软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(6):2348−2363 [doi: 10.13328/j.cnki.jos.006363]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于混洗差分隐私的直方图发布方法*

张啸剑, 徐雅鑫, 夏庆荣

(河南财经政法大学 计算机与信息工程学院, 河南 郑州 450046)

通信作者: 张啸剑, E-mail: xjzhang82@126.com

摘 要: 基于中心化/本地化差分隐私的直方图发布已得到了研究者的广泛关注. 用户的隐私需求与收集者的分析精度之间的矛盾直接制约着直方图发布的可用性. 针对现有直方图发布方法难以有效同时兼顾用户隐私与收集者分析精度的不足, 提出了一种基于混洗差分隐私的直方图发布算法 HP-SDP (histogram publication with shuffled differential privacy). 该算法结合本地哈希编码技术所设计的混洗应答机制 SRR (shuffled randomized response), 能够以线性分解的方式扰动用户数据以及摆脱数据值域大小的影响. 结合 SRR 机制产生的用户消息, 设计了一种基于堆排列技术的用户消息均匀随机排列算法 MRS (message random shuffling), 混洗方利用 MRS 对所有用户的消息进行随机排列. 由于经过 MRS 混洗后的消息满足中心化差分隐私, 使得恶意收集者无法通过消息与用户之间的链接对目标用户进行身份甄别. 此外, HP-SDP 利用基于二次规划技术的后置处理算法 POP (post-processing)对混洗后的直方图进行求精处理. HP-SDP 算法与现有的 7 种直方图发布算法在 4 种数据集上所做实验结果表明, 其发布精度优于同类算法.

关键词: 中心化差分隐私; 本地化差分隐私; 混洗差分隐私; 直方图发布; 消息混洗; 后置处理

中图法分类号: TP306

中文引用格式: 张啸剑, 徐雅鑫, 夏庆荣. 基于混洗差分隐私的直方图发布方法. 软件学报, 2022, 33(6): 2348–2363. http://www.jos.org.cn/1000-9825/6363.htm

英文引用格式: Zhang XJ, Xu YX, Xia QR. Histogram Publication under Shuffled Differential Privacy. Ruan Jian Xue Bao/Journal of Software, 2022, 33(6): 2348–2363 (in Chinese). http://www.jos.org.cn/1000-9825/6363.htm

Histogram Publication under Shuffled Differential Privacy

ZHANG Xiao-Jian, XU Ya-Xin, XIA Qing-Rong

(College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450046, China)

Abstract: Given a distributed set D of categorical data defined on a domain $\mathcal{D}$, this work studies differentially private algorithms for releasing a histogram to approximate the categorical data distribution in D. Existing solutions for this problem mostly use central/local differential privacy models, which are two extreme assumptions of differential privacy. The two models, however, cannot balance the contradiction between the privacy requirement of users and the analysis accuracy of collectors. To remedy the deficiency caused by the current solutions under central/local differential privacy, this study proposes a differentially private method in a shuffling way, called HP-SDP, to release histogram. HP-SDP firstly employs the local hash technology to design the shuffled randomized response mechanism. Based on this mechanism, each user perturbs her/his data in a linear decomposition way of perturbation function, without worrying about the domain size, and reports the perturbed messages to the shuffler. And then, the shuffler in HP-SDP permutes the reported messages by using a uniformly random permutation method, which makes sure the shuffled messages satisfy central differential privacy, and the collector cannot reidentify a target user. Furthermore, HP-SDP adopts the convex programming technology to boost the accuracy of the released histogram. Theoretical analysis and experimental evaluations show that the proposed methods can effectively improve the utility of the histogram, and outperform the existing solutions.

* 基金项目: 国家自然科学基金(61502146, 91646203, 91746115, 62072156); 河南省自然科学基金(162300410006); 河南省科技攻关项目(202102310563); 河南财经政法大学青年拔尖人才资助计划

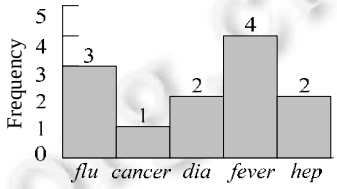
收稿时间: 2020-11-10; 修改时间: 2021-01-28; 采用时间: 2021-03-09; jos 在线出版时间: 2021-11-24

Key words: central differential privacy; local differential privacy; shuffled differential privacy; histogram publication; message shuffling; post-processing

移动互联网环境下新兴技术的快速发展与应用, 使得类别数据(categorical data)的获取与收集变得尤为容易. 类别数据的收集与分析能够有效提升产品服务与设备服务的质量, 以及向用户提供个性化体验. 直方图是类别数据分析的常用技术, 该技术使用分箱结构近似描述类别数据的分布信息, 按照类别属性划分成不相交的桶, 每个桶由频率或者计数表示其特征. 例如, 图 1(a)描述了用户所拥有的类别属性(disease), 图 1(b)是基于该类别属性产生的直方图. 然而, 类别数据通常蕴含着个人的敏感信息, 在提供给收集者或者不可信第三方时, 个人隐私有可能被泄露. 例如, 图 1(b)中疾病 dia 的频率是 2, 不可信收集者获得 dia 的频率并且操纵 Ann 后, 即可对图 1(a)中的 Bob 进行链接攻击与操纵攻击[1].

Name	Disease
Tom	*flu*
Bob	*dia*
Ann	*dia*
Lucy	*flu*
…	…

(a) User data



(b) Disease

图 1 用户数据及其对应的直方图分布

本地化差分隐私技术(local differential privacy, LDP)下的直方图发布有效解决了类别数据收集过程中个人隐私泄露问题, 该技术允许每个用户扰动自身数据之后再响应收集者的需求. 然而, LDP 下直方图发布方法存在一些不足: (1) LDP 下的直方图发布结果达不到中心化差分隐私(central differential privacy, CDP)下的发布精度, 例如, LDP 下聚集求和(SUM)误差为 $\Theta(\sqrt{n})$, 而 CDP 下的聚集求和误差仅为 $\Theta(1)$; (2) 恶意收集者可用通过扰动后的值(或者消息)与用户之间的链接关系, 对目标用户的身份进行甄别[2]; (3) LDP 的隐私预算设置比 CDP 设置的值大, 弱化了差分隐私保护程度. 与 LDP 相比, 尽管 CDP 下的直方图发布精度高, 然而由于假设中心化收集者是可信的, 并且能够看到所有用户的原始类别数据, 则 CDP 不及 LDP 安全. 混洗差分隐私(shuffled differential privacy, SDP)是处于 CDP 与 LDP 之间的一种模型, 通过混洗操作打乱了用户身份与消息之间的对应关系, 能够为用户提供类似于 LDP 模型保护的力度, 也能为收集者提供类似于 CDP 模型提供的查询与分析精度.

目前, 基于 SDP 的直方图发布方法包括 SH[3]、AUE[4]、MURS[5]以及 mixDUMP[6]算法. SH 算法结合线性组合技术对本地扰动方法 GRR[7]的输出概率进行线性分解, 利用均匀噪音掩盖真实数据. 该算法实现了隐私增强并且提升了发布精度. 然而, SH 算法由于采用 GRR 扰动方法, 其发布精度易受较大值域的影响, 大值域会造成 SH 的发布精度急剧下降. 不同于 SH 算法, AUE 算法的发布精度不受值域大小的影响. 然而, 该算法由于利用了 One-Hot 编码机制, 进而会导致通信代价与值域大小呈线性关系. 为了克服 SH 与 AUE 算法存在的不足, MURS 算法利用对称本地哈希技术把大值域哈希到较小的地址空间中, 实现了发布精度的提升与通信代价的减小. 类似于 MURS 算法, mixDUMP 算法结合多消息混洗模式实现了直方图的发布. 然而, 该算法的核心扰动方法依然是 GRR 方法, 发布精度依然受到类别数据值域大小的影响. 此外, SH、AUE、MURS 与 mixDUMP 这 4 种算法均没有详细介绍如何实现消息的混洗操作.

结合文献[1–4]可知, 利用 SDP 发布直方图依然存在诸多挑战: (1) 本地用户在设计混洗随机应答机制时, 如何避免大值域对发布精度与通信代价的影响; (2) 混洗方如何设计高效的随机排列方法来实现用户消息的混洗操作; (3) 收集者如何设计有效的后置处理技术来提升最终的发布精度. 总而言之, 目前还没有一个行之有效且满足混洗差分隐私的直方图发布算法能够克服上述方法存在的问题. 为此, 本文基于混洗差分隐私技术提出了一种直方图发布算法——HP-SDP 能够兼顾上述的问题需求.

本文的主要贡献如下:

1) 为了解决挑战 1, HP-SDP 算法提出了一种混洗随机应答机制 SRR (shuffled randomized response), 该机制利用优化本地哈希机制的线性分解方式摆脱了值域大小的影响;
2) 为了解决挑战 2, HP-SDP 算法提出了一种基于堆排列方法的用户消息均匀随机排列算法 MRS (message random shuffling). 混洗方利用 MRS 对所有用户的消息进行随机排列, 进而使得恶意收集者无法通过消息与用户之间的链接对目标用户进行身份甄别;
3) 为了解决挑战 3, HP-SDP 算法提出了一种基于二次规划技术的后置处理算法 POP (post-process), 收集者利用 POP 算法对混洗方发送过来的消息进行求精处理;
4) 理论分析了 HP-SDP 算法满足(ε,δ)-中心化差分隐私, 以及每个桶频率的无偏性、误差边界以及最大偏差. 通过合成数据与真实数据实验分析, HP-SDP 算法具有较高的可用性.

1 相关工作

基于差分隐私的直方图发布得到较为广泛的研究. CDP 下直方图发布通过收集所有用户的原始数据来发布噪音直方图. LAP[8]、Boost[9]及 NoiseFirst[10]是 CDP 下直方图发布的典型代表. LAP 算法直接在直方图的每个桶中添加拉普拉斯噪音来保护相应的计数值, 该算法产生的发布误差为 $O(\sqrt{n}/\varepsilon)$. Boost 算法采用层次树的节点记录桶中的计数, 再结合层次树高度与拉普拉斯机制完成直方图发布, 该算法的发布误差为 $O(\sqrt{\log 3n}/\varepsilon)$. 不同于 Boost 算法, NoiseFirst 算法首先利用拉普拉斯机制对每个桶添加噪音, 再利用基于动态规划的分组策略对所有噪音桶计数进行聚类分组, 然后发布重构之后的直方图, 该算法的发布误差为 $O(\sqrt{n-d}/\varepsilon)$, 其中, d 为类别属性的值域大小. LDP 下的直方图发布算法通过基于用户本地扰动的数据构建直方图. Rappor[11]、SHist[12]与 TreeHist[13]是 LDP 下的直方图发布代表算法. Rappor 算法结合一元编码与布隆过滤技术, 把类别属性的整个值域哈希到较小的值域中, 结合哈希值域统计每个类别属性的频率, 其发布误差为 $O(d/\varepsilon\sqrt{n})$. SHist 方法采用随机矩阵投影技术对类别属性的值域进行编码, 随机扰动 1 个比特位发送给收集者, 发布误差为 $O(\sqrt{\log d}/\varepsilon\sqrt{n})$. 相比于 SHist 算法, 为了减少计算代价, TreeHist 算法借助于计数概要与 Hadamard 转换技术构建满足 LDP 的前缀树, 遍历前缀树中的各个节点即可获得相应的直方图, 该算法的发布误差为 $O(\sqrt{\log(n)\log d}/\varepsilon\sqrt{n})$.

根据上述分析可知, CDP 与 LDP 下的直方图发布算法各自存在不同的优缺点. CDP 下的直方图发布误差低, 然而收集者泄露用户隐私的风险高. LDP 下的用户隐私泄露风险低, 然而直方图发布误差高. SDP 模型的出现有效地平衡了中心化与本地化差分隐私的缺点. ESA[14]是首个混洗技术与差分隐私技术融合的框架, 用户本地扰动的消息以匿名通道的方式传送给混洗方, 进而使得收集者无法重甄别目标用户的身份. 然而, 该框架没有给出混洗差分隐私的形式化定义以及误差边界. DDPS[15]算法结合单消息混洗框架给出了混洗差分隐私的形式化定义, 并实现了二进制数据的非交互式聚集查询, 该算法将 GRR 算法分解成伯努利分布与均匀分布来本地扰动二进制数据. 然而, 该算法的通信代价高且可用性比较低, 相应的查询误差为 $O(\sqrt{\log d\log 1/\delta}/\varepsilon n)$. 不同于 DDPS 算法, DPSMS[16]算法利用多消息扰动-混洗模型与 IKOS[17]安全协议实现了实数聚集查询, 并取得了近似于拉普拉斯机制所产生的误差 $O(1/\varepsilon)$. 然而, 该算法由于使用了安全协议导致了很高的通信代价. 类似于 DPSMS 算法, CESS[18]算法采用多消息并行化扰动-混洗框架实现了[0,1]区间上的求和查询, 该算法利用多个混洗协议与 IKOS 协议实现了分布式拉普拉斯机制产生的误差效果, 其查询误差为 $O(1/\varepsilon)$. 上述算法通常关注于实数域内的聚集查询, 而涉及直方图发布的研究较少. 目前, SH、AUE、MURS 以及 mixDUMP 是 SDP 下直方图发布的典型的代表算法. SH 算法利用单消息混洗框架实现了实数型数值上的直方图发布, 尽管该算法的发布精度与通信代价优于 DDPS 算法, 但其自身的发布精度受值域大小的影响, 值域越大, 发布误差越高. AUE 算法结合 One-Hot 编码机制实现了混洗差分隐私下的直方图发布, 然而该算法的通信代价与类别属性的值域线性相关, 并且不满足本地化差分隐私. MURS 与 mixDUMP 算法分布利用对称本地哈希编码机制与 GRR 算法实现了直方图发布. 尽管这两种算法借助于多消息提升了发布精度, 然而它们

与 SH、AUE 类似, 没有给出具体的混洗排列方法以及后置处理方法. 基于上述分析, 本文提出一种基于单消息单混洗方框架且满足混洗差分隐私的直方图发布算法 HP-SDP, 该算法利用快速随机排列与后置处理技术提升最终的发布精度.

2 基础知识与问题描述

2.1 混洗差分隐私

不同于 CDP 与 LDP 隐私保护技术, SDP 利用拼接与排列技术处理所有用户本地扰动之后的消息向量, 确保混洗操作满足(ε,δ)-CDP, 并完成 LDP 向 CDP 的过渡. 3 种差分隐私保护模型的形式化定义如下所示.

设 D ($D\in\mathcal{D}$)为分布式类别数据集, $D=\{v_1,v_2,\ldots,v_n\}$由 n 个类别数据构成且$\forall v_i\in\mathcal{D}$, $d=|\mathcal{D}|$. n 个类别数据分布在 n 个用户手中.

定义 1((ε,δ)-中心化差分隐私). 设 D 与 D'彼此相差一条类别数据且互为近邻关系. 给定一个直方图发布协议$\mathcal{M}$, $\mathcal{Y}$为$\mathcal{M}$的输出域, 若$\mathcal{M}$在 D 与 D'上任意输出结果的概率满足下列不等式, 则$\mathcal{M}$满足(ε,δ)-中心化差分隐私:

$$\Pr[\mathcal{M}(D)\in\mathcal{Y}]\leqslant e^{\varepsilon}\times\Pr[\mathcal{M}(D')\in\mathcal{Y}]+\delta \tag{1}$$

其中, ε表示隐私预算, δ为($\delta\in(0,1]$)隐私泄露风险概率, e 表示自然对数的底数.

定义 2((ε,δ)-本地化差分隐私). 设 v_i 与 v_i' 为值域$\mathcal{D}$上任意两条不同类别数据. 给定一个直方图发布协议$\mathcal{M}=(\mathcal{R},\varepsilon)$, $\mathcal{Y}$为$\mathcal{M}$的输出域, 若$\mathcal{M}$在 v_i 与 v_i' 上任意输出结果的概率满足下列不等式, 则$\mathcal{M}$满足(ε,δ)-本地化差分隐私:

$$\Pr[\mathcal{M}(v)\in\mathcal{Y}]\leqslant e^{\varepsilon}\times\Pr[\mathcal{M}(v')\in\mathcal{Y}]+\delta \tag{2}$$

其中, $\mathcal{R}$表示本地随机应答机制, ε表示隐私预算, δ为($\delta\in(0,1]$)隐私泄露风险概率, e 表示自然对数的底数.

定义 3((ε,δ)-混洗差分隐私). 设$\mathcal{M}=(\mathcal{R},\mathcal{S},\mathcal{A})$. 每个用户 u_i 利用$\mathcal{R}$: $\mathcal{D}\to\mathcal{Y}$扰动 v_i: $y_i=\mathcal{R}(v_i)$. 令 $M=\{y_1,y_2,\ldots,y_n\}$, $\mathcal{S}(M)$为混洗之后的输出结果, 其值域为$\mathcal{Y}'$. 如果$\mathcal{S}(M)$: $\mathcal{Y}\to\mathcal{Y}'$满足($\varepsilon,\delta$)-中心化差分隐私, 则$\mathcal{M}$满足($\varepsilon,\delta$)-混洗差分隐私:

$$\Pr[\mathcal{S}(M)\in\mathcal{Y}']\leqslant e^{\varepsilon}\times\Pr[\mathcal{S}(M')\in\mathcal{Y}']+\delta \tag{3}$$

其中, $\mathcal{R}$表示本地混洗随机应答机制, $\mathcal{S}$表示混洗排列机制, $\mathcal{A}$表示收集端的直方图发布需求, ε表示隐私预算, δ为($\delta\in(0,1]$)隐私泄露风险概率, e 表示自然对数的底数.

2.2 随机应答机制

由定义 1–定义 3 可知: 若要实现混洗差分隐私, 需要设计合理的本地混洗随机应答协议$\mathcal{R}$以及混洗协议$\mathcal{S}$. 目前, 随机应答机制[19]是实现本地扰动的常用技术. 在用户发送数据 v_i 之前, 对其进行随机扰动. 该机制的原始思想是用户在响应敏感布尔问题查询时, 以概率 p 真实应答, 以 q 的概率给出相反的应答. 目前, 基于随机应答机制出现了以 GRR 与 OLH[20]为代表的本地扰动方法.

- GRR 方法.

给定 v_i 与 v_j, 且 $v_i, v_j\in\{1,2,\ldots,d\}$, GRR 方法如公式(4)所示:

$$\Pr[GRR(v_i)=v_j]=\begin{cases}p=\dfrac{e^{\varepsilon}}{e^{\varepsilon}+d-1}, & v_j=v_i\\ q=\dfrac{1-p}{d-1}, & v_j\neq v_i\end{cases} \tag{4}$$

其中, v_i 表示用户拥有的数据; v_j 表示值域$\{1,2,\ldots,d\}$中的其他数据; d 表示类别值域的大小; e 表示自然对数的底数.

- OLH 方法.

给定 v_i 且 $v_i\in\{1,2,\ldots,d\}$, OLH 利用哈希簇$\mathcal{H}$把 v_i 编码成$\{1,2,\ldots,g\}$ ($g<<d$)中某个哈希值, 即 $x=H(v_i)$, 且

$x\in\{1,2,\ldots,g\}$, $H\in\mathcal{H}$. OLH 本地扰动思想如公式(5)所示:

$$\Pr[OLH(x)=y]=\begin{cases}p=\dfrac{\mathrm{e}^{\varepsilon}}{\mathrm{e}^{\varepsilon}+g-1}, & x=y\\ q=\dfrac{1}{\mathrm{e}^{\varepsilon}+g-1}, & x\neq y\end{cases} \tag{5}$$

其中, v_i 表示用户拥有的数据, x 表示 v_i 经过哈希函数 H 哈希之后的值, y 表示哈希值域$\{1,2,\ldots,g\}$中的任意值, g 表示哈希函数值域的大小, d 表示类别值域的大小, e 表示自然对数的底数.

2.3 问题描述

给定 n 个用户、单个可信的混洗方和一个收集者. 每个用户拥有类别数据 v_i, 且$\forall v_i\in\mathcal{D}$, v_i 通过$\mathcal{R}$本地扰动后发送消息给混洗方$\mathcal{S}$. 混洗方混洗所有消息, 即$\mathcal{S}(\mathcal{R}(v_1),\mathcal{R}(v_2),\ldots,\mathcal{R}(v_n))$. 收集者基于混洗后的消息向量构建直方图, 即$\mathcal{A}(\mathcal{S}(\mathcal{R}(v_1),\mathcal{R}(v_2),\ldots,\mathcal{R}(v_n)))$. 本文要解决的问题是: 在发布直方图的过程中, 如何使得$\mathcal{S}(\mathcal{R}(v_1),\ldots,\mathcal{R}(v_n))$满足$(\varepsilon,\delta)$-CDP, 且发布结果的均方差较小. 均方差 MSE (mean squared error)的表达式为

$$MSE(F,\tilde{F})=\frac{1}{d}\sum_{v\in\mathcal{D}}\mathrm{E}(f_v-\tilde{f}_v)^2 \tag{6}$$

其中, F 与 $\tilde{F}$ 分别表示原始直方图与估计直方图, f_v 与 $\tilde{f}_v$ 分别表示类别数据(或者桶)v 的真实频率与估计频率, d 表示类别值域的大小.

3 直方图发布算法 HP-SDP

3.1 直方图发布的原则

基于相关工作的分析可知, 在设计新的基于混洗差分隐私直方图发布算法时, 需要考虑 3 个原则.

1) 针对现有算法无法有效地处理类别属性值域过大的问题, 所设计的算法应尽可能地利用压缩技术(例如哈希转换)将原始值域转换到较小的值域空间;
2) 针对现有算法没有顾及如何具体实现用户消息的混洗排列操作, 所设计的算法应尽可能地体现该操作的高效性;
3) 针对现有算法没有考虑收集者如何设计有效的后置处理技术问题, 所设计的算法应尽可能地体现后置处理技术在提高直方图发布精度方面的作用.

针对原则 1–原则 3, 本文基于文献[5]提出了结合优化本地哈希技术的单消息混洗随机应答机制, 并实现了用户消息的随机混洗. 在此基础上, 提出了一种有效的直方图发布框架 HP-SDP, 如图 2 所示. 其中, $E(\cdot)$表示用户的本地编码机制、$\mathcal{R}$表示混洗随机应答机制、$\mathcal{S}$表示混洗排列机制、$\mathcal{A}$表示收集者的直方图发布需求. 该框架与 CDP 与 LDP 下的直方图发布存在本质上的区别: (1) 该框架中用户数据依然在用户端, 与(ε_c,δ)-CDP 中收集者完全掌控用户的原始数据不同; (2) 该框架满足(ε_c,δ)-CDP, 且发布精度高于(ε,δ)-LDP; (3) 从定义 1–定义 3 可知, 该框架能够实现隐私增强, 即ε越小, 隐私保护程度越高.

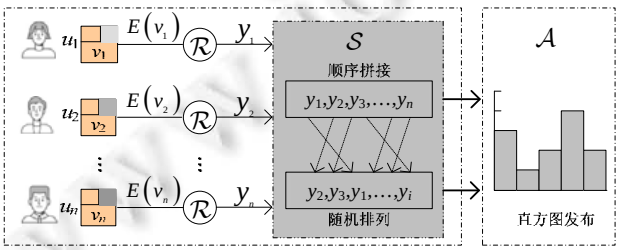


图 2 直方图发布框架 HP-SDP

3.2 HP-SDP算法

本节介绍 HP-SDP 算法, 该算法包括混洗随机扰动、随机排列以及直方图发布等操作, 具体实现细节如算法 1 所示.

算法 1. HP-SDP.

输入: n 个用户, 每个用户的数据 v, 隐私预算ε, 哈希函数的值域$\mathcal{G}$, $|\mathcal{G}|=g$, γ;

输出: 直方图 $\bar{F}$.

1. $M\leftarrow\varnothing$, $\tilde{F}\leftarrow\varnothing$;

用户端: $\mathcal{R}$

2. **for** users i=1 to n **do**
3. User i computes $\langle H_i,y_i\rangle=SRR(v_i,\varepsilon)$; //第 i 个用户利用 SRR 算法本地扰动 v_i, 产生$\langle H_i,y_i\rangle$
4. User i sends $\langle H_i,y_i\rangle$ to the shuffler; //第 i 个用户将扰动结果$\langle H_i,y_i\rangle$发送给混洗方
5. **end for**

混洗方: $\mathcal{S}$

6. Shuffler concatenates each pair $\langle H_i,y_i\rangle$: $M\leftarrow M\cup\langle H_i,y_i\rangle$; //混洗方拼接所有用户发来的消息
7. Shuffler randomly permutates σ: $\sigma=MRS(M)$; //混洗方利用 MRS 算法均匀随机混洗所有的消息
8. Shuffler sends σ to the collector; //混洗方将混洗后的消息向量发送给收集者

收集方: $\mathcal{A}$

9. **for each** $\langle H_i,y_i\rangle\in\sigma$ **do**
10. $\tilde{f}_v=\dfrac{1}{n}\cdot\dfrac{\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}-n\gamma\dfrac{g-1}{g}}{1-2\gamma\dfrac{g-1}{g}}$; //收集方估计每个类别数据 v 所对应的频率
11. $\tilde{F}\leftarrow\tilde{F}\cup\tilde{f}_v$; //估计类别数据值域中每个值的频率
12. **end for**
13. $\bar{F}=POP(\tilde{F})$; //对估计后的直方图进行后置处理
14. **return** $\bar{F}$.

HP-SDP 算法基于 SDP 来解决直方图的发布问题. 首先, 每个用户利用混洗随机应答机制 SRR 本地处理自己的数据并产生相应的消息, 然后再将消息发送给混洗方(步骤 2–步骤 5). 混洗方把所有的用户消息拼接成为 M(步骤 6), 再对 M 中的消息进行随机混洗排列, 然后将混洗结果发送给收集者(步骤 7、步骤 8). 收集者结合混洗后的消息向量σ构建相应的直方图并对其进行后置处理(步骤 9–步骤 13). 由算法 1 可知, 混洗随机应答机制 SRR、混洗随机排列 MRS 以及后置处理 POP 是 HP-SDP 算法的核心步骤. 下面, 首先阐述 SRR 算法的具体实现细节.

为了有效解决类别属性值域过大带来的影响, 本文基于文献[5], 采用哈希技术对原始值域$\mathcal{D}$进行本地哈希变换, 使其映射到一个较小的值域空间$\mathcal{G}(\mathcal{H}:\mathcal{D}\to\mathcal{G})$中, 设 $g=|\mathcal{G}|$. 然后基于$\mathcal{G}$, 利用公式(5)对哈希值进行本地扰动. 基于此思路, 提出了一种混洗随机应答机制 SRR. 不同于 LDP 下的本地扰动方法, SRR 通过对 GRR 方法的输出概率线性分解成为两种概率的组合, 以部分噪音值融合部分真实值来输出最终的扰动结果. SRR 算法输出某个消息的概率如公式(7)所示:

$$\forall_{y\in\mathcal{G}}\Pr[SRR(H(v))=y]=(1-\gamma)\boldsymbol{I}_{\{H(v)=y\}}+\gamma\Pr[Uniform(\mathcal{G})=y] \tag{7}$$

其中, $\boldsymbol{I}_{\{H(v)=y\}}$为标识函数, 若 $H(v)=y$, 则 $\boldsymbol{I}_{\{H(v)=y\}}=1$, 并且 $H(v)$以$(1-\gamma)$的概率扰动成 y.

由文献[3]可知, $\gamma=\sum\min\Pr[SRR(H(v))=y]=g/g+\mathrm{e}^{\varepsilon}-1$, $\Pr[Uniform(\mathcal{G})=y]=1/g$.

SRR 算法的具体实现细节如算法 2 所示.

算法 2. SRR.

输入: 用户的数据 v 且 $v\in\mathcal{D}$, 隐私预算ε, 哈希函数的值域$\mathcal{G}$, $|\mathcal{G}|=g$, γ;

输出: M.

1. $M\leftarrow\varnothing$;
2. Uniformly pick a hash function H from $\mathcal{H}$, and obtain $\langle H,H(v)\rangle$ ($H\in\mathcal{H}$,$H(v)\in\mathcal{G}$); //利用哈希函数 H 对用户数据 v 进行编码
3. Perturb $\langle H,H(v)\rangle$ into $\langle H,y\rangle$ as $\langle H,y\rangle=\begin{cases}\langle H,H(v)\rangle, & \text{w.p. } 1-\gamma\left(\dfrac{g-1}{g}\right)\\ Uniform(\mathcal{G}), & \text{w.p. } \gamma\left(\dfrac{g-1}{g}\right)\end{cases}$;

   //对数据 v 所对应的哈希地址 $H(v)$进行本地扰动, “w.p”表示“以概率”
4. **return** $M=M\cup\langle H,y\rangle$.

SRR 算法首先在哈希家族$\mathcal{H}$中随机选择一个哈希函数 H 对用户的数据 v 进行本地哈希编码(步骤 2), 获得$\langle H,H(v)\rangle$后, 对其进行混洗随机扰动之后获得$\langle H,y\rangle$, 其中, 以概率 $1-\gamma(g-1/g)$扰动成$\langle H,H(v)\rangle$, 以概率$\gamma(g-1/g)$扰动成值域$\mathcal{G}$中的任意值(步骤 3).

由 LDP 中的随机应答机制可知: 每个用户使用的应答机制越具有随机性, 就越能够保护自己的数据. 而在 SDP 模型中, 混洗方把收集的所有消息随机排列成一个向量 M, 且 M 中蕴含着部分真实值 $H(v_i)$. 因此, SRR 比起 LDP 的随机应答机制向 v_i 添加的随机噪音相对较少. 然而, SRR 与 LDP 下的随机扰动机制隐私保护程度相同. 由 SRR 算法的步骤 3 可知: 在整个消息向量 M 中, 一部分是以概率 $1-\gamma(g-1/g)$生成的真实值, 另一部分是以概率$\gamma(g-1/g)$生成的随机值, 该算法正是利用部分随机值掩盖了另一部分真实值.

当混洗方$\mathcal{S}$获得 M 后, 需要对其中的真实值与随机值进行混洗排列. 文献[1−4]大都采用费雪耶兹(Fisher-Yates)传统随机排列算法来混洗 M 中的消息, 该算法一次随机排列的时间复杂度为 $O(n^2)$. 尽管费雪耶兹随机排列算法的每种排列结果是等概率的$(1/n!)$, 但当 n 非常大时, 相应的时间复杂度较高. 而如何在保证混洗效率的前提下设计高效的混洗算法至关重要. 本文基于堆排列技术设计了一种高效的消息向量混洗算法 MRS 实现$\{\langle H_1,y_1\rangle,\langle H_2,y_2\rangle,\ldots,\langle H_n,y_n\rangle\}\rightarrow\{\langle H_1,y_1\rangle,\langle H_2,y_2\rangle,\ldots,\langle H_n,y_n\rangle\}$的映射关系. 该算法的具体细节如算法 3 所示.

算法 3. MRS.

输入: 混洗方$\mathcal{S}$收集到的消息向量 $M=\{\langle H_1,y_1\rangle,\langle H_2,y_2\rangle,\ldots,\langle H_n,y_n\rangle\}$;

输出: 消息排列结果σ.

1. $\mathcal{Y}\leftarrow\varnothing$; //$\mathcal{Y}$表示全排列集合
2. Create a integer array B of size n to control the iteration; //数组 B 用来控制迭代次数
3. $B[1]=1$, $B[2]=2$, …, $B[i]=i$…;
4. $i=1$; //i 用来控制 M 的索引
5. **while** $i<n$ **do**
6. $B[i]\leftarrow B[i]-1$;
7. **if** i is odd **then**
8. $j\leftarrow B[i]$;
9. **else**
10. $j\leftarrow 0$;
11. *swapmessages*($\langle H_j,y_j\rangle,\langle H_i,y_i\rangle$); //交换第 i 与第 j 个消息位置
12. add ($\langle H_j,y_j\rangle,\langle H_i,y_i\rangle$) to $\mathcal{Y}$;
13. $i=1$;
14. **while** $B[i]==0$ **do**

15. $B[i]\leftarrow i$;

16. $i\leftarrow i+1$;

17. **end while**

18.**end while**

19. Uniformly pick a permutation σ from $\mathcal{Y}'$;

20. return σ.

在 MRS 算法中, 混洗方收集到消息向量 M 后, 对 M 进行均匀排列, 随机均匀地生成一个消息向量σ. 具体思路是:

- 当循环遍历第 i 个元素时, 如果 i 是奇数, 则交换第 1 个和第 i 个元素; 如果 i 是偶数, 则交换第 $j(j\in B[i])$个元素和第 i 个元素. 其中, 数组 $B[i]$用来控制迭代次数(步骤 5–步骤 11);
- 然后, 将交换后的前 i–1 个元素的排列附加到当前最后一个元素(第 i 个元素)(步骤 14–步骤 17).

MRS 算法生成所有全排列的时间复杂度为 $O(n^2\log n)$, 而该算法随机输出一个排列结果的时间复杂度为 $O(n\log n)$.

3.2.1 HP-SDP 算法的隐私性分析

给定分布式数据集 $D=\{v_1,v_2,\ldots,v_n\}$. 经过 SRR 扰动与 MRS 混洗后, 获得$\sigma=\{\langle H_1,y_1\rangle,\langle H_2,y_2\rangle,\ldots,\langle H_n,y_n\rangle\}$. 为了证明方便, 把 SRR 扰动操作与 MRS 混洗操作记作$\mathcal{S}\circ\mathcal{R}$, 令$\mathcal{M}(D)=\mathcal{S}\circ\mathcal{R}(D)$. 根据 HP-SDP 算法分析可知, 所有用户发送的消息经过$\mathcal{S}\circ\mathcal{R}$操作后能够满足$(\varepsilon_c,\delta)$-CDP, 其中, ε_c 表示中心化隐私预算.

定理 1. $\mathcal{M}(D)$满足(ε_c,δ)-中心化差分隐私, 其中, $\varepsilon_c\leqslant\sqrt{\dfrac{14\ln(2/\delta)(\mathrm{e}^{\varepsilon}+g-1)}{n-1}}$.

证明: 给定 D 及其近邻 D'. $\mathcal{Y}'$表示$\mathcal{M}$的输出值域, 即由$\langle H_i,y_i\rangle$组成, 其中, H_i 表示$\mathcal{H}$中的第 i 个哈希函数, $y_i=GRR(H_i(v_i))$. σ表示$\mathcal{Y}'$中任意一种随机排列结果.

若要证明$\mathcal{M}$满足(ε_c,δ)-CDP, 则需要证明 $\Pr_{y\sim\mathcal{M}(D)}\left[\dfrac{\Pr[\mathcal{M}(D)=y]}{\Pr[\mathcal{M}(D')=y]}\geqslant\mathrm{e}^{\varepsilon_c}\right]\leqslant\delta$ 即可.

假设 D 与 D'只在第 n 个用户的值不同, 即 $v_n\neq v_n'$, 第 n 个用户为目标攻击用户. 假设 $H(v_n)$被哈希到值域 $[g]$中的第 1 个位置, $H(v_n')$ 被哈希到值域$[g]$中的第 2 个位置, 则可以分两种情况证明上述不等式.

- 情况 1

如果第 n 个用户以概率$\gamma(g-1/g)$发送随机值, 则 $\Pr[\mathcal{M}(D)=y]=\Pr[\mathcal{M}(D')=y]$. 根据差分隐私定义可知, 该情况满足$(\varepsilon_c,\delta)$-CDP;

- 情况 2

如果第 n 个用户以概率 $1-\gamma(g-1/g)$发送真实值, 则需要$\mathcal{M}$方法对其进行保护. 该情况证明如下.

结合 MRS 算法可知, n 个用户的消息随机排列后存在 $n!$种σ, 则对 D 扰动和混洗后的概率如下所示:

$$\begin{aligned}\Pr[\mathcal{M}(D)=y]&=\sum_{\sigma}\Pr[\sigma]\Pr[\mathcal{M}(D)=y\mid\sigma]\\&=\sum_{\sigma}\frac{1}{n!}\Pr[\mathcal{M}(D)=y\mid\sigma]\\&=\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\Pr[H_{\sigma(i)}]I_{\{H_{\sigma(i)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\Pr[H_{\sigma(i)}]\frac{1}{g}\times\Pr[H_{\sigma(n)}]I_{\{H_{\sigma(n)}(v_n)=y_{\sigma(n)}\}}\right)\\&=\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\frac{1}{h}I_{\{H_{\sigma(i)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\frac{1}{h}\frac{1}{g}\times\frac{1}{h}I_{\{H_{\sigma(n)}(v_n)=y_{\sigma(n)}\}}\right)\end{aligned}\tag{8}$$

其中,

- $\Pr[\mathcal{M}(D)=y]$表示以随机排列σ为条件的$\mathcal{M}$输出概率;
- h 表示$\mathcal{H}$中哈希函数的个数;

- $\Pr[H_{\sigma(i)}]$表示随机排列的第 i 个位置所对应的哈希函数的抽样概率;
- I 为标识函数，当 $H_{\sigma(i)}(v_i)=y_i$ 时，该值为 1；否则为 0.

同理，对 D'有以下等式成立:

$$\begin{aligned}\Pr[\mathcal{M}(D')=y]&=\sum_{\sigma}\Pr[\sigma]\Pr[\mathcal{M}(D')=y\mid\sigma]\\&=\sum_{\sigma}\frac{1}{n!}\Pr[\mathcal{M}(D')=y\mid\sigma]\\&=\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\Pr[H_{\sigma(i)}]I_{\{H_{\sigma(n)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\Pr[H_{\sigma(i)}]\frac{1}{g}\times\Pr[H_{\sigma(n)}]I_{\{H_{\sigma(n)}(v'_n)=y_{\sigma(n)}\}}\right)\\&=\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\frac{1}{h}I_{\{H_{\sigma(n)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\frac{1}{h}\frac{1}{g}\times\frac{1}{h}I_{\{H_{\sigma(n)}(v'_n)=y_{\sigma(n)}\}}\right)\end{aligned}\tag{9}$$

则根据公式(8)与公式(9)可知:

$$\frac{\Pr[\mathcal{M}(D)=y]}{\Pr[\mathcal{M}(D')=y]}=\frac{\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\frac{1}{h}I_{\{H_{\sigma(i)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\frac{1}{h}\frac{1}{g}\times\frac{1}{h}I_{\{H_{\sigma(i)}(v_n)=y_{\sigma(n)}\}}\right)}{\sum_{\sigma}\frac{1}{n!}\left(\prod_{i\in[n-1]}\frac{1}{h}I_{\{H_{\sigma(i)}(v_i)=y_i\}}\times\prod_{i\in[n-1]}\frac{1}{h}\frac{1}{g}\times\frac{1}{h}I_{\{H_{\sigma(i)}(v'_n)=y_{\sigma(n)}\}}\right)}=\frac{\sum_{\sigma}I_{\{H_{\sigma(n)}(v_n)=y_{\sigma(n)}\}}}{\sum_{\sigma}I_{\{H_{\sigma(n)}(v'_n)=y_{\sigma(n)}\}}}=\frac{n_1}{n_2}\tag{10}$$

其中,

- n_1 表示哈希值域[g]中第 1 个位置的用户计数，该计数符合二项分布 $N_1\sim B(n-1,\gamma/g)+1$;
- n_2 表示哈希值域[g]中第 2 个位置的用户计数，该计数符合二项分布 $N_2\sim B(n-1,\gamma/g)$,

则 $\Pr_{y\sim\mathcal{M}(D)}\left[\frac{\Pr[\mathcal{M}(D)=y]}{\Pr[\mathcal{M}(D')=y]}\geqslant e^{\varepsilon_c}\right]=\Pr\left[\frac{N_1}{N_2}\geqslant e^{\varepsilon_c}\right]$.

设$\mu=E[N_2]=\gamma(n-1)/g$. $N_1/N_2\geqslant e^{\varepsilon_c}$ 蕴含着 $N_1\geqslant\mu e^{\varepsilon_c/2}$, $N_2\leqslant\mu e^{-\varepsilon_c/2}$，则根据布尔不等式可知:

$$\begin{aligned}\Pr\left[\frac{N_1}{N_2}\geqslant e^{\varepsilon_c}\right]&=\Pr[N_1\geqslant\mu e^{\varepsilon_c/2}\cup N_2\leqslant\mu e^{-\varepsilon_c/2}]\\&=\Pr[N_1\geqslant\mu e^{\varepsilon_c/2}]+\Pr[N_2\leqslant\mu e^{-\varepsilon_c/2}]-\Pr[(N_1\geqslant\mu e^{\varepsilon_c/2})\cap(N_2\leqslant\mu e^{-\varepsilon_c/2})]\\&\leqslant\Pr[N_1\geqslant\mu e^{\varepsilon_c/2}]+\Pr[N_2\leqslant\mu e^{-\varepsilon_c/2}]\\&=\Pr[N_2+1\geqslant\mu e^{\varepsilon_c/2}]+\Pr[N_2\leqslant\mu e^{-\varepsilon_c/2}]\\&=\Pr\left[N_2-\mu\geqslant\mu\left(e^{\varepsilon_c/2}-1-\frac{1}{\mu}\right)\right]+\Pr[N_2-\mu\leqslant\mu(e^{-\varepsilon_c/2}-1)]\end{aligned}\tag{11}$$

根据切诺夫不等式以及文献[3]中的切诺夫边界理论可知:

$$\Pr\left[N_2-\mu\geqslant\mu\left(e^{\varepsilon_c/2}-1-\frac{1}{\mu}\right)\right]\leqslant\exp\left(-\frac{\mu}{3}\left(e^{\varepsilon_c/2}-1-\frac{1}{\mu}\right)^2\right),$$

$$\Pr[N_2-\mu\leqslant\mu(e^{-\varepsilon_c/2}-1)]\leqslant\exp\left(-\frac{\mu}{2}(e^{-\varepsilon_c/2}-1)^2\right),$$

则公式(11)可表示为

$$\Pr\left[\frac{N_1}{N_2}\geqslant e^{\varepsilon_c}\right]\leqslant\exp\left(-\frac{\mu}{3}\left(e^{\varepsilon_c/2}-1-\frac{1}{\mu}\right)^2\right)+\exp\left(-\frac{\mu}{2}(1-e^{-\varepsilon_c/2})^2\right)\tag{12}$$

为了使得 $\Pr\left[\frac{N_1}{N_2}\geqslant e^{\varepsilon_c}\right]\leqslant\delta$ 成立，当$\varepsilon_c\leqslant1$ 时，公式(12)右侧的两项可表示为

$$\exp\left(-\frac{\mu}{2}(1-e^{-\varepsilon_c/2})^2\right)\leqslant\delta/2\tag{13}$$

$$\exp\left(-\frac{\mu}{3}\left(\mathrm{e}^{\varepsilon_c/2}-1-\frac{1}{\mu}\right)^2\right)\leqslant\delta/2 \tag{14}$$

结合公式(13)、公式(14), 根据文献[3]可知 $\mu=\dfrac{\gamma(n-1)}{g}\geqslant\dfrac{14\ln(2/\delta)}{\varepsilon_c^2}$, 进而可知 $\varepsilon_c\leqslant\sqrt{\dfrac{14\ln(2/\delta)(\mathrm{e}^{\varepsilon}+g-1)}{n-1}}$.

因此, $\mathcal{M}$满足 $\left(\sqrt{\dfrac{14\ln(2/\delta)(\mathrm{e}^{\varepsilon}+g-1)}{n-1}},\delta\right)$-中心化差分隐私.

3.2.2 HP-SDP 算法的可用性分析

HP-SDP 算法的可用性主要从直方图每个桶频率的无偏性、每个桶频率产生的方差以及每个桶频率产生的最大偏差来度量. 每个用户的类别数据经过 SRR 扰动与 MRS 混洗后会与真实值存在偏差. 收集者$\mathcal{A}$若不对每个桶的频率进行修正, 则发布结果会存在较大偏差.

定理 2. 假设 f_v 与 $\tilde{f}_v$ 分别表示桶(或类别数据)v 的真实频率与估计频率, 则 $E[\tilde{f}_v]=f_v$ 成立.

证明: 假设 v 被哈希且扰动成 y 的概率为 $\Pr[y]=f_v\left(1-\gamma\dfrac{g-1}{g}\right)+(1-f_v)\gamma\dfrac{g-1}{g}$, 假设 v 被哈希且扰动成非 y(即 y')的概率为 $\Pr[y']=f_v\gamma\dfrac{g-1}{g}+(1-f_v)\left(1-\gamma\dfrac{g-1}{g}\right)$, 结合 $\Pr[y]$与 $\Pr[y']$构造相应的极大似然函数如公式(15)所示:

$$\begin{aligned}L(f_v)&=[\Pr[y]]^{m_1}\times[\Pr[y']]^{n-m_1}\\&=\left[f_v\left(1-\gamma\frac{g-1}{g}\right)+(1-f_v)\gamma\frac{g-1}{g}\right]^{m_1}\times\left[f_v\gamma\frac{g-1}{g}+(1-f_v)\left(1-\gamma\frac{g-1}{g}\right)\right]^{n-m_1}\end{aligned} \tag{15}$$

其中, m_1 表示相应 v 值被哈希且扰动成 y 的用户个数, g 表示哈希函数的值域大小.

对公式(15)两边取对数, 可获得公式(16):

$$\ln(L(f_v))=m_1\ln\left[f_v\left(1-\gamma\frac{g-1}{g}\right)+(1-f_v)\gamma\frac{g-1}{g}\right]+(n-m_1)\ln\left[f_v\gamma\frac{g-1}{g}+(1-f_v)\left(1-\gamma\frac{g-1}{g}\right)\right] \tag{16}$$

然后对公式(16)两边的 f_v 求偏导后, 可获得公式(17):

$$\frac{\partial\ln(L(f_v))}{\partial f_v}=\frac{m_1\left(1-\gamma\dfrac{g-1}{g}-\gamma\dfrac{g-1}{g}\right)}{f_v\left(1-\gamma\dfrac{g-1}{g}\right)+(1-f_v)\gamma\dfrac{g-1}{g}}+\frac{(n-m_1)\left(\gamma\dfrac{g-1}{g}-1+\gamma\dfrac{g-1}{g}\right)}{f_v\gamma\dfrac{g-1}{g}+(1-f_v)\left(1-\gamma\dfrac{g-1}{g}\right)} \tag{17}$$

令公式(17)=0, 则可获得公式(18):

$$f_v\left(n-2n\gamma\frac{g-1}{g}\right)=m_1-n\gamma\frac{g-1}{g} \tag{18}$$

由于无法获知桶 v 的真实频率 f_v, 则需要根据公式(18)来表示出 f_v 的估计量, 如公式(19)所示:

$$\begin{aligned}\tilde{f}_v&=\frac{1}{n}\cdot\frac{m_1-n\gamma\dfrac{g-1}{g}}{1-2\gamma\dfrac{g-1}{g}}\\&=\frac{1}{n}\cdot\frac{\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}-n\gamma\dfrac{g-1}{g}}{1-2\gamma\dfrac{g-1}{g}}\end{aligned} \tag{19}$$

接下来结合公式(19)证明 $E[\tilde{f}_v]=f_v$ 成立:

$$E[\tilde{f}_v]=E\left(\frac{1}{n}\cdot\frac{\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}-n\gamma\frac{g-1}{g}}{1-2\gamma\frac{g-1}{g}}\right)=\frac{1}{n}\cdot\frac{E\left(\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}\right)-n\gamma\frac{g-1}{g}}{1-2\gamma\frac{g-1}{g}}$$

$$=\frac{1}{n}\cdot\frac{\left(nf_v\left(1-\gamma\frac{g-1}{g}\right)+(n-nf_v)\gamma\frac{g-1}{g}\right)-n\gamma\frac{g-1}{g}}{1-2\gamma\frac{g-1}{g}}$$

$$=\frac{1}{n}\cdot\frac{nf_v-nf_v\gamma\frac{g-1}{g}-nf_v\gamma\frac{g-1}{g}}{1-2\gamma\frac{g-1}{g}}=\frac{1}{n}\cdot nf_v=f_v.$$

证明完毕. □

由于 SRR 与 MRS 算法的本地扰动与混洗, 在估计 $\tilde{f}_v$ 的无偏性时会产生相应的误差, 而这种误差是衡量 HP-SDP 算法可用性的主要标准之一. 本文利用方差 $Var(\tilde{f}_v)$ 来度量 HP-SDP 算法产生的误差.

定理 3. 设 f_v 与 $\tilde{f}_v$ 分别表示桶 v 的真实频率与估计频率, 估计 $\tilde{f}_v$ 产生方差为 $Var[\tilde{f}_v]=\dfrac{\gamma\frac{g-1}{g}-\left(\gamma\frac{g-1}{g}\right)^2}{n\left(1-2\gamma\frac{g-1}{g}\right)^2}$.

证明: 根据公式(17)以及方差公式可知:

$$Var[\tilde{f}_v]=Var\left[\frac{1}{n}\cdot\frac{\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}-n\gamma\frac{g-1}{g}}{1-2\gamma\frac{g-1}{g}}\right]$$

$$=\frac{1}{n^2}\cdot\frac{Var\left(\sum_{i=1}^{n}I_{\{H_i(v)=y_i\}}-n\gamma\frac{g-1}{g}\right)}{\left(1-2\gamma\frac{g-1}{g}\right)^2}$$

$$=\frac{1}{n^2\left(1-2\gamma\frac{g-1}{g}\right)^2}\cdot\left[nf_v\left(1-\gamma\frac{g-1}{g}\right)\gamma\frac{g-1}{g}+(n-nf_v)\gamma\frac{g-1}{g}\left(1-\gamma\frac{g-1}{g}\right)\right]$$

$$=\frac{1}{n^2\left(1-2\gamma\frac{g-1}{g}\right)^2}\cdot\left[n\gamma\frac{g-1}{g}-n\left(\gamma\frac{g-1}{g}\right)^2\right]$$

$$=\frac{\gamma\frac{g-1}{g}-\left(\gamma\frac{g-1}{g}\right)^2}{n\left(1-2\gamma\frac{g-1}{g}\right)^2}.$$

□

根据定理 2 与定理 3 以及公式(6), 可以获得估计 $\tilde{f}_v$ 产生的均方差 MSE, 如定理 4 所示.

定理 4. 估计 $\tilde{f}_v$ 所产生的均方差 MSE 为 $Var(\tilde{f}_v)$.

证明: 根据公式(6)以及定理 2 和定理 3, $\tilde{f}_v$ 产生的均方差 MSE 为

$$MSE=\frac{1}{d}\sum_{v\in\mathcal{D}}E(f_v-\tilde{f}_v)^2=\frac{1}{d}\sum_{v\in\mathcal{D}}(Var[\tilde{f}_v]+[E[\tilde{f}_v]-f_v]^2)=\frac{1}{d}\sum_{v\in\mathcal{D}}Var[\tilde{f}_v]=Var[\tilde{f}_v].$$

□

根据定理 2 与定理 3 可以推理出每个桶中频率的无偏性与方差, 而如何度量 f_v 与 $\tilde{f}_v$ 之间的最大偏差是一个挑战性的问题. 接下来由定理 5 给出说明.

定理 5. 给定 f_v 与 $\tilde{f}_v$, 则 $\max_{v\in\mathcal{D}}|\tilde{f}_v-f_v|=O\left(\frac{\sqrt{\ln(1/\beta)}}{\varepsilon\sqrt{n}}\right)$ 至少以概率 $1-\beta$ 成立, 其中, n 为用户个数.

证明: 为了证明方便, 给定 n 中任意一个用户 u_i 所拥有的值为 v. 经过本地哈希编码后, 若其对应的编码满足 $H_i(v)=y_i$, 则 $c(v)=1$. 经过本地 SRR 扰动之后, v 的计数为 $c'(v)$. 如 SRR 算法中的第 3 行所示: 当扰动概率为 $1-\gamma(g-1/g)$时, $c'(v)=1$. 因此, 可知 $c'(v)-c(v)$为一个随机变量, 则以下针对 $c'(v)-c(v)$的方差推理成立:

$$Var[c'(v)-c(v)]=Var[c'(v)]=E[(c'(v))^2]-E[c'(v)]^2=E[(c'(v))^2]-(c(v))^2\leqslant E[(c'(v))^2]=1\cdot\left(1-\gamma\left(\frac{g-1}{g}\right)\right)^2=O(1/\varepsilon^2).$$

$c(v)$与 $c'(v)$的取值范围为{1,0}, 则随机变量 $c'(v)-c(v)$的取值范围为{−1,0,+1}, 于是可知:

$$|c'(v)-c(v)|\leqslant e^{s}-g+1/e^{s}+g-1<t.$$

根据伯恩斯坦不等式可知如下不等式成立:

$$\begin{aligned}\Pr[|f'(v)-f(v)|\geqslant nt]&\leqslant 2\times\exp\left(-\frac{(nt)^2}{2\cdot\sum_i^n Var[c_i'(v)-c_i(v)]+\frac{2nt}{3}\left(\frac{e^{\varepsilon}-g+1}{e^{\varepsilon}+g-1}\right)}\right)\\&\leqslant 2\times\exp\left(-\frac{nt^2}{\frac{2}{n}\cdot\sum_i^n Var[c_i'(v)-c_i(v)]+\frac{2t}{3}\left(\frac{e^{\varepsilon}-g+1}{e^{\varepsilon}+g-1}\right)}\right)\\&=2\times\exp\left(-\frac{nt^2}{O(1/\varepsilon^2)+t\cdot O(1/\varepsilon)}\right),\end{aligned}$$

则存在 $t=\sqrt{\ln(1/\beta)}/\varepsilon\sqrt{n}$, 使得$|f'(v)-f(v)|<t$ 以概率 $1-\beta$成立. □

定理 2–定理 5 分别从无偏性、方差、均方差与最大偏差估计了 HP-SDP 算法的可用性. 然而, 由于 SRR 本地扰动的原因, 可能会导致直方图的一些桶的频率出现负值, 或者所有桶的频率之和不等于 1. 由公式(19)可知: 所有桶的频率应位于区间[0,1], 并且所有桶的频率之和为 1. 基于此, 需要对所有桶的估计值进行后置处理, 而后置处理的结果需要满足一致性约束条件.

定义 4(一致性约束). 如若 HP-SDP 算法的输出结果 $\bar{F}$ 满足: (1) $\bar{F}$ 中的任意值属于[0,1]; (2) $\bar{F}$ 中的所有值之和等于 1, 则 HP-SDP 算法满足一致性约束条件.

因此, 如何在一致性约束条件下求出 $\bar{F}$, 是后置处理的关键问题所在. 本文把该问题转换为约束条件下的二次规划问题[9], 其形式化表达如公式(20)所示:

$$\begin{cases}\text{minimize}:\sum_{v\in\mathcal{D}}(\bar{f}_v-\tilde{f}_v)^2\\ \text{subject to}:\sum_{v\in\mathcal{D}}\bar{f}_v=1,\ \forall v:\bar{f}_v\geqslant 0\end{cases}\tag{20}$$

其中, $\forall v:\tilde{f}_v\in\tilde{F},\bar{f}_v\in\bar{F}$.

公式(20)即是 POP 后置处理算法的核心技术. 根据文献[21]中的 KKT 条件, 可以对公式(20)中的目标函数进行松弛处理, 如公式(21)所示:

$$\begin{cases}\text{minimize}:\sum_{v\in\mathcal{D}}(\bar{f}_v-\tilde{f}_v)^2+a_1+a_2\\ \text{where}\ \sum_{v\in\mathcal{D}}\bar{f}_v=1,\ \forall v:0\leqslant\bar{f}_v\leqslant 1\\ a_1=b\cdot\sum_{v\in\mathcal{D}}\bar{f}_v\\ a=\sum_{v\in\mathcal{D}}\lambda_v\cdot\bar{f}_v,\ \forall v:\lambda_v\cdot\bar{f}_v=0\end{cases}\tag{21}$$

根据文献[21]可知: 如果对 v 的值域$\mathcal{D}$进行划分$\mathcal{D}_0\cup\mathcal{D}_1$, 并且满足$\forall v\in\mathcal{D}_0$ 时, $\bar{f}_v=0$; 以及$\forall v\in\mathcal{D}_1$ 时,

$\overline{f}_v>0\wedge\lambda_v=0$. 进而可知$\mathcal{D}_1$中的 $\overline{f}_v$ 可行解为 $\overline{f}_v=\tilde{f}_v-\frac{1}{|\mathcal{D}_1|}(\sum_{v\in\mathcal{D}_1}\tilde{f}_v-1)$. 因此, HP-SDP 算法中的 POP 后置处理的结果如公式(22)所示:

$$\overline{f}_v=\begin{cases}\tilde{f}_v-\frac{1}{|\mathcal{D}_1|}(\sum_{v\in\mathcal{D}_1}\tilde{f}_v-1), & \forall v\in\mathcal{D}_1\\ 0, & \forall v\in\mathcal{D}_0\end{cases} \tag{22}$$

由文献[22]可知, 后置处理的结果同样满足(ε_c,δ)-中心化差分隐私.

4 实验结果与分析

实验平台是 4 核 Intel CPU(4 GHz)、8 G 内存、Win 7 系统, 代码采用 Python 实现. 实验采用 Normal 数据集、Zipf 数据集、IPUMS 数据集以及 Kosarak 数据集. 其中,

- Normal 与 Zipf 是两个合成数据集, 数据量为 600 000, 值域为 600;
- IPUMS 是 1940 年的美国人口普查数据集, 按照 1%的比例进行采样, 使用其中的城市属性, 数据中包含 602 325 个用户和 915 个城市;
- Kosarak 是在匈牙利网站上点击流的数据集, 包含 100 万个用户, 42 178 种不同的值, 对于每条数据流, 随机选取一项作为用户的数据.

4 种数据集的具体信息见表 1.

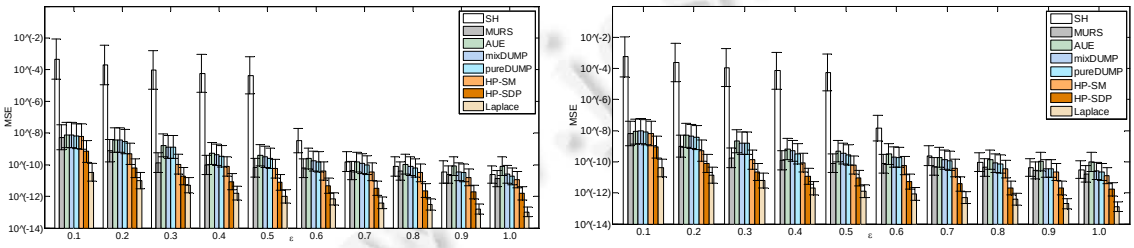
表 1 实验数据集描述

名称	分布	用户数	值域
Normal	Normal	600 000	600
Zipf	Zipf	600 000	600
IPUMS		602 325	915
Kosarak		1 000 000	42 178

结合上述数据集, 采用均方差(mean square error, MSE)度量 SH、MURS、AUE、mixDUMP、pureDUMP、HP-SM、HP-SDP、Laplace 算法[22]的误差. 其中, HP-SM 是 HP-SDP 算法去掉后置处理操作之后的算法. 隐私参数ε的选取分别为 0.1、0.2、0.3、0.4、0.5、0.6、0.7、0.8、0.9、1.0.

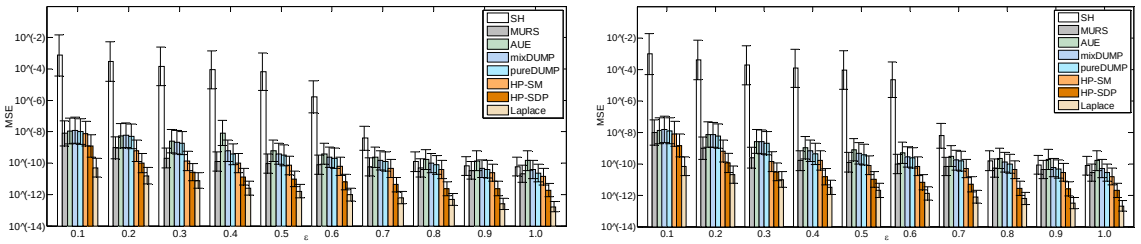
4.1 基于Normal和Zipf数据集的8种算法的MSE比较

图 3 和图 4 描述了上述 8 种算法在 Normal 和 Zipf 数据集上 MSE 值的比较结果. 由实验结果可以发现: 当ε从 0.1 变化到 1.0 时, 所有的 MSE 均呈下降趋势. 其原因是噪音的多少与ε成反比, ε越大, MSE 越小. HP-SM 和 HP-SDP 算法优于除 Laplace 算法之外的 6 种算法, 而 HP-SDP 算法的 MSE 接近 Laplace 算法的 MSE. 当δ=10^{-6}, ε从 0.1 变化到 0.5 时, SH 算法的 MSE 在 Normal 与 Zipf 数据集上比 HP-SM 算法高出 5 个与 6 个数量级, 比 HP-SDP 算法的 MSE 高出 6 个与 7 个数量级. mixDUMP、pureDUMP 与 SH 算法均采用 GRR 机制进行本地扰动, 然而该机制容易受到值域大小的影响, 值域越大, 精度越低. 因此, 这 3 种算法的精度低于 HP-SDP 算法. 当δ从 10^{-6} 变化到 10^{-9}, ε从 0.1 变化到 1.0 时, MURS 算法采用本地哈希技术所取得的精度接近 HP-SM, 但却低于 HP-SDP 算法. 其主要原因是 HP-SDP 算法利用了快速混洗与后置处理技术.



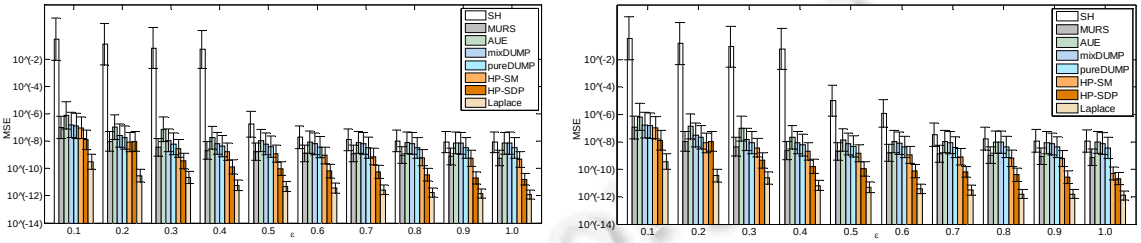
(a) δ=10^{-6} (b) δ=10^{-7}

图 3 基于 Normal 数据集的 MSE 对比

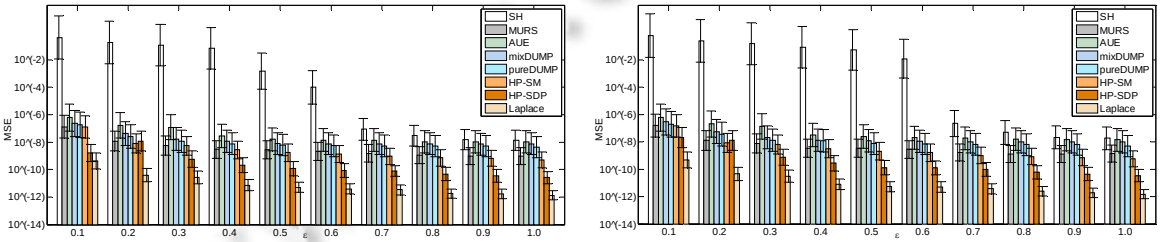


(c) δ=10^{-8} (d) δ=10^{-9}

图 3 基于 Normal 数据集的 MSE 对比(续)



(a) δ=10^{-6} (b) δ=10^{-7}

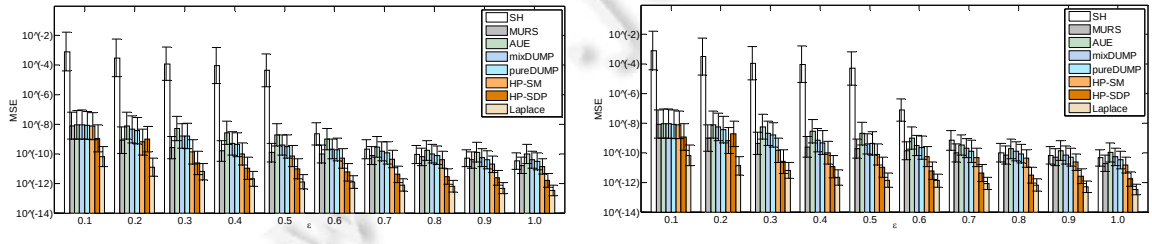


(c) δ=10^{-8} (d) δ=10^{-9}

图 4 基于 Zipf 数据集的 MSE 对比

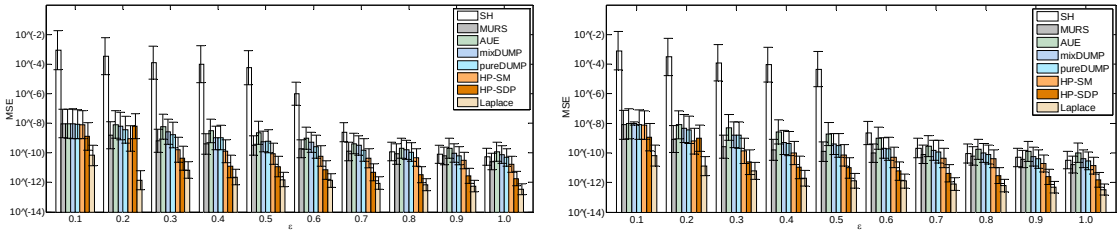
4.2 基于IPUMS与Kosarak数据集的8种算法的MSE比较

图 5 与图 6 分别描述了上述 8 种算法在 IPUMS 与 Kosarak 真实数据集上的 MSE 变化情况. 从图 5 与图 6 的实验结果可以看出: 当δ从 10^{-6} 变化到 10^{-9}, ε从 0.1 变化到 1.0 时, SH 算法的 MSE 在 IPUMS 与 Kosarak 数据集上比 HP-SM 算法高出 5 个数量级, 而 HP-SDP 算法的精度将近是 SH 算法的 1.7 倍. 相比于合成数据集, HP-SM 算法与 HP-SDP 算法的精度优势在 Kosarak 数据集上更加明显. 在ε越来越大时, HP-SDP 算法的精度更接近 Laplace 算法. 当δ=10^{-6}, 且ε从 0.1 变化到 0.5 时, mixDUMP 算法与 SH 算法的 MSE 比 HP-SDP 算法高出 3 个与 6 个数量级. HP-SDP 算法取得的精度是 MURS 算法的近 3 倍. 此外, 由于 AUE 自身不满足本地化差分隐私, 其 MSE 一直高于 HP-SM 算法与 HP-SDP 算法. 上述结果的主要原因是: HP-SDP 算法采用了基于二次规划技术的后置处理方法, 进一步提升了直方图发布的精度.



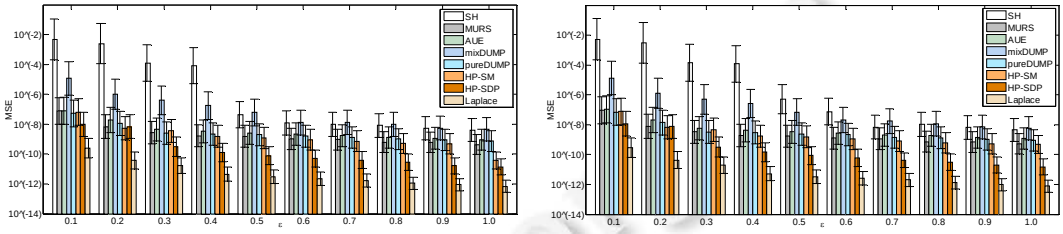
(a) δ=10^{-6} (b) δ=10^{-7}

图 5 基于 IPUMS 数据集的 MSE 对比

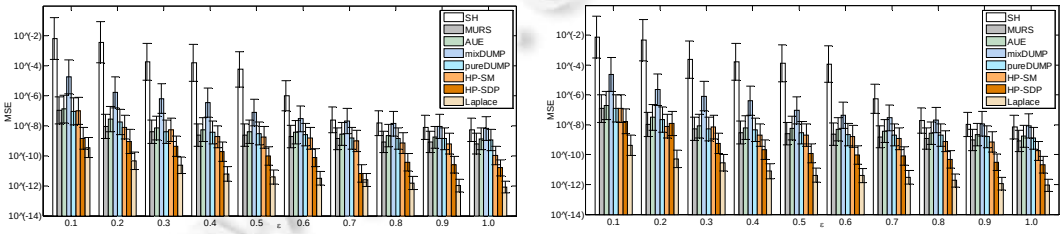


(c) δ=10^{-8} (d) δ=10^{-9}

图 5 基于 IPUMS 数据集的 MSE 对比(续)



(a) δ=10^{-6} (b) δ=10^{-7}



(c) δ=10^{-8} (d) δ=10^{-9}

图 6 基于 Kosarak 数据集的 MSE 对比

5 结束语

本文针对混洗差分隐私行下直方图的发布问题，结合中心化/本地化差分隐私存在的不足，提出了一种集成本地哈希编码、快速混洗以及后置处理的直方图发布算法 HP-SDP，该算法利用线性分解的形式重构了本地哈希扰动机制，利用类似于堆排序技术实现了随机混洗机制，最后利用二次规划技术实现了后置处理机制.通过 4 种数据集与现有 7 种方法进行均方误差对比分析，其结果表明，HP-SDP 算法的精度最接近中心化差分隐私下的拉普拉斯机制的精度.这也是混洗差分隐私框架的初衷.今后的工作考虑如下两个方面：(1) 如何在多消息单一混洗框架下实现直方图发布；(2) 如何在多消息多混洗框架下实现直方图发布.

References:

[1] Cheu A, Smith AD, Ullman JR. Manipulation attacks in local differential privacy. CoRR abs/1909.09630, 2019.

[2] Wang TH, Xu M, Ding BL, Zhou JR, Hong C, Huang ZC, Li NH, Jha S. Improving utility and security of the shuffler-based differential privacy. Proc. of the VLDB Endowment, 2020, 13(13): 3545−3558.

[3] Balle B, Bell J, Gascón A, Nissim K. The privacy blanket of the shuffle model. In: Proc. of the CRYPTO, Vol.2. 2019. 638−667.

[4] Balcer V, Cheu A. Separating local & shuffled differential privacy via histograms. In: Proc. of the ITC. 2020. 1:1−1:14

[5] Wang TH, Xu M, Ding BL, Zhou JR, Li NH, Jha S. Practical and robust privacy amplification with multi-party differential privacy. CoRR abs/1908.11515, 2019.

[6] Li XC, Liu WR, Chen ZY, Huang KZ, Qin Z, Zhang L, Ren K. DUMP: A dummy-point-based framework for histogram estimation in shuffle model. CoRR abs/2009.13738, 2020.

[7] Kairouz P, Bonawitz K, Ramage D. Discrete distribution estimation under local privacy. In: Proc. of the ICML. 2016. 2436−2444.

[8] Dwork C. Differential privacy. In: Proc. of the ICALP, Vol.2. 2006. 1−12.

[9] Hay M, Rastogi V, Miklau G, Suciu D. Boosting the accuracy of differentially private histograms through consistency. Proc. of the VLDB Endowment, 2010, 3(1): 1021−1032.

[10] Xu J, Zhang ZJ, Xiao XK, Yang Y, Yu G, Winslett M. Differentially private histogram publication. VLDB Journal, 2013, 22(6): 797−822.

[11] Erlingsson Ú, Pihur V, Korolova A. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Proc. of the ACM Conf. on Computer and Communications Security. 2014. 1054−1067.

[12] Bassily R, Smith AD. Local, private, efficient protocols for succinct histograms. In: Proc. of the STOC. 2015. 127−135.

[13] Bassily R, Nissim K, Stemmer U, Thakurta AG. Practical locally private heavy hitters. In: Proc. of the NIPS. 2017. 2288−2296.

[14] Bittau A, Erlingsson Ú, Maniatis P, Mironov I, Raghunathan A, Lie D, Rudominer M, Kode U, Tinnés J, Seefeld B. Prochlo: Strong privacy for analytics in the crowd. In: Proc. of the SOSP. 2017. 441−459.

[15] Cheu A, Smith AD, Ullman J, Zeber D, Zhilyaev M. Distributed differential privacy via shuffling. In: Proc. of the EUROCRYPT. 2019. 375−403.

[16] Balle B, Bell J, Gascon A, Nissim K. Differentially private summation with multi-message shuffling. arXiv: 1906.09116, 2019.

[17] Ishai Y, Kushilevitz E, Ostrovsky R, Sahai A. Cryptography from anonymity. IACR Cryptology ePrint Archive, 2006, 84: 20.

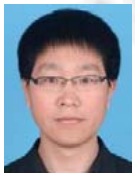
[18] Balle B, Bell J, Gascon A, Nissim K. Private summation in the multi-message shuffle model. arXiv: 2002.00817, 2020.

[19] Warner SL. Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 1965, 60(309): 63−69.

[20] Wang TH, Blocki J, Li NH, Jha S. Locally differentially private protocols for frequency estimation. In: Proc. of the USENIX Security Symp. 2017. 729−745.

[21] Wang TH, Lopuhaä-Zwakenberg M, Li ZT, Skoric B, Li NH. Locally differentially private frequency estimation with consistency. In: Proc. of the NDSS. 2020.

[22] Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations & Trends® in Theoretical Computer Science, 2014, 9(3-4): 211−407.



张啸剑(1980－), 男, 博士, 副教授, CCF 会员, 主要研究领域为数据隐私保护, 差分隐私, 联邦学习.

徐雅鑫(1996－), 女, 硕士生, 主要研究领域为本地化差分隐私, 混洗差分隐私.

夏庆荣(1996－), 男, 硕士生, 主要研究领域为隐私保护.