improving the data utility of the sanitized streams. Moreover, for hierarchical aggregated streams publishing, we further propose a data-adaptive mechanism HierAdaPub that leverages an optimal privacy budget allocation strategy to minimize the total perturbation errors. Extensive experiments on real-world and synthetic datasets demonstrate that our mechanisms substantially outperform the state-of-the-art solutions in terms of both data utility and data adaptivity while achieving strong privacy guarantees.

Keywords data stream publishing, data adaptiveness, differential privacy, spatio-temporal correlations, data utility

Teng WANG was born in 1995. She received her B.S. degree from School of Software in XiDian University, China, in 2015. She is currently pursuing her Ph.D. degree in School of Computer Science and Technology, Xi'an Jiaotong University. She was a visiting student at School of Computer Science and Engineering in Nanyang Technological University (NTU) in Singapore from 2018 to 2019. Her research interests include mobile crowdsensing systems (MCS) and data privacy-preserving and analysis with applications to the Internet of Things.

Xuebin REN was born in 1989. He received his Ph.D. degree from Department of Computer Science and Technology from Xi'an Jiaotong University (XJTU), China, in 2017. He is currently a lecturer at School of Computer Science and Technology in XJTU. He was a visiting Ph.D. student at Department of Computing in Imperial College London from 2016 to 2017. He is currently looking at the security and privacy issues of big data analysis and machine learning in distributed and edge computing systems such as the Internet of Things and cyber-physical systems.

Xinyu YANG was born in 1973. He received his diploma in computer science and technology from Xi'an Jiaotong University (XJTU), China, in 2001 and his B.S., M.S., and Ph.D. degrees from XJTU in 1995, 1997, and 2001, respectively. He is currently a professor at School of Computer Science and Technology in XJTU. His research interests include wireless communication, mobile ad hoc networks, and network security and privacy.



Jun ZHAO was born in 1989. He received his Ph.D. degree in electrical and computer engineering from Carnegie Mellon University (CMU) in the US and his B.S. degree from Shanghai Jiaotong University in China. He is currently an assistant professor in School of Computer Science and Engineering in Nanyang Technological University (NTU) in Singapore. His research interests include blockchains, security, and privacy in the Internet of Things and deep learning.