混洗差分隐私研究综述

张啸剑[1,2],王浩锋[1],傅继彬[1]
(1. 河南财经政法大学计算机与信息工程学院,河南郑州 450046;
2. 河南省物联大数据处理与安全工程技术研究中心,河南郑州 450121)

摘 要: 基于中心化差分隐私(Central Differential Privacy,CDP)与本地化差分隐私(Local Differential Privacy,LDP)的数据查询和分析已得到了研究者的广泛关注. 数据查询与分析的解决方法在CDP/LDP下取得不断突破的同时也凸显出相应的局限性,其局限性源自CDP/LDP是针对收集者信任度变化而设置的两个极端模型. CDP假设用户完全信任收集者,收集者结合用户的原始数据产生噪声来响应分析者的查询,响应的误差较低. 然而,该模型中的用户无法掌控自己的原始隐私数据. LDP假设用户不信任收集者,用户只是把本地扰动结果传输给收集者. 然而,该模型下查询与分析的误差很高. 混洗差分隐私(Shuffled Differential Privacy,SDP)模型的出现有效均衡了CDP与LDP之间的矛盾. 本文对SDP的保护模型、实现机制、研究方向以及存在的问题进行系统地综述. 首先介绍SDP的理论基础,主要包括SDP模型、SDP框架以及满足SDP算法的核心思想. 重点介绍当前该领域的研究热点:聚集查询估计、直方图估计、频率/均值估计以及机器/联邦学习等,对相应的研究热点进行总结与归纳. 在对已有技术深入对比分析的基础上,指出了混洗差分隐私保护技术的未来发展方向.

关键词: 中心化差分隐私;本地化差分隐私;混洗差分隐私;数据查询;数据分析;机器学习

基金项目: 国家自然科学基金(No.62072156);河南省高等学校重点科研项目计划基础研究专项(No.25ZX012);河南省自然科学杰出青年基金(No.252300421061)

中图分类号: TP309.2 **文献标识码:** A **文章编号:** 0372-2112(2025)12-4787-24

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250017

A Survey on Shuffled Differential Privacy

ZHANG Xiao-jian[1,2], WANG Hao-feng[1], FU Ji-bin[1]
(1. *School of Computer & Information Engineering, Henan University of Economics and Law, Zhengzhou, Henan* 450046, *China*;
2. *Henan Engineering Technology Research Center of IoT Big Data Processing and Security, Zhengzhou, Henan* 450121, *China*)

Abstract: Query and analysis of users' data with centralized differential privacy (CDP) and local differential privacy (LDP) have attracted considerable attention in recent years. The solutions to this problem have been proposed constantly, and the corresponding limitations are also highlighted, which originate from the fact that CDP and LDP are the two extreme models with the changing for collector's trust. In the CDP model, users fully trust the collector, and report their raw data. The collector perturbs the raw data to respond to the query, which error is low. Users in the CDP model, however, cannot control their raw private data. While, in the LDP model, users do not trust the collector and only report the noise data. The query error over the noise reports is high. The shuffled differential privacy (SDP) model effectively balances the contradiction between CDP and LDP. This paper surveys the state of the art of SDP for data query and analysis. The mechanisms and properties of this model are described, while our focus is put on summation query, histogram estimation, frequency and means estimation, and machine /federated learning, etc. Following the comprehensive comparison and analysis of existing works, future research directions are put forward.

Key words: central differential privacy; local differential privacy; shuffled differential privacy; data queries; data analysis; machine learning

Foundation Item(s): National Natural Science Foundation of China (No.62072156); Basic Research Special Projects of Key Research Projects in Higher Education Institutes in Henan Province (No.25ZX012); Natural Science Outstanding Youth Science Foundation of Henan Province (No.252300421061)

收稿日期:2025-01-03;录用日期:2025-03-24;责任编辑:覃怀银

1 引言

由 Dwork 提出的差分隐私保护模型凭借其坚实的数学基础得到国内外学者的广泛研究. 过去十几年中学术界通常采用中心化差分隐私(Central Differential Privacy, CDP)与本地化差分隐私(Local Differential Privacy, LDP)作为数据查询与分析的保护模型. 然而, CDP 与 LDP 是针对收集者信任度变化而设置的 2 个极端模型. CDP 模型假设用户完全信任收集者, 收集者掌控所有用户的原始数据. 结合用户原始数据添加适当噪声来响应分析者的查询. 该模型下数据查询与分析误差比较低. 例如, 利用拉普拉斯机制[1]响应聚集求和的查询误差为 $O(1/\varepsilon)$[2]. LDP 模型假设用户不信任收集者, 原始数据掌控在用户自己手中. 每个用户把自身数据本地扰动后发送给收集者. 该模型下查询与分析误差很高. 例如, 聚集求和查询的误差为 $O(\sqrt{n}/\varepsilon)$[3]. LDP 模型对收集者的信任度假设比较符合当前用户的隐私需求, 进而促使谷歌、微软、苹果以及优步等公司把该模型商用到自己的产品中. 然而, LDP 模型的查询与分析精度达不到 CDP 模型的精度, 隐私预算的设定值大于 CDP 模型下的设定值. 与 LDP 模型相比, CDP 模型尽管查询与分析误差低, 但是该模型没有 LDP 模型安全, 用户无法掌控自己的原始数据.

由上述聚集求和查询例子可知, CDP 模型与 LDP 模型的查询与分析误差存在很大间隙. 因此, 我们想问: 在 CDP 模型与 LDP 模型之间是否存在这样一种模型, 它能够为用户提供类似于 LDP 模型保护的力度, 也能为收集者提供接近于 CDP 模型的查询与分析精度. 混洗差分隐私模型(Shuffled Differential Privacy, SDP)的出现能够同时满足这两种需求. 在 SDP 模型中, 用户把自身数据本地扰动成消息向量, 再把消息发送给混洗方. 混洗方随机排列并且混洗后发送给收集者. 混洗之后的消息向量通常满足 CDP, 并取得接近于 CDP 模型的误差精度. 例如, 聚集求和查询的误差为 $O\left(\log(n/\delta)/\varepsilon\right)$[4]. 因此, 本文综述 SDP 模型下最新研究进展和研究方向, 一方面对 SDP 模型的研究背景、基本定义、实现机制以及其与 CDP/LDP 模型的区别进行阐述; 另一方面, 对当前 SDP 的研究方向进行分析, 并阐述最新研究进展. 着重介绍 SDP 模型下的隐私保护框架、攻击模型、隐私放大, 以及对目前应用于统计查询(聚集查询、频数统计、直方图估计等)、机器学习与联邦学习等方面的方法进行对比分析. 最后, 针对 SDP 模型的特性, 提出未来研究方向并加以具体分析.

2 基础知识

2.1 中心化/本地化/混洗差分隐私定义

设 $D(D\in\mathcal{D})$ 为分布式数据集, $D=\{v_1, v_2, \cdots, v_n\}$ 由 n 个数据(类别型或者数值型数据)构成. n 个数据分布在 n 个用户手中. 在介绍技术细节之前, 首先介绍 CDP、LDP 以及 SDP 的形式化定义.

定义 1 中心化差分隐私[1]. 设 D 与 D' 相差 1 条记录. 给定一个查询或者分析协议 $\mathcal{P}=(\mathcal{R},\mathcal{A},\varepsilon)$, $\mathcal{Y}$ 为 $\mathcal{P}$ 的输出域, 对于 $\mathcal{Y}$ 的任意子集 $\mathcal{O}\subseteq\mathcal{Y}$, 若 $\mathcal{P}$ 在 D 与 D' 上任意输出结果的概率满足下列不等式, 则 $\mathcal{P}$ 满足 (ε,δ)-CDP.

$$\Pr\left[\mathcal{P}(D)\in\mathcal{O}\right]\leqslant \mathrm{e}^{\varepsilon}\times\Pr\left[\mathcal{P}(D')\in\mathcal{O}\right]+\delta \tag{1}$$

其中, ε 表示隐私预算, $\delta(\delta\in(0,1])$ 为隐私泄露的风险概率, $\mathcal{A}$ 表示收集者, $\mathcal{R}$ 表示中心化扰动机制, $\delta=0$ 时, 为纯 ε-CDP.

定义 2 本地化差分隐私[5]. 设 v_i 与 v_i' 为 D 上任意 2 条记录. 给定查询分析协议 $\mathcal{P}=(\mathcal{R},\mathcal{A},\varepsilon)$, $\mathcal{Y}$ 为 $\mathcal{R}$ 的输出域, 对于 $\mathcal{Y}$ 的任意子集 $\mathcal{O}\subseteq\mathcal{Y}$, 若 $\mathcal{R}$ 在 v_i 与 v_i' 上任意输出结果的概率满足下列不等式, 则 $\mathcal{P}$ 满足 (ε,δ)-LDP.

$$\Pr\left[\mathcal{R}\left(v_i\right)\in\mathcal{O}\right]\leqslant \mathrm{e}^{\varepsilon}\times\Pr\left[\mathcal{R}\left(v_i'\right)\in\mathcal{O}\right]+\delta \tag{2}$$

其中, ε 表示隐私预算, $\delta(\delta\in(0,1])$ 为隐私泄露的风险概率, $\mathcal{A}$ 表示收集者, $\mathcal{R}$ 表示本地扰动机制, $\delta=0$ 时, 为纯 ε-LDP.

定义 3 混洗差分隐私[6]. 给定 $\mathcal{P}=(\mathcal{R},\mathcal{A},\varepsilon)$. 每个用户 u_i 利用 $\mathcal{R}:\mathcal{D}\to\mathcal{Y}$ 扰动记录 v_i: $y_i=\mathcal{R}(v_i)$. 令 $M=\{y_1,y_2,\cdots,y_n\}$ 为混洗方收集到的消息向量, $\mathcal{S}(M)$ 为混洗后的输出结果, 其值域为 $\mathcal{Y}'$. 对于 $\mathcal{Y}'$ 的任意子集 $\mathcal{O}\subseteq\mathcal{Y}'$, 如果 $\mathcal{S}(M):\mathcal{Y}\to\mathcal{Y}'$ 满足下列不等式, 则 $\mathcal{S}(M)$ 满足 (ε,δ)-CDP, 进而 $\mathcal{P}$ 满足 (ε,δ)-SDP.

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

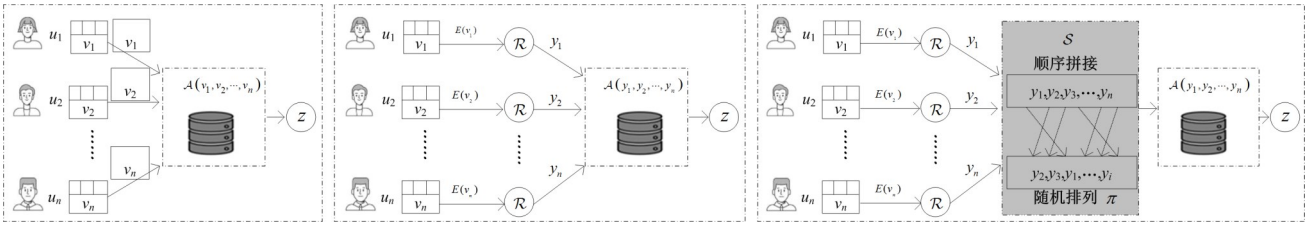
其中, ε 表示隐私预算, δ 为 $(\delta\in(0,1])$ 隐私泄露风险的概率, $\mathcal{A}$ 表示收集者, $\mathcal{R}$ 表示本地扰动机制, $\mathcal{S}$ 表示混洗方协议, $\delta=0$ 时, 为纯 ε-SDP. 如果从 (ε,δ)-CDP 的角度理解 SDP, 即删除或添加一条消息不会影响混洗方发布消息的总体分布.

2.2 中心化/本地化/混洗差分隐私的实现框架

图 1 描述了 CDP、LDP 以及 SDP 三种差分隐私保护模型的实现框架. 图 1(a)描述了 CDP 保护框架. 在该框架中, 每个用户 u_i 把自己的原始数据 v_i 发送给收集者 $\mathcal{A}$. 收集者 $\mathcal{A}$ 利用噪声机制(拉普拉斯机制[1]、高斯机制[7]等)生成满足 (ε,δ)-CDP 的噪声结果 Z. 由图 1(a)可知, $\mathcal{A}$ 的全局敏感性[1] $\Delta\mathcal{A}$ 与 ε 直接决定着 Z 的可用性. 若能较好地控制 $\Delta\mathcal{A}$ 大小以及合理地分配 ε, 则 Z 具有较高的可用性. 目前控制 $\Delta\mathcal{A}$ 大小的常用技术包括裁剪[8]、边界约束[9]等. ε 的分配策略包括均匀分配[10]、几何分配[10]以及自适应分配[11]等. CDP 保护框架的一个重要前提是每个 u_i 完全信任 $\mathcal{A}$, 并且认为 $\mathcal{A}$ 不会泄露自己的敏感数据 v_i. 然而在实际应用系统中, u_i 依旧迟疑共享自己的数据给 $\mathcal{A}$, 其主

要原因在于实际的系统部署很难找到完全可信的收集者$\mathcal{A}$.

CDP框架下每个u_i的担忧促进了LDP模型的发展. 图1(b)解释了LDP框架下每个用户的隐私保护原理. 该框架允许每个用户u_i本地扰动自身数据v_i,将扰动后的消息y_i报告给$\mathcal{A}$. 根据图1(b)可知$y_i=\mathcal{R}\left(E(v_i)\right)$,其中$E$表示本地编码机制,$\mathcal{R}$表示本地扰动机制. LDP框架下,收集者$\mathcal{A}$仅利用$\{y_1,y_2,\cdots,y_n\}$输出查询结果$Z$. LDP框架成立的前提是$u_i$完全不信任收集者$\mathcal{A}$,$Z$的可用性直接由$E$与$\mathcal{R}$决定. 好的编码机制$E$不但能够减少$u_i$与$\mathcal{A}$之间的通信代价,还能较好地保留$v_i$内在的信息. 一元编码[12]、哈希编码[13]、矩阵转换[14]、哈德码转换[15]等均是常用的编码机制. Rappor[12]、OUE[13]、OLH[13]、HRR[16]、GRR[5]、Lap[1]、Duchi[17]、PM[18]是常用的本地扰动机制. 尽管研究者追求LDP模型下理想的编码机制与扰动机制,然而该模型所产生的估计方差不可避免地与用户的个数n成线性关系. 例如,LDP模型下采用Lap机制[1]估计均值的方差为$n8/\varepsilon^2$.



(a) 中心化差分隐私保护框架　(b) 本地化差分隐私保护框架　(c) 混洗差分隐私保护框架

图1　三种差分隐私模型的保护框架

通过分析图1(a)与图1(b)可知,CDP与LDP两种框架可用性之间存在很大的鸿沟. 这种鸿沟主要源自用户对收集者$\mathcal{A}$的信任度假设的变化. 为了填补这种鸿沟,谷歌Brian团队提出了“编码-混洗-分析”ESA架构[19],首次把混洗技术融入LDP框架之中,进而产生SDP保护框架. 该架构把LDP扰动的消息向量以匿名通道的方式发送给混洗方,使得收集者$\mathcal{A}$无法重甄别目标用户的身份. 图1(c)描述了SDP通用性框架. 类似于LDP框架中的本地操作,u_i生成消息$y_i=\mathcal{R}\left(E(v_i)\right)$,并将$y_i$报告给混洗方$\mathcal{S}$,$\mathcal{S}$利用随机排列技术$\pi$对所有用户的消息向量进行拼接与随机排列操作,即$\pi$:$\{y_1,y_2,\cdots,y_n\}\mapsto\{y_1,y_2,\cdots,y_n\}$成立. 混洗方$\mathcal{S}$把混洗后的消息向量$\{y_1,y_2,\cdots,y_n\}$发送给收集者$\mathcal{A}$. SDP框架能够确保混洗后消息向量满足$(\varepsilon,\delta)$-CDP,并完成从LDP向CDP的过渡. 即SDP模型能够为用户提供类似于LDP模型下的保护程度,也能够为收集者$\mathcal{A}$提供类似于CDP模型下的查询与分析精度.

2.3　中心化/本地化/混洗差分隐私的实现机制

噪声机制与随机响应机制是实现CDP、LDP与SDP保护的主要技术. 阐述3种模型常用的实现机制,解释不同机制之间的逻辑关系与区别.

2.3.1　CDP模型的实现机制

拉普拉斯机制、高斯机制以及指数机制是实现CDP的3种常用机制. 由定义1可知,给定D及其近邻D',则$\mathcal{P}(D)=\mathcal{A}(D)+z$成立,其中$z$表示随机噪声的大小. $\mathcal{A}$的全局敏感性$\Delta\mathcal{A}$与隐私预算ε直接决定着z值的大小. 全局敏感性如式(4)所示:

$$\Delta\mathcal{A}_p=\max_{D\sim D'}\left\|\mathcal{A}(D)-\mathcal{A}(D')\right\|_p \tag{4}$$

(1)若采用拉普拉斯机制[1]生成随机噪声z,则z满足式(5),其形式可表示为Lap($\Delta\mathcal{A}/\varepsilon$).

$$\Pr[z]=\frac{\varepsilon}{2\Delta\mathcal{A}_1}\cdot\exp\left(-\frac{\varepsilon|z|}{\Delta\mathcal{A}_1}\right) \tag{5}$$

(2)若采用高斯机制[7](Gau)生成随机噪声z,则z满足高斯分布,如式(6)所示:

$$z\sim\mathcal{N}\left(0,\left(\frac{\Delta\mathcal{A}_2}{\varepsilon}\sqrt{C\log\left(1/\delta\right)}\right)^2\right) \tag{6}$$

其中,C为常数,用来约束$\Delta\mathcal{A}_2$的大小.

由式(5)与式(6)可知,拉普拉斯机制与高斯机制通常保护数值函数的计算过程. 文献[20]把打分函数融入指数机制来保护类别数据的计算过程. 打分函数通常用来衡量某个类别数据在数据集D中的重要程度.

(3)若收集者$\mathcal{A}$采用指数机制[20]保护类别数据r,则r被输出的概率满足式(7):

$$\Pr[r]\propto\exp\left(\frac{\varepsilon}{2\Delta u}\cdot u(D,r)\right) \tag{7}$$

其中,打分函数$u(D,r)$可衡量类别数据r在D中的重要程度,$\Delta u=\max_r\max_{D\sim D'}\left|u(D,r)-u(D',r)\right|$.

根据式(5)~式(7)可知,高斯机制与拉普拉斯机制是指数机制的2种特例. 其主要原因在于拉普拉斯机制与高斯机制产生的数值型噪声z可以转换成打分函数的形式. 符合拉普拉斯机制的打分函数$u(D,z)=-\left\|z-\mathcal{A}(D)\right\|_1$,符合高斯机制的打分函数$u(D,z)=-\left\|z-\mathcal{A}(D)\right\|_2$. 因此,我们把式(5)与式(6)写成式(7)的形式,如式(8)、式(9)所示:

$$\Pr_{\text{Lap}}[z]\propto\exp\left(-\frac{\varepsilon\left\|z-\mathcal{A}(D)\right\|_1}{\Delta\mathcal{A}_1}\right) \tag{8}$$

$$\Pr_{\text{Gau}}[z]\propto\exp\left(-\frac{\varepsilon^2\left\|z-\mathcal{A}(D)\right\|_2^2}{C\left(\Delta\mathcal{A}_2\right)^2\log 1/\delta}\right) \tag{9}$$

收集者$\mathcal{A}$在采用CDP模型保护每个用户的敏感数据时,通常采用CDP模型的序列组合性[21]与并行组合性[21]两种特性来证明整个处理过程满足(ε,δ)-CDP.

2.3.2 LDP模型的实现机制

WRR[22]是实现(ε,δ)-LDP的早期代表性机制. 该机制思想是用户在响应敏感的布尔问题时,以概率p真实应答,以$1-p$给出相反的应答. 为了使WRR满足(ε,δ)-LDP,通常设置$\ln(p/1-p)\leqslant\varepsilon$,进而可知$p=\mathrm{e}^{\varepsilon}/1+\mathrm{e}^{\varepsilon}$. 由于WRR仅能本地扰动值域$d$=2的0/1值,进而出现了$d$>2的扰动机制GRR[5].

(1)GRR扰动机制. 任意给定数据v_i与v_j,且v_i、$v_j\in\{1,2,\cdots,d\}$,其中d为值域大小,GRR机制见式(10):

$$\Pr\left[\mathrm{GRR}\left(v_i\right)=v_j\right]=\begin{cases}p=\dfrac{\mathrm{e}^{\varepsilon}}{\mathrm{e}^{\varepsilon}+d-1}, & v_i=v_j\\ q=\dfrac{1-p}{d-1}, & v_i\neq v_j\end{cases} \tag{10}$$

GRR机制结合整个值域d本地扰动给定的值,所产生的误差为$\Theta(d/\varepsilon\sqrt{n})$. 该机制的误差与通信代价直接受值域$d$的影响. 为了弥补GRR机制的不足,出现了以一元编码、哈希编码与Hadamard转换为基础的OUE机制[13]、OLH机制[13]与HRR机制[16].

(2)OUE扰动机制. 给定v_i且$v_i\in\{1,2,\cdots,d\}$,OUE利用一元编码把v_i编码成长度为d的0-1向量,记为$\boldsymbol{B}$. $\boldsymbol{B}$中仅第v_i位置为1,其余$d-1$位值为0. OUE本地扰动的思想如式(11)所示:

$$\Pr\left[\boldsymbol{B}'[i]=1\right]=\begin{cases}p=\dfrac{1}{2}, & B[i]=1\\ q=\dfrac{1}{\mathrm{e}^{\varepsilon}+1}, & B[i]=0\end{cases} \tag{11}$$

其中,$\boldsymbol{B}'$为扰动后的向量.

(3)OLH扰动机制. 给定v_i且$v_i\in\{1,2,\cdots,d\}$,OLH机制利用哈希簇$\mathcal{H}$把v_i哈希编码成$\{1,2,\cdots,g\}(g\ll d)$中某个哈希值,即$x=H(v_i)$,且$x\in\{1,2,\cdots,g\}$,$H\in\mathcal{H}$. OLH本地扰动的思想如式(12)所示,其中$y\in\{1,2,\cdots,g\}$.

$$\Pr\left[\mathrm{OLH}(x)=y\right]=\begin{cases}p=\dfrac{\mathrm{e}^{\varepsilon}}{\mathrm{e}^{\varepsilon}+g-1}, & x=y\\ q=\dfrac{1}{\mathrm{e}^{\varepsilon}+g-1}, & x\neq y\end{cases} \tag{12}$$

(4)HRR扰动机制. 给定v_i且$v_i\in\{1,2,\cdots,d\}$,d为2的幂次方,利用Hadamard转换$\phi\in\{\pm1/\sqrt{d}\}^{d\times d}$把$v_i$转换为长度为$d$的向量$v_i'=\{\pm1/\sqrt{d}\}^d$. 用户$u_i$随机选取一个索引值$j\in[d]$,对$v_i'[j]$使用WRR进行扰动. HRR本地扰动的思想如式(13)所示,其中$y_i[j]\in\{\pm1/\sqrt{d}\}$.

$$\Pr\left[y_i[j]=\frac{1}{\sqrt{d}}\right]=\begin{cases}\dfrac{\mathrm{e}^{\varepsilon}}{\mathrm{e}^{\varepsilon}+1}, & v_i'[j]=\dfrac{1}{\sqrt{d}}\\ \dfrac{1}{\mathrm{e}^{\varepsilon}+1}, & v_i'[j]=\dfrac{-1}{\sqrt{d}}\end{cases} \tag{13}$$

OLH机制与HRR机制的误差均为$O\left(\sqrt{\log d}/\varepsilon\sqrt{n}\right)$. 相比于GRR机制,尽管OLH与HRR机制的精度明显得到提升,但是还是无法接近CDP模型下的精度.

2.3.3 SDP模型的实现机制

根据图1(c)可知,用户u_i在数据v_i报告给混洗方$\mathcal{S}$之前,同样需要本地扰动v_i值. SDP模型的本地扰动机制与LDP模型的扰动机制存在什么样的本质区别? 为什么SDP模型能够确保混洗后的消息向量满足(ε,δ)-CDP? 设$\mathcal{R}$为LDP模型的扰动机制. SDP模型的扰动思想是把$\mathcal{R}$的输出概率分解为2种概率的线性组合形式,如式(14)所示:

$$\begin{aligned}&\forall_{y\in[d]}\Pr\left[\mathcal{R}\left(v_i\right)=y\right]\\&=\left(1-\gamma\right)I_{\{y=v_i\}}+\gamma\Pr\left[\mathrm{Unif}([d])=y\right]\end{aligned} \tag{14}$$

其中,I为标识函数,若$y=v_i$则I=1. 即y以$(1-\gamma)$的概率等于v_i值,以γ的概率均匀随机地从值域$[d]$中选取另外一个值. 文献[4]把GRR机制的输出概率分布分解成部分真实值分布与均匀随机噪声分布的线性组合,用户利用组合分布本地扰动数据. 具体形式见式(15):

$$\begin{aligned}&\forall_{y\in[d]}\Pr\left[\mathrm{GRR}\left(E\left(v_i\right)\right)=y\right]\\&=\left(1-\gamma\right)I_{\{y=E(v_i)\}}+\gamma\Pr\left[\mathrm{Unif}([d])=y\right]\end{aligned} \tag{15}$$

其中,$\gamma=d/(\mathrm{e}^{\varepsilon}+d-1)$,$\Pr[\mathrm{Unif}[d]=y]=1/d$,$I$为标识函数,若$y=E(v_i)$,则$I$=1.

根据LDP保护模型下的本地扰动机制可知,每个用户的报告值越具有随机性,对其真值的保护效果越好. 而在SDP保护模型中,混洗方$\mathcal{S}$把收集到的所有消息随机排列成一个多重集合M,且M蕴含着部分真实值(参阅式(14)与式(15)). SDP保护模型下的扰动机制比起LDP模型下的扰动机制向v_i添加的随机噪声相对较少. 其主要原因是在集合M中,一部分是以概率$1-\gamma(d-1/d)$生成的真实值,另一部分是以概率$\gamma(d-1/d)$生成的随机值,其扰动内涵正是利用部分随机值掩盖了另一部分真实值. 从混洗方$\mathcal{S}$的角度来分析,我们希望

混洗后的结果满足(ε_c,δ)-CDP,也就是式(16)成立. 文献[22]从二项分布的角度证明了式(16):

$$\Pr_{y\sim y}\left[\frac{\Pr\left[S\left(M\left(D\right)\right)=y\right]}{\Pr\left[S\left(M\left(D'\right)\right)=y\right]}\geqslant e^{\varepsilon_c}\right]\leqslant\delta \tag{16}$$

在证明混洗结果满足(ε_c,δ)-CDP的过程中,可以找到ε_l与ε_c之间的逻辑关系,其中ε_l与ε_c分别为LDP与CDP模型下的隐私预算. 例如文献[22]给出的关系如式(17)所示:

$$\varepsilon_c=O\left(\log\left(\frac{n\varepsilon_l^{\,2}}{\log(1/\delta)}\left(1-\frac{\gamma}{14}\right)\right)\right) \tag{17}$$

从式(17)可知,SDP模型保护可以实现隐私放大.

定义4　隐私放大. 每个用户利用满足(ε_l,δ)-LDP的本地扰动机制产生消息并发送给混洗方$\mathcal{S}$. $\mathcal{S}$拼接排列所有消息向量后发给收集者. 则该混洗协议$\mathcal{S}$满足(ε_c,δ)-CDP,且$\varepsilon_c\ll\varepsilon_l$. 从差分隐私定义可知,隐私预算越小保护程度越强. 这种隐私增强效应即是隐私放大.

表1描述了CDP/LDP/SDP三种模型的异同点. CDP模型与LDP模型之间存在着隐私性与可用性的平衡问题,进而产生了SDP模型. 对收集者$\mathcal{A}$的信任角度变换,这三种模型的优缺点不尽相同. SDP模型下对$\mathcal{A}$信任度越靠近CDP模型下的信任度设置,查询与分析的精度越高. SDP模型能够通过混洗方$\mathcal{S}$的随机排列使得目标用户的身份隐藏在所有用户之中. 相比于CDP模型,SDP模型进行了本地扰动,对用户数据的保护更强;相比于LDP模型,$\mathcal{S}$通过输出随机排列的不确定性避免了对目标用户身份重新甄别.

表1　中心化/本地化/混洗差分隐私模型的异同点

模型名称	信任度假设	实现机制	模型优点	模型缺点	相同点	主要应用
CDP	完全信任收集者	拉普拉斯机制[1]、高斯机制[7]、指数机制[20]等	查询与分析的精度较高	用户无法掌控自己的原始数据	3种模型均满足差分隐私的序列组合性质,以及后置处理性质等	统计查询[23]与SGD模型[24]等
LDP	完全不信任收集者	WRR[22]、GRR[5]、Lap[1]、OLH[13]、HRR[16]等	数据安全性较高	查询与分析精度比较低		频率/均值/分布估计[25]/聚类分析[26]
SDP	由不信任向信任偏移	Blanket[4]、PBS[27]、ICEA[28]、PAFAM[29]等	能够弥补CDP与LDP存在的不足,实现隐私放大	需要假设混洗方可信,否则需要引入数据加密技术		聚集查询/均值/分布估计[30~33]

3　混洗差分隐私的攻击类型

在SDP框架中,给定协议$\mathcal{P}=(\mathcal{R},\mathcal{S},\mathcal{A})$,令本地扰动消耗的隐私预算为$\varepsilon_l$,即$\mathcal{R}$满足$(\varepsilon_l,\delta)$-LDP. 混洗方$\mathcal{S}$的输出结果满足$(\varepsilon_c,\delta)$-CDP. 假定数据集$D$中第$n$个用户$u_n$为被攻击的目标用户. 根据图2中SDP的3种模型,探讨对u_n不同的攻击类型,具体描述如表2所示.

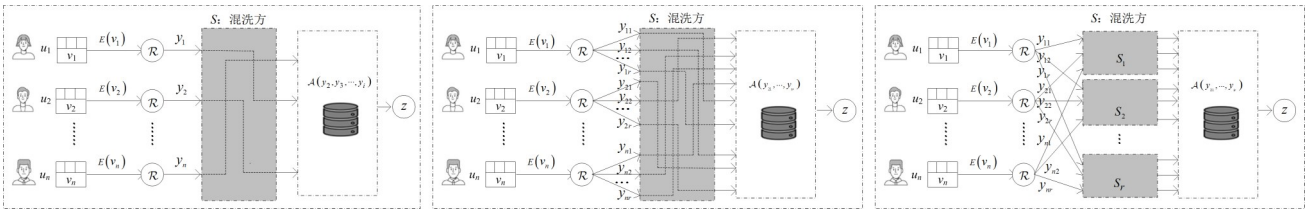
(1)假设收集者$\mathcal{A}$是攻击者,混洗方$\mathcal{S}$可信,n个用户诚实. $\mathcal{A}$接收的消息是来自$\mathcal{S}$随机排列的结果. $\mathcal{S}$切断了n个用户与他们扰动值之间的连接关系. $\mathcal{A}$无法根据扰动值连接推理出某个用户的身份. 目标u_n的数据满足协议$\mathcal{P}$提供的隐私保护程度ε_c,实现了隐私放大.

(2)假设收集者$\mathcal{A}$是攻击者,$\mathcal{A}$与n个用户中部分恶意用户共谋,混洗方$\mathcal{S}$可信. $\mathcal{S}$无法把目标u_n的数据隐藏在$n-1$个数值中,进而无法实现混洗方$\mathcal{S}$所能达到的隐私放大效果. 对u_n的数据保护仅满足(ε_l,δ)-LDP.

(3)假设收集者$\mathcal{A}$是攻击者,混洗方$\mathcal{S}$不可信且与$\mathcal{A}$共谋,n个用户诚实. 此时,所有用户数据的保护程度从(ε_c,δ)-CDP退化到(ε_l,δ)-LDP. 因此,目标用户u_n的数据仅受到(ε_l,δ)-LDP的保护. 假设有多个混洗方[如图2(c)],收集者$\mathcal{A}$与多个混洗方合谋,如果其中一个混洗方诚实的执行混洗操作,混洗后的消息向量就能够满足(ε_c,δ)-CDP.

(4)在通信过程中存在攻击情况下,即用户在把数据从本地扰动后发送给收集者$\mathcal{A}$的过程中被攻击. 该通信过程包括本地端到混洗方$\mathcal{S}$、混洗方$\mathcal{S}$到收集者$\mathcal{A}$两个过程. 由于每个用户均利用本地扰动机制$\mathcal{R}$保护了自身数据,则每个扰动结果均满足(ε_l,δ)-LDP. 为了防止通信过程中的攻击,通常利用加密技术保护消息从本地端到混洗方$\mathcal{S}$、混洗方$\mathcal{S}$到收集者$\mathcal{A}$的传输过程. 例如,安全多方计算[34]、秘密共享[35]、同态加密[36]等技术配合差分隐私完成聚集查询.

以上攻击类型均假定每个用户均遵循本地保护协议. 然而,实际应用中会存在部分用户不遵循协议的情况,这类用户称为恶意用户. 文献[37~39]考虑到模型的鲁棒性,即是在SDP模型下,寻找最多可容忍多少恶意用户,也能达到所有用户都遵循协议时的模型精度. 此类情况下,以文献[40]所提出协议所能达到的精度作为基准,文献[34]得出结论:当恶意用户在总体用户中占比小于1/3时,保护协议仍能达到文献[40]的精度,即能够满足中心化ε_c的隐私保护水平.



(a) 单消息单混洗框架 (b) 多消息单混洗框架 (c) 多消息多混洗框架

图2 混洗差分隐私的保护框架

表2 洗差分隐私下常见攻击类型

攻击类型			数据的隐私保护程度
类型一	收集者是攻击者,对目标用户身份重甄别[34]		混洗后的隐私保护程度为(ε_c,δ)-CDP
类型二	收集者与部分恶意用户合谋[35,36]		混洗后的隐私保护程度退化为(ε_l,δ)-LDP
类型三	单个混洗方情况下[35, 36]	收集者和混洗方合谋	混洗后的隐私保护程度退化为(ε_l,δ)-LDP
	多个混洗方情况下[35, 36]	收集者和大于1个混洗方合谋.假设最多有1个混洗方诚实	混洗后的隐私保护程度为(ε_c,δ)-CDP
类型四	通信过程存在攻击[35,36]		混洗后的隐私保护程度退化为(ε_l,δ)-LDP

4 混洗差分隐私保护框架与研究方向

图1对通用的CDP/LDP/SDP模型框架给予描述.而SDP模型根据用户发送消息的多少以及混洗方的多少来划分不同的保护框架.基于这些保护框架产生了系列研究方向.

4.1 混洗差分隐私的保护框架

目前基于混洗差分隐私的保护框架通常有3种具体形式:单消息单混洗框架、多消息单混洗框架,以及多消息多混洗框架,具体结构如图2所示.图2(a)描述了SDP模型的单消息单混洗框架.该框架中,每个用户u_i首先利用$E(\cdot)$对数值v_i进行编码,然后利用$\mathcal{R}$本地扰动$E(v_i)$(如式(13)所示),本地生成消息$y_i=\mathcal{R}(E(v_i))$.最后u_i把y_i发送给混洗方$\mathcal{S}$.$\mathcal{S}$利用随机排列函数π对n个消息形成的向量$\boldsymbol{M}=(y_1,y_2,\cdots,y_n)$进行随机排列,再把混洗结果$\pi(\boldsymbol{M})$发送给收集者$\mathcal{A}$进行统计分析,最后$\mathcal{A}$给出响应结果$Z$.

图2(a)保护框架的主要挑战是如何设计保护协议$\mathcal{P}=(\mathcal{R},\mathcal{S},\mathcal{A})$,以及高效的混洗协议$\mathcal{S}$.所设计的协议不但能够满足每个用户的本地隐私需求,即是满足LDP,也能够为收集者$\mathcal{A}$提供接近CDP模型下的查询与分析精度.文献[41~44]指出混洗协议$\mathcal{S}$的主要作用是切断用户与其所发送消息之间的连接,防止目标用户身份遭到重甄别.目前大多数混洗协议均采用Fisher-Yates[45]传统随机排列技术实现消息向量的排列.该技术一次随机排列的时间复杂度为$O(n^2)$.尽管该技术输出每种排列的概率均等于$(1/n!)$,当n很大时,相应的时间复杂度较高.如何在保证混洗效率的前提下设计高效的混洗方法至关重要.

尽管图2(a)中框架能够提高查询精度以及防止用户身份重甄别,然而文献[46]发现,无论该框架把LDP下哪种扰动机制(例如,TreeHist[47]、GRR[5]、HRP[48,49])线性分解成混洗本地扰动机制,其相应的误差依然是$\Omega\left(\min(\sqrt[4]{n},\sqrt{d})\right)$.根据文献[27~29,40,46]可知,当每个用户发送多个消息时,查询与分析的误差明显减少.图2(b)描述了多消息单混洗框架.该框架中用户u_i将其值v_i编码后生成$E(v_i)$.与图2(a)中单消息不同,u_i利用扰动机制$\mathcal{R}$生成多个消息,即$(y_{i1},y_{i2},\cdots,y_{ir})=\mathcal{R}\left(E(v_i)\right)$.每个用户把自己的消息向量发送给混洗方$\mathcal{S}$,$\mathcal{S}$随机排列所有的消息向量后发送给收集者$\mathcal{A}$,$\mathcal{A}$结合相应的查询需求响应最终结果$Z$.然而,该框架中发送消息的多少直接导致通信代价与精度之间的矛盾.发送的消息越多则精度越高,而不可避免地会导致较高的通信代价.为了实现查询精度与通信代价之间的均衡,在响应某特定查询与分析时,每个用户发送多少个消息,每个消息占多少字节是该框架研究的重点.

图2(a)与图2(b)中的模型均假设只存在单个混洗方$\mathcal{S}$,如果$\mathcal{S}$与收集者$\mathcal{A}$合谋、或者某些攻击者攻破了$\mathcal{S}$,则目标用户的身份依然会被甄别.文献[50~55]描述了r个混洗方并行混洗的框架,如图2(c)所示.在多消息多混洗框架中,给出r个并行混洗方$\{\mathcal{S}_1,\mathcal{S}_2,\cdots,\mathcal{S}_r\}$,每个混洗方独立地对$n$个用户的消息进行混洗.每个用户$u_i$产生$r$个消息$(y_{i1},y_{i2},\cdots,y_{ir})=\mathcal{R}(E(v_i))$,然后把相应的消息发送相应的混洗方,例如,$y_{ir}$发送给$\mathcal{S}_r$.这样布置的好处是直接降低混洗方$\mathcal{S}$与收集者$\mathcal{A}$合谋的概率.在图2($c$)的框架中,主要考虑如何降低计算复杂度、通信代价以及如何利用IKOS[56](Inference Kernel for Open Static analyzers)协议提升安全聚集查询的精度.

4.2 混洗差分隐私的研究方向

单消SDP模型起着CDP/LDP模型之间的桥梁作用,在理论与实际应用方面具有重要价值.类似于CDP/LDP模型,SDP模型首先应用于简单的二进制聚集

查询、实数域聚集查询,然后扩展到直方图查询、频率估计、机器学习以及联邦学习领域. 目前SDP模型的主要研究方向如表3所示.

表3　混洗差分隐私保护的主要研究方向

研究方向		代表方法
统计查询	二进制聚集查询	文献[27,34,40,54,55,57]
	实数聚集查询	文献[4,27~29,50,51]
	直方图查询	文献[4,27,35,44,47,55,57]
	频率估计	文献[4,27,35,55,57]
统计分析	机器学习	文献[43,53]
	联邦学习	文献[58]
隐私放大		文献[4,27,41]

从表3中可SDP研究方向包括统计查询问题、隐私放大以及机器学习和联邦学习问题的研究. 这些研究方向均源自图2三种保护框架. 第5节、第6节与第7节结合上述研究方向,对目前国内外的研究现状进行阐述分析.

5　混洗差分隐私下的统计查询

结合图2中的3个SDP框架,本节详细分析支持统计查询的相关方法,并从防御攻击类型、查询误差、通信代价等几个方面对比相应方法的性能. 以下5.1、5.2节详细阐述了SDP模型下的聚集查询、直方图查询、频率估计等.

5.1　基于混洗差分隐私的聚集查询

SDP模型的聚集查询通常分为二进制和实数聚集查询. 常用的操作包括Sum、Count、Avg等. 例如,结合二进制数据查询1出现的次数.

5.1.1　基于二进制数据的聚集查询

假设每个用户u_i拥有值$v_i\in\{0,1\}$. 给定聚集查询函数$f(v_1,v_2,\cdots,v_n)$,则二进制聚集查询通常可以表示为$f(v_1,v_2,\cdots,v_n)=\sum v_i$. PBS[27]是单消息单混洗框架(如图2(*a*)所示)的早期代表,利用"先扰动-后混洗"协议避免了表2中攻击类型一. 该方法首先利用直接编码技术把v_i本地编码成二进制数据,即$v_i=E(v_i)$. 再根据式(13)重写WRR[22]机制的输出概率,如式(18)所示:

$$\forall_{y\in\{0,1\}}\Pr\left[\mathcal{R}\left(E\left(v_i\right)\right)=y\right]=\left(1-\gamma\right)I_{\left\{y=E\left(v_i\right)\right\}}+\frac{1}{2}\gamma \quad (18)$$

其中,$\gamma=2/e^{\varepsilon}+1$. I为标识函数,若$y=E(v_i)$,则$I=1$.

根据式(18)可知,PBS是把WRR机制分解成伯努利分布与均匀分布来本地扰动二进制数据. n个用户中有部分用户利用伯努利分布Ber(1/2)产生随机噪声,剩余的用户发送真实数据. 混洗方$\mathcal{S}$把n个消息$(y_1,y_2,\cdots,y_n)$随机混洗后发送给$\mathcal{A}$,收集者$\mathcal{A}$修正噪声聚集结果$(\sum y_i-n\gamma/2)/(1-\gamma)$,且无偏估计误差为$O\left(\sqrt{\log(1/\delta)}/\varepsilon\right)$. 该误差明显低于LDP模型下的聚集误差$O\left(\sqrt{n}/\varepsilon\right)$. 并且PBS方法能够保证当$\varepsilon_l\sim\ln(n)$时,达到$\varepsilon_c=1$的隐私放大效果.

为了进一步减少聚集查询误差,ZSUM[40,57]方法利用图2(*b*)模型实现了多消息单混洗协议下的二进制聚集查询. 每个用户u_i产生消息向量$\boldsymbol{v}_i+\boldsymbol{z}_i$,其中$\boldsymbol{z}_i\sim\mathrm{Ber}\left(1-50\ln(2/\delta)/\varepsilon^2 n\right)$. 混洗方接收到所有消息向量后进行均匀随机排列. 最终修正后二进制聚集查询可表示为$Z=Z^*-p$,其中,$p=1-50\ln(2/\delta)/\varepsilon^2 n$,$Z^*=\left(\sum\boldsymbol{v}_i+\boldsymbol{z}_i\right)/n$,$\sum\boldsymbol{z}_i$符合$\mathrm{Bin}\left(n,1-50\ln(2/\delta)/\varepsilon^2 n\right)$二项分布. 由于二项分布的有界性,若$\sum\boldsymbol{v}_i=0$,则$\sum\boldsymbol{v}_i+\mathrm{Bin}(n,p)=0$. 相比于PBS的直接编码,ZSUM方法利用One-Hot技术编码用户数据,并以概率$1-50\ln(2/\delta)/\varepsilon^2 n$生成$\boldsymbol{v}_i+\boldsymbol{z}_i$. 然而,这种生成$\boldsymbol{v}_i+\boldsymbol{z}_i$的方式并不满足$(\varepsilon,\delta)$-LDP. 与PBS相同,ZSUM方法同样对所有用户的消息进行均匀混洗. 而在聚集查询处理时,该方法相当于在最终求和结果之上添加满足二项分布的噪声. 相比于PBS的聚集误差,ZSUM方法的误差为$O\left(\log(1/\delta)/n\varepsilon^2\right)$.

尽管PBS与ZSUM处理二进制聚集查询取得精度高于LDP模型下的方法,但是二者仍存在两点不足:PBS的误差下界不够严谨;ZSUM的通信代价较高. 基于二者的不足,DDM[55]方法通过结合一元编码与关联噪声策略分解负二项分布$\mathrm{NBin}(1,e^{-\varepsilon})$来产生$n$个独立同分布噪声,每个用户利用该噪声本地扰动自身数据,其中$\mathrm{NBin}(1,e^{-\varepsilon})$具有无穷可分分布性质. 收集者$\mathcal{A}$聚集混洗方$\mathcal{S}$发送的消息后响应二进制聚集查询. 从$\mathcal{A}$的角度来看,修正聚集结果可表示为$\sum v_i+\mathcal{F}$,其中$\mathcal{F}$服从NBin分布. 不同于PBS的WRR扰动机制和ZSUM以概率生成噪声机制,DDM采用$\mathrm{NBin}(1/n,e^{-\varepsilon})$扰动编码数据. 3种方法的混洗策略相同,均采用随机排列消息. 此外,相比于PBS,ZSUM与DDM均采用多消息策略减少聚集查询的误差. 与ZSUM相比,DDM以较小的通信代价采用无穷可分分布NBin获得了接近离散拉普拉斯机制产生的噪声误差$O(1/\varepsilon)$. 其中离散拉普拉斯机制可以表示为$\mathcal{P}(D)=\mathcal{A}(D)+z$,$z$的概率密度函数见式(19):

$$\Pr[z]=\frac{1}{\sum\limits_{z=-\infty}^{+\infty}\exp\left(-\dfrac{\varepsilon|z|}{\Delta\mathcal{A}_1}\right)}\cdot\exp\left(-\frac{\varepsilon|z|}{\Delta\mathcal{A}_1}\right) \quad (19)$$

PBS、ZSUM与DDM方法均是从静态数据的角度处理聚集查询. PSYM[34]方法结合图2(*b*)框架利用对称几何噪声实现了数据流上的二进制聚集查询. 每个用户利用式(20)扰动v_i值,收集者$\mathcal{A}$结合$\mathcal{S}$随机混洗后的消息可计算出修正后的聚集结果,如式(21)所示.

$$\forall_{y\in\{0,1\}}\Pr\left[\mathcal{R}\left(E\left(v_i\right)\right)=y\right]=\left(1-\gamma\right)I_{\left\{y=E\left(v_i\right)\right\}}+\gamma\mathrm{SG}(\varepsilon) \quad (20)$$

其中，SG(ε)表示对称几何分布，$\gamma\approx 1/\varepsilon^2 n$. I为标识函数，若$y=E(v_i)$，则$I=1$.

$$Z=\frac{1}{2}\left(\sum_{i=1}^{n}y_i+n\right)=\frac{1}{2}\left(\sum_{i=1}^{n}v_i+\sum_{i=1}^{N}\eta_i+n\right) \tag{21}$$

其中，$|N|\sim \mathrm{Bin}(n,p)$，$p=2(\mathrm{e}^{\varepsilon}+1)/n(\mathrm{e}^{\varepsilon}-1)^2$，$\eta_i\sim \mathrm{SG}(\varepsilon)$.

以上方法均是从近似混洗差分隐私来响应二进制聚集查询，即(ε,δ)-SDP中$\delta>0$. 文献[54]从纯SDP的角度响应二进制聚集查询，即$\delta=0$时，满足ε-SDP. PDP>1方法[54]结合图2(b)框架利用离散拉普拉斯机制实现混洗本地扰动，且混洗结果满足纯SDP. PDP>1方法本地扰动策略如式(22)所示：

$$\forall_{y\in\{0,1\}}Pr\left[\mathcal{R}\left(v_i\right)=y\right]=\left(1-\gamma\right)l\left(\frac{d-1}{2}+v_i\right)+\gamma\mathrm{DLap}\left(\frac{d}{2},s\right) \tag{22}$$

其中，$l\left(v_i+(d-1)/2\right)$表示值$v_i+(d-1)/2$以概率为1的分布；$\mathrm{DLap}(d/2,s)$表示离散拉普拉斯分布，$d$表示用户发送的消息个数，$s$表示分布规模.

表4分析了SDP下二进制聚集查询的主要方法. ε-CDP模型下的二进制聚集查询误差为$O(1/\varepsilon)$. ε-LDP模型下的聚集误差为$O(\sqrt{n}/\varepsilon)$. 从(ε,δ)-SDP模型的角度来看，表4中相应方法的误差均在$O(1/\varepsilon)$与$O(\sqrt{n}/\varepsilon)$之间. 其中有些方法以牺牲通信代价来减少误差. 这些方法取得较为合理误差的主要原因是SDP模型下用户对收集者的信任假设性比CDP合理，查询误差比LDP模型的低. 然而，这些方法如何拓展到高维的二进制数据是个挑战性问题.

5.1.2 基于[0,1]实数的聚集查询

假设每个用户u_i拥有值$v_i\in[0,1]$，相应聚集查询可表示为$\sum v_i$. 在[0,1]实数值域中，(ε,δ)-CDP模型取得的查询误差为$O(1/\varepsilon)$[1]，而(ε,δ)-LDP下的查询误差为$\Theta(\sqrt{n}/\varepsilon)$[3]. PRS[27]是结合图2($b$)模型实现[0,1]值域内聚集查询的早期代表. 该方法利用随机凑整编码策略，对用户值$v_i\in[0,1]$随机凑整成二进制值$b_i\in\{0,1\}$. 为了减少随机凑整误差，需要对v_i进行r轮凑整操作，记为$(b_1,b_2,\cdots,b_r)\in\{0,1\}^r$，则$\mathrm{E}(\sum b_i/r)=v_i$成立. 而对于$(b_1,b_2,\cdots,b_r)$中每个二进制数，采用式(18)对其进行本地扰动. 混洗方$\mathcal{S}$收到每个用户的r条消息后，对$n\times r$条消息进行随机混洗. 尽管收集者能够获得误差为$O\left(\log(n/\delta)/\varepsilon\right)$的查询结果，但是却需要每个用户发送$O(\varepsilon\sqrt{n})$比特的消息，相应的通信代价较高.

表4 SDP下二进制聚集查询主要方法对比

方法名称	所属框架	攻击类型	优点	缺点	通信代价	误差
PBS[27]	单消息单混洗框架	一	直接编码, WRR扰动	仅支持二进制聚集查询	1	$O\left(\sqrt{\log(1/\delta)}/\varepsilon\right)$
ZSUM[40]	多消息单混洗框架	一	若真实结果是0,则估计结果也是0	不满足LDP	2	$O\left(\log(1/\delta)/n\varepsilon^2\right)$
PSYM[34]	多消息单混洗框架	一、二	利用对称几何噪声本地处理数据	通信代价高	$O\left(\log(1/\delta)/\varepsilon^2\right)$	$O\left(\sqrt{\log(1/\delta)}/\varepsilon\right)$
PDP>1[54]	多消息单混洗框架	一	实现了纯SDP,即$\delta=0$	通信代价较高	$O\left(\log n/\varepsilon\right)$	$O\left(\sqrt{\log(1/\delta)}/\varepsilon^{3/2}\right)$
DDM[55]	多消息单混洗框架	一	查询精度与CDP下Lap机制相同	单消息误差边界不清楚	$1+O\left(\log^2(1/\delta)/\gamma n\varepsilon^2\right)$	$O(1/\varepsilon)$

Blanket[4]采用图2(a)混洗模型响应[0,1]值域上的聚集查询. 不同于PRS的随机凑整编码策略，Blanket利用定点编码方式，将值$v_i\in[0,1]$离散化成$\{0,1,\cdots,b\}$中某个值. 具体编码方式是利用$b+1$个桶对v_i进行随机凑整，如式(23)所示：

$$E\left(v_i\right)=\lfloor v_i\times b\rfloor+\mathrm{Ber}\left(v_i\times b-\lfloor v_i\times b\rfloor\right) \tag{23}$$

其中，Ber($\cdot$)表示伯努利分布.

相比于PRS的r轮随机凑整编码，Blanket的定点编码策略仅需要1次编码即可，其通信代价为1比特. 结合编码后的值，利用式(14)对其本地扰动. 与PRS的混洗策略相同，Blanket对n个用户报告的单消息进行随机混洗，收集者$\mathcal{A}$结合n个消息响应最终的聚集查询. 相比于Blanket，若PRS采用单消息策略，即$r=1$，则其聚集误差为$O(\sqrt{n}/\varepsilon)$. 而Blanket的误差仅为$\Theta(n^{1/3}/\varepsilon)$. 尽管Blanket的误差与通信代价较低，较小的离散化粒度(即式(23)中的b值较小)会导致较高的误差. 不同于Blanket，ICEA[28]基于图2(b)中框架利用隐形衣(Invisibility Cloak)编码方式把v_i值隐藏在m个离散值中，即$E_{N,k,m}(v_i)=\{y_1,y_2,\cdots,y_m\}$. 其中$\{y_1,y_2,\cdots,y_m\}$的前$m-1$个值是随机值，而第$m$个值是修正值，如式(24)所示：

$$y_m=\left(\lfloor v_i\times k\rfloor-\sum_{j=1}^{m-1}y_j\right)\bmod N \tag{24}$$

其中，k，N为整数.

PRS的凑整编码和Blanket的定点编码是为了初步转换用户数据，ICEA利用隐形衣编码生成了零和噪声，而零和噪声效果能够实现噪声相互抵消. 这种噪声相互抵消的特质使得用户可以添加更多的噪声来保护自身数据. Blanket的单消息操作无法实现全部噪声相互抵消，只能采用较大的离散化参数b来产生较少的噪声. 此外，ICEA无需离散化操作，并且以多消息的形式

进行随机混洗,其聚合精度高于Blanket与PRS. 尽管ICEA的聚集查询误差较低,然而该方法没有讨论如何设置与限制m值,且通信代价较高.

不同于PRS、ICEA与Blanket的单混洗方模型,RECP[50,51]基于图2(c)模型把每个用户的v_i值分解成m个消息$(y_1,y_2,\cdots,y_m)$后分别发送给对应的m个并行混洗方. 与Blanket的定点编码相似,RECP把v_i值进行本地序列性定点编码,编码方式如式(25)所示. 而与Blanket不同的是,RECP需要制定m个编码精度$\{b_1,b_2,\cdots,b_m\}$.

$$v_i \approx s_1 q_1^{-1} + s_2 q_2^{-2} + \cdots + s_m q_m^{-m} \tag{25}$$

其中,$q_j=\prod_{l=1}^{j} b_l$,$s_j=\left\lfloor q_j v_i - b_j \left\lfloor q_{j-1} v_i \right\rfloor \right\rfloor$,$q_0=0$.

利用随机凑整策略对第m个消息s_m进行本地处理,其目的是对原始值v_i进行无偏估计. 对于剩余的$1,2,\cdots,m-1$个消息,结合式(26)对其进行本地扰动.

$$\begin{aligned}&\forall_{y\in[1..s_{m-1}]}\Pr\left[\mathcal{R}\left(s_i\right)=y\right]\\&=\left(1-\gamma_i\right)I_{\{y=s_i\}}+\gamma_i\Pr\left[\mathrm{Unif}([1..s_{m-1}])=y\right]\end{aligned} \tag{26}$$

其中,$\Pr[\mathrm{Unif}[1,\cdots,s_{m-1}]=y]=1/m-1$,$I$为标识函数,若$y=E(v_i)$,则$I=1$. $\gamma_i=O(b_i/n)$.

相比于PRS、ICEA与Blanket,RECP以多消息的形式采用m个混洗方随机混洗n个用户发送的消息. 其安全性高于PRS、ICEA与Blanket,原因是PRS、ICEA与Blanket采用单混洗模型,如果混洗方与收集者共谋,则该类方法退化成(ε,δ)-LDP. 而RECP可以允许$m-1$个混洗方与收集者共谋. 为了提高聚集查询精度,RECP采用差分隐私强组合性质[7]来均衡定点编码造成的误差与扰动造成的误差,在误差为$O\left(\log\left(\log n\sqrt{\log(1/\delta)}\right)/\varepsilon\right)$时,获取最优$m=O\left(\log(\log n)\right)$. 与PRS、ICEA、Blanket类似,RECP的误差受到n与δ的影响. 不同于RECP,CESS[50]结合图2(c)混洗模型把$[0,1]$上的聚合查询转换成IKOS[56]协议下有限集合的安全加法问题. 该方法对每个用户的v_i值定点编码后,添加由Polya分布函数产生的噪声,如式(27)所示. 其中IKOS协议是把任意v_i、v_j分解为m条秘密共享$\{y_{i1},y_{i2},\cdots,y_{im}\}$与$\{y_{j1},y_{j2},\cdots,y_{jm}\}$,且$\sum_{j\in[m]}y_{ij}=\sum_{i\in[m]}y_{ji}$,二者的分布距离满足不等式:$\mathrm{Dis}\left(\{y_{i1},y_{i2},\cdots,y_{im}\},\{y_{j1},y_{j2},\cdots,y_{jm}\}\right)\leqslant 2^{-\Omega(\sigma)}$,其中$\sigma$是安全因子. 针对噪声值$y_i$,CESS利用IKOS将其分解成$m$份可加的密码共享$\{y_{i1},y_{i2},\cdots,y_{im}\}$,然后再分别发送对应的$m$个并行混洗方. 每个混洗方随机排列$n$条消息后发送给收集者,收集者结合$nm$条消息与IKOS的安全加法特点来响应聚集查询.

$$\begin{aligned}y_i=&\left\lfloor v_i\times b\right\rfloor+\mathrm{Ber}\left(v_i\times b-\left\lfloor v_i\times b\right\rfloor\right)\\&+\mathrm{Polya}\left(1/n,\lambda\right)-\mathrm{Polya}\left(1/n,\lambda\right)\end{aligned} \tag{27}$$

其中,b为定点编码粒度,Ber(·)表示伯努利分布,Polya(·)表示Polya分布,λ表示噪声尺度.

相比于PRS、ICEA、Blanket与ICEA,CESS的误差不再受n与δ的影响,其原因是该方法利用Polya分布模拟离散拉普拉斯分布来添加噪声,误差达到(ε,δ)-CDP模型下的离散拉普拉斯误差$O(1/\varepsilon)$. 然而该方法由于没有较好地控制用户与混洗方之间通信消息大小,会导致较高的通信代价.

类似于CESS,PAFAM[29]基于图2(b)框架非交互式地实现了$[0,1]$域上的聚集查询. 该方法把v_i离散化到某个离散区间$\{0,1,\cdots,b\}$,然后再利用IKOS安全协议把离散后的v_i值分割成m个消息,即$SM(v_i)=\{y_1,y_2,\cdots,y_m\}$. 其中$\{y_1,y_2,\cdots,y_{m-1}\}$源自区间$\{0,1,\cdots,b\}$中的均匀抽样,而$y_m=\mathrm{SM}(v_i)-\sum_{j=1}^{m-1}y_j$. 与CESS相比,PAFAM以$m=O(\log n)$的通信代价来控制用户与混洗方之间的消息交互. 此外,该方法以IKOS协议的σ-安全性实现了与CESS相同的隐私保护程度和误差,即$O(1/\varepsilon)$.

尽管CESS与PAFAM以较小的通信代价取得了$O(1/\varepsilon)$误差,然而这些方法依然存在几点不足:每个用户至少发送3个消息才能达到$O(1/\varepsilon)$误差;通信代价随着$\log(1/\delta)/\log n$递增. 为了弥补这些方法的不足,Δ-SUM[59]在DDM[55]的基础上仅要求每个用户发送$1+O(1)$个消息即可达到$O(1/\varepsilon)$误差. 该方法利用式(28)把用户的$v_i\in[0,1]$值随机离散化成y_i,且$y_i\in\{0,1,\cdots,\Delta\}(\Delta>1)$. 每个用户结合自己的离散值$y_i$,通过$\Delta$-SUM方法产生联合噪声对$y_i$进行本地扰动.

$$y_i=\begin{cases}\left\lfloor v_i\Delta\right\rfloor, & \text{以概率}1-\left(v_i-\left\lfloor v_i\Delta\right\rfloor\right)\\\left\lfloor v_i\Delta\right\rfloor+1, & \text{以概率}v_i-\left\lfloor v_i\Delta\right\rfloor\end{cases} \tag{28}$$

联合噪声包括两部分:无限可分分布D_{central}产生中心噪声、D^s分布产生零-和噪声,即D^s产生的噪声能够相互抵消. 其中,D_{central}产生噪声是为了使其结果逼近离散拉普拉斯机制产生的噪声结果,即$D_{\mathrm{central}}-D_{\mathrm{central}}\sim\mathrm{DLap}(\varepsilon/\Delta)$,$\mathrm{DLap}(\varepsilon/\Delta)$表示离散拉普拉斯分布. 尽管$\Delta$-SUM的通信代价优于同类方法,然而如何设计合理的Δ值是个挑战性问题,过大的Δ值导致计算复杂性过高,同时可能导致最终的聚集结果精度过低;而过小的Δ值导致实数v_i的离散粒度过小.

表5综合分析了SDP下$[0,1]$值域内实数聚集查询的主要方法. 类似于SDP下的二进制聚集查询,这些方法在$[0,1]$内聚集查询结果的误差同样在$O(1/\varepsilon)$与$\Theta(\sqrt{n}/\varepsilon)$之间. 其中,CESS、PAFAM以及IKOSP方法均

表5 SDP下[0,1]内实数聚集查询主要方法对比

方法名称	所属框架	攻击类型	优点	缺点	通信代价	误差
PRS[27]	多消息单混洗框架	一	一元编码, WRR扰动	通信代价高	$O(\varepsilon\sqrt{n})$	$O\left(\log(n/\delta)/\varepsilon\right)$
Blanket[4]	单消息单混洗框架	一	定点编码, GRR扰动, 隐私放大	细粒度离散化导致查询误差高	1	$\Theta(n^{1/3}/\varepsilon)$
ICEA[28]	多消息单混洗框架	一、二	用户添加的噪声能够实现零和	通信代价较高	$O\left(\log(n/\varepsilon\delta)\right)$	$O\left(\sqrt{\log(1/\delta)}/\varepsilon\right)$
RECP[50,51]	多消息多混洗框架	一、三	定点编码, GRR扰动	多次交换导致计算复杂度高	$O\left(\log(\log n)\right)$	$O\left(\log\left(\log n\sqrt{\log(1/\delta)}\right)/\varepsilon\right)$
CESS[50]	多消息多混洗框架	一、四	IKOS机制,模拟Laplace机制	通信代价较高	$O\left(\log(n/\delta)\right)$	$O(1/\varepsilon)$
PAFAM[29]	多消息单混洗框架	一、四	IKOS机制,发送消息个数为常数	计算复杂度高	$O\left(1+\log(1/\delta)/(\log n)\right)$	$O(1/\varepsilon)$
Δ-SUM[59]	多消息单混洗框架	一、二	能够实现零和噪声	过大的Δ导致计算复杂度高	$O\left(1+\log(1/\delta)/\varepsilon\sqrt{n}\right)$	$O(1/\varepsilon)$

是采用IKOS安全协议把用户的实数值分割成多个消息进行聚集响应. 虽然这些方法的误差已接近于(ε,δ)-CDP模型下的拉普拉斯误差,然而,分割粒度的选择与计算复杂性是这类方法的不足. 类似于CESS, Δ-SUM能够实现零-和机制实现部分噪声之和为零,然而如何选择合适Δ值很困难. 此外,表5中的方法如何拓展到高维$[0,1]$值域上的聚集操作是个挑战性问题.

5.2 基于混洗差分隐私的直方图查询

直方图能够准确地获取数据分布的概要. 该技术按照一定粒度把数据离散化为不相交的桶,每个桶通过频率近似描述整体数据的统计信息(例如,Heavy Hitter[47]、频率查询[48]以及分布估计[49]等). 给定有限集合$[d]=\{1,2,\cdots,d\}$、n个用户$\{u_1,u_2,\cdots,u_n\}$以及收集者$\mathcal{A}$. 在图2三种混洗差分隐私模型保护下,收集者$\mathcal{A}$通过构造每个桶的频率或者计数$\left(\text{Count}(v_i)=\left|\{u_j|v_j=v_i,\text{且 }v_i,v_j\in[d]\}\right|\right)$来响应直方图查询. R^{PH}[4]是图2(a)模型下响应直方图查询的典型代表方法. 其利用GRR机制[5]的线性分解形式(如式(14)所示)对用户的$v_i\left(v_i\in[d]\right)$进行本地扰动,混洗方$\mathcal{S}$随机混洗所有消息之后发送给收集者$\mathcal{A}$. R^{PH}[4]与Blanket[4]类似,同样是利用均匀分布的噪声数据掩盖目标用户的真实值. 然而,该方法要求每个用户发送$O(\log d)$比特的消息才能完成直方图查询. 根据式(14)可知,值域d越大,GRR机制的扰动误差越大,通信代价越高. 相比于R^{PH}[4]的定点编码, BitSum[27]利用One-Hot编码把用户的v_i值编码成长度为d的二进制串. 定点编码与One-Hot的主要区别是前者需要设置合适的离散化粒度,而后者直接根据值域d进行编码. BitSum通过执行d次按位扰动方法PBS[27]来本地处理编码后的二进制串. 与R^{PH}的混洗方式相同, BitSum对所有用户扰动后的二进制串进行随机混洗,收集者结合混洗结果重构直方图并响应聚集查询,其响应误差低于R^{PH},原因是R^{PH}的采用单消息模型,而BitSum采用图2(b)的多消息模型. 尽管BitSum直方图查询精度高于R^{PH},然而该方法的误差与通信代价依旧随着值域d的增加而增加,为了达到$O\left(\sqrt{\log d\log(1/\delta)}/n\varepsilon\right)$精度,每个用户发送$O(d)$比特的消息.

为了减少R^{PH}与BitSum的通信代价与误差,文献[46,60]结合图2(b)模型分别采用Hadamard机制与Count Min机制设计了Private-coin[60]与Public-coin[46]响应直方图查询. Private-coin结合混洗差分隐私的内在机理重写了Hadamard机制,记为$\mathcal{HR}$,如式(29)所示. 给定用户u_i的值$v_i\left(v_i\in[d]\right)$, u_i利用哑元填充技术把v_i增广成为长度为k的0/1串,即$\left\|v_i\right\|_1=k$. 利用$\mathcal{HR}$对于增广后的v_i中每个非零位索引j随机生成τ个索引$\left\langle a_{j,1},a_{j,2},\cdots,a_{j,\tau}\right\rangle$. 针对该索引向量,再随机生成$\rho$个假元组$\left\langle a_{g,1},a_{g,2},\cdots,a_{g,\rho}\right\rangle$且$1\leqslant g\leqslant\rho$. ρ个假元组起到了“隐私毯子(Blanket)”作用,来隐藏真实索引$\left\langle a_{j,1},a_{j,2},\cdots,a_{j,\tau}\right\rangle$. 混洗方$\mathcal{S}$接收到每个用户所发送的$1+\rho$个消息后,随机排列后发送给$\mathcal{A}$, $\mathcal{A}$生成满足(ε,δ)-CDP的直方图. ρ个假数据会导致直方图每个桶计数偏离其真实计数. 为此, $\mathcal{A}$需要对每个桶计数$\hat{x}_j$进行去偏处理,形如$\hat{x}_j\leftarrow\left(\hat{x}_j-(\rho+k)n2^{-\tau}\right)/1-2^{-\tau}$. 与$\text{R}^{\text{PH}}$、BitSum相比, Private-coin利用Hadamard矩阵压缩用户数据,进而减少了用户与混洗方之间的通信代价. 然而,由于Private-coin利用私有通道,其直方图查询响应时间受到用户个数n的影响,即响应单个直方图查询的时间花费为$\tilde{O}(n)$.

$$\forall_{j\in[2d]}\Pr\left[\mathcal{HR}(j)=\left\langle a_{j,1},a_{j,2},\cdots,a_{j,\tau}\right\rangle\right]=\frac{1}{1+\rho}I_{\left\{j=<a_{j,1},a_{j,2},\cdots,a_{j,\tau}>\right\}}+\frac{\rho}{1+\rho}\left[\Pr\left[\mathrm{Unif}(2d)\right]\right]\times\frac{1}{\mathrm{C}_{2d}^{\tau}}\tag{29}$$

其中,ρ表示假元个数且$\rho=O\left(\log(1/\delta)/\varepsilon^2\right)$,$I$为标识函数,若$j=a_{j,1},a_{j,2},\cdots,a_{j,\tau}$,则$I=1$.

不同于Private-coin,Public-coin方法结合图2(b)模型利用Count-Min结构[61]与WRR[22]机制响应直方图查询. 该方法利用哈希簇$\left\{h_t:[d]\to[g],\forall t\in[\rho]\right\}$把用户值$v_i$哈希到一个$[\rho]\times[g]$矩阵$\boldsymbol{M}$中,$\boldsymbol{M}$的每一行记为$v'_{it}$. 每个哈希函数最多产生$\rho$个假数据. 若哈希后的值为$h_t(v_i)\in[g]$,则在该行的$h_t(v_i)$处置1,其余置为0. 利用式(17)对$v'_{it}$的每位进行本地扰动,即以1-$\gamma$的概率选取$v'_{it}$中索引为$j$位置的值,以$\gamma$的概率选取其他值. 每个用户最多产生$g+g\times\rho$个消息给混洗方,其中$\rho=O(\log d)$. 收集者$\mathcal{A}$对每个桶进行去偏处理后获得相应的计数为$\hat{x}_j\leftarrow\max\left\{\min\left\{C[t,h_t[j]]-rn:t\in[\rho]\right\},0\right\}$. 与Private-coin的匿名通道相比,Public-coin使用了公共通道,响应查询时间为$O(\log d)$.

与Private-coin类似,pureDUMP[62]方法基于图2(b)模型利用假元组产生器生成符合$\mathrm{Bin}(\rho,1/d)$分布的ρ个假数据. 用户把v_i值与ρ个假数据一起报告给混洗方$\mathcal{S}$,其中$\rho=O\left(d\log(1/\delta)/n\varepsilon^2\right)$. 收集者$\mathcal{A}$对结合混洗方报告的消息,对每个桶的计数进行修正$\hat{x}_j\leftarrow(\hat{x}_j-n\rho/d)/n$. $\mathcal{A}$结合所有桶的修正值,以$O\left(d\log(1/\delta)/n\varepsilon^2\right)$的通信代价响应直方图查询. 与pureDUMP相比,Private-coin的ρ个假数据来自$\mathcal{HR}$机制的输出空间,而pureDUMP没有采用任何本地扰动机制,仅利用$\mathrm{Bin}(\rho,1/d)$产生ρ个假数据保护用户的真实值v_i. 因此,ρ值的大小直接决定着v_i的保护层度. 然而,pureDUMP没有给出如何选择合适的ρ值来均衡直方图的隐私性、精度与通信代价. 尽管pureDUMP采用了与Public-coin类似的公共通道技术共享假元组,然而,该方法由于没有利用Hadamard与Count Min结构压缩原始数据,造成通信代价较高.

与pureDUMP类似,mixDUMP[62]同样采用$\mathrm{Bin}(\rho,1/d)$分布生成ρ个假数据,再利用GRR[5]机制的线性分解形式(如式(14)所示)对用户值v_i进行扰动. 每个用户把扰动值与ρ个假数据一起报告给混洗方$\mathcal{S}$,其中$\rho=O\left(14d\log(4/\delta)/n\varepsilon^2\right)$. 收集者利用混洗方的消息对每个桶的计数进行修正,修正结果可以表示为$\hat{x}_j\leftarrow\left(\hat{x}_j-(n\lambda+n\rho)/d\right)/(1-\lambda)$. 相比于pureDUMP,mixDUMP利用假数据增强了GRR机制的保护强度. Private-coin同样利用假设数据增强$\mathcal{HR}$机制的隐私保护强度,而其通信代价却低于mixDUMP,其原因是Private-coin利用Hadamard结构压缩了原始数据. Private-coin、Public-coin、pureDUMP以及mixDUMP方法不允许部分用户共谋,每个用户均诚实地产生ρ个假数据. 不同上述这些方法,$\mathrm{P}_{\mathrm{FLIP}}$[44]采用One-Hot编码把用户的$v_i$值编码成长度为$d$的0/1串,同时允许$\rho$个共谋假用户生成长度为$d$的0串,且$\rho>132(\mathrm{e}^{\varepsilon}+1/\mathrm{e}^{\varepsilon}-1)^2\ln(4/\delta)/5n$. 利用WRR[22]机制分别对长度为$d$的0/1串与0串进行本地扰动. 为了避免$d$过大带来高通信代价,$\mathrm{P}_{\mathrm{FLIP}}$结合Count-Min编码结构中的哈希转换对编码后的0/1串进行压缩. 然后再利用WRR扰动压缩后的值. 尽管$\mathrm{P}_{\mathrm{FLIP}}$的通信代价与防共谋方面优于Private-coin、Public-coin与mixDUMP,但无法防止用户与混洗方$\mathcal{S}$共谋.

Private-coin、Public-coin以及$\mathrm{P}_{\mathrm{FLIP}}$分别依赖于Hadamard转换与哈希转换提高了直方图查询精度. 然而,这几种方法的查询误差依然没有脱离值域大小d的影响,d值过大会导致直方图查询误差过高. 基于此,$\mathrm{P}^{\mathrm{hist}}$[57]方法首先利用One-Hot机制对用户真值$v_i$进行本地编码,再利用ZSUM[40,57]方法对编码后的每一位b_j进行扰动,即$\boldsymbol{m}_j\leftarrow j\cdot\mathrm{ZSUM}_{(\varepsilon,\delta)}(b_j)$. 每个用户发给混洗方$\mathcal{S}$的向量是$\boldsymbol{y}_i=\left\langle\boldsymbol{m}_1,\boldsymbol{m}_2,\cdots,\boldsymbol{m}_d\right\rangle$. 收集者$\mathcal{A}$基于混洗后的向量集合$\left\langle\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n\right\rangle$即可响应直方图查询. 尽管$\mathrm{P}^{\mathrm{hist}}$方法的查询误差与值域$d$无关,然而该方法的通信代价却为$O(d)$. 此外,该方法不满足$(\varepsilon,\delta)$-LDP.

从BitSUM[27]到$\mathrm{P}^{\mathrm{hist}}$[57]分析可知,这些方法的本地扰动机制大都是WRR[22]机制线性分解形式的变种,混洗模型均是单混洗方,而误差精度均没有接近$O(1/\varepsilon)$. PEOS[35,36]方法采用多消息多混洗的框架(如图2(c)所示)响应直方图查询. PEOS方法利用OLH[13]机制对每个用户的v_i值进行扰动,然后结合同态加密[39]把扰动值以秘密共享的方式分解成r份,再把每份消息报告给对应的r个混洗方. 为了防止部分用户与收集者$\mathcal{A}$共谋,每个混洗方对于每个消息随机添加ρ个假数据. 收集者$\mathcal{A}$收集$nr(1+\rho)$条混洗消息后,需要对每个桶的计数进行修正,修正结果为$\hat{x}_j\leftarrow(\hat{x}_j\frac{n+\rho}{n}-\frac{\rho}{nd})$,结合修正结果即可响应直方图查询. 相比于Private-coin方法的Hadamard转换、Public-coin方法的Count-Min转换以及$\mathrm{P}_{\mathrm{FLIP}}$的One-Hot编码,PEOS方法利用哈希编码对用户值v_i进行本地映射到,并采用OLH机制本地扰动哈希地址. 相比于其他编码方法,PEOS方法的误差不受值域d的影响. 然而由于每个混洗方加入ρ个随机值并且使用了加密算法使得PEOS方法的通信代价与计算代价很高. 不同于上述方法,CDHR[55]利用无限可分噪声机制产生联合噪声来本地扰动每个用户的v_i值. 该方法首先结合

一元编码处理v_i值，然后结合$\mathcal{D}^1=\mathcal{D}^2$=NBin(1, $e^{-\varepsilon}$)以及$\mathcal{D}^3$=NBin(r, p)三种无限可分分布对编码之后的v_i值进行扰动. 设$Z_1\sim\mathcal{D}^1$、$Z_2\sim\mathcal{D}^2$、$Z_3\sim\mathcal{D}^3$，每个用户发送$v_i+Z_1+Z_3$个+1以及Z_2+Z_3个−1给$\mathcal{S}$. $\mathcal{A}$结合$\mathcal{S}$发送的±1数目重构直方图的桶计数来响应直方图查询.

目前，(ε,δ)-CDP模型下直方图查询精度最好能够达到$\Theta\left(\min\left(\log(1/\delta)/n\varepsilon,\ \log d/\varepsilon, n\right)\right)$[63]，而($\varepsilon$,$\delta$)-LDP模型下的直方图查询精度最好为$\Theta\left(\sqrt{\log d}\right)/\sqrt{n}\,\varepsilon\Big)$[14]. 表6从通信代价与查询精度方面综合分析了(ε,δ)-SDP模型下直方图查询(或者频率查询)的主要方法. 由表6可知，(ε,δ)-SDP模型下的直方图查询精度均处于区间$\left[\Theta\left(\min\left(\log(1/\delta)/n\varepsilon,\ \log d/\varepsilon, n\right)\right), \Theta\left(\sqrt{\log d}/\sqrt{n}\,\varepsilon\right)\right]$. 在图2($b$)模型下，大多数方法在通信代价与查询精度之间寻求平衡点，其中P_{FLIP}方法的通信代价最小，但其查询精度仍受值域d的影响. P^{hist}方法的查询精度较高，脱离了值域d的影响，然而，该方法不满足(ε,δ)-LDP. 图2(c)模型下的方法通常利用多消息多混洗方式防御多种攻击，同时提供查询精度. 然而该类方法的计算复杂性较高. 目前表6中的方法大都针对低维直方图查询，同时假设每个维度值域均相等的条件下响应查询. 因此，如何设计针对用户数据分布不均且维度值域不相等的高维直方图查询方法是个大的挑战.

表6 SDP下直方图查询主要方法对比

方法名称	所属框架	攻击类型	优点	缺点	通信代价	误差
R^{PH}[22]	单消息单混洗框架	一	定点编码, 隐私放大	通信代价较高	$O(\log d)$	$O\left(n^{1/3}\log^{2/3}(1/\delta)/\varepsilon^{4/3}\right)$
BitSum[27]	多消息单混洗框架	一	One-Hot编码, 按位扰动	通信代价较高	$O(d)$	$O\left(\sqrt{\log d\log(1/\delta)}/\varepsilon n\right)$
Private-coin[60]	多消息单混洗框架	一	假数据填充,Had-amard编码	误差受d的影响	$O\left(\log(1/\varepsilon\delta)/\varepsilon^2\right)$	$O\left(\frac{\log d}{n}+\sqrt{\log d\log(1/\varepsilon\delta)}/\varepsilon n\right)$
Public-coin[46]	多消息单混洗框架	一、三	Count-Min数据结构	误差仍受d的影响	$O\left(\log^3 d\log\left(\log(d/\delta)\right)/\varepsilon^2\right)$	$O\left(\sqrt{\log^3 d\log\left(\log(d/\delta)\right)}/\varepsilon n\right)$
pureDUMP[62]]	多消息单混洗框架	一、三	假数据填充,无需扰动	通信代价较高	$O\left(d\log(1/\delta)/\varepsilon^2 n\right)$	$O\left(\sqrt{\log(1/\delta)}/\varepsilon n\right)$
mixDUMP[62]	多消息单混洗框架	一、三	假数据填充,WRR扰动	误差仍受d的影响	$O\left(d\log(1/\delta)/\varepsilon^2 n\right)$	$O\left(\frac{1}{n\varepsilon}\left(1+d/(e^{10\varepsilon}-1)\right)\sqrt{\log(1/\delta)}\right)$
P_{FLIP}[44]	多消息单混洗框架	一、三	One-hot编码,WRR扰动	误差仍受d的影响	2	$O\left(\frac{\log d}{n}+\sqrt{\log d\log(1/\delta)}/\varepsilon n\right)$
P^{hist}[57]	多消息单混洗框架	一、二	误差不受值域大小约束	不满足LDP	$O(d)$	$O\left(\log(1/\delta)/\varepsilon^2 n\right)$
CDHR[55]	多消息单混洗框架	一、三	无限可分分布产生噪声	通信代价较高	$1+O\left(d\log^2(1/\delta)/\gamma\varepsilon^2 n\right)$	$O\left(\log d/n\varepsilon\right)$
PEOS[35,36]	多消息多混洗框架	一、三、四	同态加密、OLH扰动	计算代价高	$O(\log n)$	$O\left(\log(2/\delta)/\varepsilon^2 n^2\right)$

6 混洗差分隐私下的统计分析

在机器学习过程中，无论是梯度、中间模型还是最终模型参数均有可能泄露用户的隐私. 基于此，(ε,δ)-CDP与(ε,δ)-LDP模型下的机器学习得到了较为充分的研究. 这些模型下的机器学习、深度学习以及联邦学习等均可归结为隐私优化(凸优化与非凸优化)问题. 而隐私优化问题又可以归结为经验风险最小化问题(ERM)，具体含义如式(30)所示：

$$\arg\min_{\theta\in\mathcal{C}}\left(\mathcal{L}(\theta;D):=\frac{1}{n}\sum_{i=1}^{n}\ell\left(\theta;d_i\right)\right)\tag{30}$$

其中，分布式数据集$D=\{d_1,d_2,\cdots,d_n\}$，$\mathcal{L}(\theta;D)$与$\ell(\theta;d_i)$分别表示目标函数与损失函数，$\mathcal{C}$通常表示凸集合.

随机梯度下降(Stotastic Gradient Descent, SGD)[64]是解决ERM问题比较有效的方法. ERMOP[65]方法是(ε,δ)-CDP模型下利用SGD解决凸优化问题的早期代表，该方法利用拉普拉斯机制[1]与高斯机制[7]对目标函数进行扰动，进而获得满足(ε,δ)-CDP模型的全局模型θ. 相比于ERMOP，Aexp-sam[66]方法针对凸优化问题同样采用目标函数扰动来求解全局模型θ的最小下界. 不同于ERMOP方法与Aexp-sam方法，DP-SGD[8]方法针对深度学习下的非凸优化问题，利用MA(Moments

Accountant)技术控制每次迭代梯度的噪声大小以及隐私损失. 相比于(ε,δ)-CDP模型下梯度迭代时的强组合性质[7],DP-SGD方法求解模型θ时的隐私损失更小. 不同于(ε,δ)-CDP模型,(ε,δ)-LDP模型下的θ求解相当于分布式SGD问题. 隐私分布式SGD的关键在于每个用户u_i基于本地数据d_i更新梯度时如何使其满(ε,δ)-LDP,以及如何控制用户端与收集端的通信代价. 例如,DME[67]方法利用k-位随机量化技术[68]对每个用户的局部梯度向量进行裁剪,每个用户对裁剪后的向量添加满足二项分布Bin(m,p)的噪声. (ε,δ)-CDP与(ε,δ)-LDP模型下的联邦学习也得到了充分的研究. 例如DP-FedAvg[69]方法利用高斯机制与MA技术求解满足(ε,δ)-CDP的θ. 与DP-SGD方法不同,DP-FedAvg方法要求每个用户的数据不离开本地. 相比于DP-FedAvg,MB-SGD[70]方法允许每个用户采取高斯机制本地扰动自身的梯度参数,并把噪声梯度报告给服务器. 服务器结合式(30)对每一轮的全局参数模型θ进行聚合.

上述这些方法均是在(ε,δ)-CDP与(ε,δ)-LDP模型下讨论机器学习、深度学习与联邦学习. 而基于(ε,δ)-SDP模型下的分析工作较少. 本节结合图2中的3种混洗差分隐私模型,详细分析3种模型下的机器学习与联邦学习研究进展.

6.1　混洗差分隐私下的机器学习

不同于(ε,δ)-CDP与(ε,δ)-LDP模型下机器学习模型的训练方法,基于(ε,δ)-SDP模型求解全局模型参数θ时,每个用户结合自身的数据d_i本地扰动局部模型参数θ_i,并以消息向量的形式发送给混洗方$\mathcal{S}$,而收集者$\mathcal{A}$结合$\mathcal{S}$发送的所有参数向量,利用式(28)求解最终的全局模型参数θ. 最终的全局参数θ通常满足(ε,δ)-CDP. MLDP-SGD[53]是基于(ε,δ)-SDP模型的早期求解全局参数θ的方法. 该方法把LDP-SGD[71]方法嵌入ESA架构[19]中来训练全局θ. 在第t轮迭代中,该轮中的所有用户结合自身数据与式(28)求解本地梯度g_t. 利用Duchi[71]机制扰动g_t后报告给混洗方$\mathcal{S}$. 收集者$\mathcal{A}$对所有噪声梯度的混洗结果进行均值化处理后,更新第t轮的参数θ_t,即$\theta_{t+1}\leftarrow\theta_t-\eta_t G_t$,其中$G_t$表示第$t$轮所有噪声梯度的平均值. 此外,MLDP-SGD方法结合强组合性质[72]使得每轮的噪声梯度混洗结果满足(ε,δ)-CDP. 不同于MLDP-SGD方法,DPSGD[43]在每轮迭代过程中首先对n个数据点进行无放回采样,被抽取的用户结合自身数据点计算梯度g_t. 本地裁剪g_t后添加随机高斯噪声,即:$b_i+\left(g_t/\max\left(1,\|g_t\|_2\right)\right)$,其中,$b_i\sim\mathcal{N}(0,\sigma^2 I_d)$,$\|g_t\|_2$表示$g_t$的$L_2$范数. 然后,每个用户根据$\theta'_{t+1}\leftarrow\theta_t-\eta_t\left(b_i+\left(g_t/\max\left(1,\|g_t\|_2\right)\right)\right)$($\eta_t$表示本地学习率)计算本地模型参数$\theta'_{t+1}$,最后再把$\theta'_{t+1}$报告给混洗方$\mathcal{S}$. 收集者$\mathcal{A}$均值化处理所有的本地模型参数即可获得全局模型参数$\theta$. $\mathrm{P}_{\mathrm{SGD}}$[73]利用$\mathrm{P}_{\mathrm{VEC}}$[73]方法与小批量SGD技术以序列交互的方式学习全局参数模型θ. 不同于DPSGD[43]方法的本地随机高斯噪声,$\mathrm{P}_{\mathrm{VEC}}$[73]是$\mathrm{P}^{\mathrm{hist}}$[57]方法在多个维度上的扩展,分别利用One-Hot机制与ZSUM[40]方法对数据d_i的每个维度进行编码与扰动. $\mathrm{P}_{\mathrm{SGD}}$[73]方法在序列交互式求解全局$\theta$过程中,每个用户仅参加单轮迭代,结合本地数据$d_i$利用$\mathrm{P}_{\mathrm{VEC}}$方法求解$g_t$后,报告给混洗方$\mathcal{S}$. $\mathcal{A}$同样均值化所有轮的θ_t后即可获得全局模型θ. 不同于$\mathrm{P}_{\mathrm{SGD}}$[73],$\mathrm{P}_{\mathrm{GD}}$[73]结合全交互式混洗协议求解$\theta$,允许单个用户参与多轮混洗,进而最终全局模型$\theta$的误差低于$\mathrm{P}_{\mathrm{SGD}}$.

不同于上述基于(ε,δ)-SDP的方法,$\mathrm{A}_{\mathrm{cldp}}$[74]方法利用纯差分隐私的扩展模型Rényi-差分隐私[75,76]训练θ. 该方法利用基于二次采样的混洗模型保护θ的训练过程. 在第t轮训练中,$\mathrm{A}_{\mathrm{cldp}}$方法利用二次采样技术以概率$\gamma$抽取$\gamma n$个用户参与该轮训练. 被抽中的用户结合数据$d_i$求解梯度$g_t\leftarrow\nabla_{\theta t}f(\theta_t,d_i)$. 利用GRR机制[5]本地扰动裁剪后的梯度,然后把扰动后的梯度值发送给混洗方$\mathcal{S}$. 收集者$\mathcal{A}$利用混洗后的γn个梯度更新第t轮的参数θ_t. 通过二次采样技术,$\mathrm{A}_{\mathrm{cldp}}$方法实现了隐私放大,并给出了Rényi-差分隐私的上下边界. 现有基于(ε,δ)-SDP模型的全局模型θ求解方法通常采用均匀采样技术抽取每轮的用户. 用户被选中后必须参加该轮的迭代训练. 然而实际系统中用户掉线、网络通信差等因素会直接影响模型θ的训练精度. 基于此,FDPSGD[77]方法利用用户随机签入策略克服了上述方法中用户掉线问题. 在第t轮迭代中,每个用户以随机签入概率ζ决定是否参与该轮训练,该签入概率符合二项分布Bern(ζ). 若某个用户成功加入第t轮训练,则该用户基于本地数据d_i计算梯度、裁剪梯度以及添加高斯噪声后,发送给混洗方$\mathcal{S}$. 收集者$\mathcal{A}$结合式(28)更新本轮的全局模型θ. FDPSGD满足Rényi-差分隐私且实现隐私放大,并给出比强组合定理更紧的上下边界.

目前(ε,δ)-CDP模型下利用SGD求解全局参数θ的最小误差为$O\left(1/\sqrt{n}+\sqrt{d}/n\varepsilon\right)$[78],$(\varepsilon,\delta)$-LDP模型下求解$\theta$的最小误差为$O\left(1/\sqrt{n}+\sqrt{d}/\sqrt{n}\,\varepsilon\right)$[17]. 根据表7可知,基于$(\varepsilon,\delta)$-SDP模型下的学习方法误差均处于$\left[O\left(1/\sqrt{n}+\sqrt{d}/n\varepsilon\right),O\left(1/\sqrt{n}+\sqrt{d}/\sqrt{n}\,\varepsilon\right)\right]$之间. 局部模型参数的混洗操作起到了隐私放大作用. 然而,客户端如何控制噪声大小、如b何裁剪局部梯度以及如何通过梯度压缩减少通信代价仍然是挑战性问题.

6.2　混洗差分隐私下的联邦学习

联邦学习遵循“数据不动,模型动”的原则解决数

表7 SDP下基于SGD求解θ的主要方法对比

方法名称	所属框架	攻击类型	损失函数构造特点	缺点	通信代价	误差
MLDP-SGD[53]	多消息多混洗框架	一、二、三、四	满足L-利普希茨,凸函数	通信代价高	$O(d)$	$\log^2 n\sqrt{d}/\varepsilon n$
DPSGD[43]	多消息单混洗框架	一	满足L-利普希茨,强凸函数	梯度删减阈值无法定界	$O(d)$	$O\left(\left(\log^2(n/\delta)\,d\log(1/\delta)\right)/n\varepsilon^2\right)$
P_{SGD}[73]	多消息单混洗框架	一、二	满足L-利普希茨,凸函数且光滑	通信代价高	$O(d)$	$O\left(1/\sqrt{n}+d^{2/5}/\varepsilon^{4/5}n^{4/3}\right)$
P_{GD}[73]	多消息单混洗框架	一、二	满足L-利普希茨,强凸且光滑	通信代价高	$O(d)$	$O\left(1/\sqrt{n}+d/\varepsilon^2n^2\right)$
A_{cldp}[74]	单消息单混洗框架	一	满足L-利普希茨,凸函数	没有考虑用户掉线问题	$O(\log d)$	$O\left(a^2d/\varepsilon^2n\right)$
FDPSGD[77]	多消息单混洗框架	一	满足L-利普希茨,凸函数	通信代价高	$O(d)$	$O\left(\left(\log^2(n/\delta)\,d\log(1/\delta)\right)/n\varepsilon^2\right)$

据孤岛与数据安全问题备受研究者关注[72]. 模型梯度作为一种特殊的数据类型,在用户与收集者之间频繁交互时,恶意攻击者可以对梯度进行深度攻击[79,80],进而推理出原样本信息. 基于此,出现了系列基于(ε,δ)-CDP与(ε,δ)-LDP模型下的联邦梯度保护方法. DP-FedAvg[69]与CSDP-Fed[81]方法是(ε,δ)-CDP模型下的典型代表. 每一轮迭代过程中,收集者先聚集每个抽样用户提交的梯度,对梯度裁剪聚合后添加高斯噪声. LDP-Fed[82]、D^2P-FED[83]、UDP[84]、DPFedAvg-M[85]、CE-FedAvg[86]等方法是(ε,δ)-LDP模型下保护联邦梯度的典型代表. 每轮迭代被选中的用户利用本地数据求出本轮梯度,裁剪后添加高斯噪声. $\mathcal{A}$聚集与均值化所有用户的噪声梯度后,即可获得该轮的全局模型θ.

基于(ε,δ)-SDP模型的联邦学习同样受到研究者的关注. LDP-FL[87]方法基于图2(b)模型训练全局模型θ. 在每轮迭代过程中,该方法首先假设模型θ的每个分量所处的范围是$[c-r,c+r]$,其中c表示该范围的中心点,r表示该范围的半径,其作用是限制θ每个分量的大小(类似于裁剪操作). 然后利用Duchi[71]方法本地扰动范围$[c-r,c+r]$中的每个分量. 每轮迭代中每个用户把扰动后的本地模型分割后发送给混洗方$\mathcal{S}$. 收集者结合混洗结果利用式(30)可计算出该轮的全局参数模型θ. 然而,该方法继承了Duchi方法本身的缺点,即每个分量被扰动与修正后,却远离了区间$[c-r,c+r]$中的真实值. 此外,该方法扰动θ的每个分量时需要分割隐私预算ε,而当θ的维度过大时,最终训练的模型精度比较低. 不同于LDP-FL方法,FLAME[88]方法在每轮迭代中利用Blanket[4]方法本地扰动θ的每个分量,并结合维度二次采样与top-k选择技术来避免维度过高带来的影响. 该方法首先选择k个重要的维度,然后利用Blanket(ε/k)扰动每个被选中的维度. 此外,在实现top-k个维度选择时,牺牲部分隐私预算保护被选择维度的索引. 尽管FLAME方法从θ的维度大小角度考虑联邦学习问题,然而该方法由于使用Blanket方法扰动每个分量,θ的误差不可避免地受维度大小与稀疏性影响. 维度越大越稀疏则θ的误差越大. 与FLAME方法不同,F_{dkmnt}[89]方法利用离散傅里叶变换[90]技术把θ的原始维度变换成k个傅里叶系数,再利用Blanket方法分别本地扰动k个系数. 该方法能够处理高维且稀疏的模型参数. FLAME与F_{dkmnt}方法是通过top-k采样以及傅里叶变换减少每一轮用户与收集者之间的通信代价. 然而,如何选择参数k是个难点. 不同于FLAME与F_{dkmnt}方法,CLDP-SGD[30,91]方法基于小批量SGD技术利用优化压缩方法来平衡用户与收集者之间的通信代价、模型参数θ的隐私性与可用性. 该方法在当前轮迭代中利用二次采样技术先对用户进行随机抽样,再对用户下的数据进行抽样. 被抽中的每个用户结合自身样本计算该轮梯度,形如$g_t \leftarrow g_t \leftarrow \nabla_{\theta t} f(\theta_t, d_{ij})$. 裁剪$g_t$后对其进行二进制编码,再利用Hadamard机制[16]或者Duchi机制[71]对裁剪后的梯度进行本地扰动. 混洗方$\mathcal{S}$获取该轮的所有梯度后进行混洗,然后发送给收集者$\mathcal{A}$,$\mathcal{A}$利用式(28)计算该轮的全局模型参数θ. CLDP-SGD方法通过压缩技术获得了最优的参数$k\left(k=O(\log d)\right)$,参数$\theta$的误差上下边界,并且克服了二次采样造成的数据非等概率问题,最后实现了隐私放大. 尽管CLDP-SGD方法支持每轮迭代的用户动态变化,然而每轮被混洗方抽取的用户数量是定值,忽略了该轮中用户掉线情况. 此外,CLDP-SGD由于模型θ的每个分量需要分割隐私预算仅是次优的保护方法. 基于此,CoCo[92]方法利用θ的分量负相关策略实现了最优保护效果. 与FDPSGD[77]方法类似,Dss-SGD[93]方法对CLDP-SGD方法进行了拓展. 在每轮迭代中允许用户以一定的概率随机签入. 然而,该方法由于签入概率固定,缺乏灵活性.

由表8可知,现有的方法大都采用压缩或者转换方

式来提高全局模型θ的可用性,以及减少用户与混洗方之间的通信代价. 利用随机签入方式解决用户掉线问题. 不同于分布式机器学习,联邦学习中,每个用户拥有小型数据集而不是单个数据点. 因此,如果FDPSGD[77]方法与A_{cldp}[73]方法中的本地数据是小型数据集,则这些方法同样适用于联邦学习环境. 现有的联邦学习方法应对的攻击场景较为单一,(ε,δ)-SDP模型如何结合密码技术解决多攻击场景下联邦学习是挑战性问题.

表8　SDP下联邦学习主要方法对比

方法名称	所属框架	攻击类型	损失函数构造特点	缺点	通信代价	误差
LDP-FL[87]	多消息单混洗框架	一	满足凸函数	通信代价高	$O(d)$	$O\left(r\sqrt{-\log\beta}/\varepsilon\sqrt{n}\right)$
FLAME[88]	多消息单混洗框架	一、四	满足凸函数	模型维度稀疏性问题	$O(k), k\ll d$	$O\left(\log\log n\sqrt{\log 1/\delta}/\varepsilon\right)$
F_{dkmnt}[89]	单消息单混洗框架	一	满足凸函数	参数k的选择比较困难	$O(k), k\ll d$	$O\left(k^{8/3}\sqrt{\log 1/\delta}/\varepsilon n^{5/3}\right)$
CLDP-SGD[91]	多消息单混洗框架	一	满足L-利普希茨,凸函数	忽略了用户掉线问题	$O(\log d)$	$O\left(a^2 d/n\varepsilon^2\right)$
CoCo[92]	多消息单混洗框架	一、二	满足L-利普希茨,凸函数	忽略了用户掉线问题	$O(\log d)$	$O\left(\sqrt{s\log d/\beta}/\varepsilon\sqrt{n}\right)$
Dss-SGD[93]	多消息单混洗框架	一	满足L-利普希茨,凸函数	随机签入概率固定	$q\times O(\log d)$	$O\left(d/qn\varepsilon^2\right)$

7　混洗差分隐私下的隐私放大

隐私放大效应是SDP模型的主要特点之一. 混洗方$\mathcal{S}$收集每个用户发送满足(ε_l,δ)-LDP的消息后进行混洗,而混洗后的消息向量满足(ε_c,δ)-CDP. ε_c通常被表示成$\varepsilon_c(\varepsilon_l,\delta,n)$函数,且$\varepsilon_c\ll\varepsilon_l$. 从混洗方的角度来看,隐私保护程度增强. 目前,基于图2中3种SDP模型的隐私放大问题主要集中在如何利用ε_l,δ与n计算ε_c的上下边界,进而度量隐私放大的程度. A_{sl}[41]方法是利用混洗技术实现聚集查询的早期代表. 该方法利用"先混洗-后扰动"协议交互式地实现了聚集查询与隐私放大. 当$n\geqslant 1\,000$,$0<\varepsilon_l<1/2$,$0<\delta<0.01$时,其查询结果满足隐私放大,即$\varepsilon_c=12\varepsilon_l\sqrt{(\log 1/\delta)/n}$. 然而,当$\varepsilon_l>1$时,该方法的隐私放大效果比较差,$\varepsilon_c=O\left(\mathrm{e}^{3\varepsilon_l}\sqrt{(\log 1/\delta)/n}\right)$,误差为$\Theta(n^{5/12})$且接近于$(\varepsilon_l,\delta)$-LDP下的聚集误差$\Theta(n^{1/2}/\varepsilon)$. 不同于$A_{sl}$方法,Blanket[4]方法与PBS[27]方法分别实现了$\varepsilon_l>1$情况下的隐私放大效果. 其中,PBS方法利用二进制扰动机制实现二进制数据上的聚集查询. 并且证明了当$\varepsilon_l\sim\ln(n)$时,该方法能达到$\varepsilon_c=\sqrt{32\log(4/\delta)(\mathrm{e}^{\varepsilon_l}+1)/n}$的隐私放大效果. 不同于PBS方法,Blanket把隐私放大效果拓展到以GRR为代表的多维类别数据当中,当$\delta\in(0,1)$,$1<\varepsilon_l\leqslant\log\left(n/\log(1/\delta)\right)/2$时,该方法的放大效果为$\varepsilon_c=O\left((1\wedge\varepsilon_l)\mathrm{e}^{\varepsilon_l}\sqrt{(\log 1/\delta)/n}\right)$. PBS方法的放大边界比Blanket方法的边界紧了2倍,然而该方法仅适合于二进制扰动机制. 尽管A_{sl}、Blanket、PBS这些方法给出了ε_c的放大边界,然而这些边界通常是渐进次优化的. 相比于上述方法,kRR[43]方法利用克隆技术完成本地扰动. 所谓克隆技术即是:对于任意给定的一个用户值v_i,如果该用户的值v_i按照公式$\mathcal{R}(v_i)=\mathcal{R}(v_i)/\mathrm{e}^{\varepsilon_l}+Q(v_i,v_j)(1-1/\mathrm{e}^{\varepsilon_l})$扰动,则$v_i$值以$1/\mathrm{e}^{\varepsilon_l}$的概率克隆了$\mathcal{R}(v_j)$值. 用户值被克隆扰动且混洗后,获得了更紧的优化隐私放大效果. DOSP[94]方法对kRR方法进行拓展,允许t个用户共谋的情况下,隐私放大效果依然与kRR方法保持一致,即$\varepsilon_c=O\left((1-\mathrm{e}^{-\varepsilon_l})\sqrt{\left(\mathrm{e}^{\varepsilon_l}\log 1/\delta\right)/n-t}\right)$. 相比于$A_{sl}$、Blanket、PBS、kRR与DOSP方法,Variation-ratio[95]方法利用变分-比率缩减策略计算出了一个更紧的隐私放大边界,且适用于单消息和多消息混洗框架. 由文献[35]可知,Blanket方法与kRR方法的隐私放大效果容易受类别数据值域d的影响,过大的d值容易松散放大效果. 基于此,MURS[35,36]方法结合图2(c)实现了对OLH[13]机制的隐私放大. 该方法利用哈希技术减弱了对值域d的直接依赖,即,$\varepsilon_c=\sqrt{14\log(2/\delta)(\mathrm{e}^{\varepsilon_l}+g-1)/n-1}$,且$g\ll d$. MURS方法的$\varepsilon_c$小于Blanket方法的$\varepsilon_c$,进而达到更强的隐私保护效果. 与MURS方法类似,P_{FLIP}[44]方法在估计直方图分布时结合哈希压缩技术来提升隐私放大效果,以及减少用户与混洗方之间的通信代价,其放大效果优于MURS方法.

相比于计数查询、直方图估计等简单的数据分析,SDP模型下的隐私放大效果在机器学习与联邦学习中更为显著. 由6.1、6.2节可知,机器学习与联邦学习通常采用强组合性质[72]确保整个迭代过程满足差分隐私. 而采用隐私放大技术的保护效果强于强组合性质. A_{swap}[96]方法结合固定滑动窗与随机签入技术实现梯度混洗. 在每轮迭代中每个用户以一定概率随机签入该轮梯度计算,签入的用户对自身梯度添加高斯噪声后发送给混洗方. 该方法的隐私放大效果分为$\varepsilon_l>1$与$\varepsilon_l\leqslant 1$两种情况,而$\varepsilon_l\leqslant 1$的隐私放大的边界比较紧. 根据文

献[43]可知,$\varepsilon_l>1$情况下的隐私放大在分布式机器学习与联邦学习中才更具有意义. DPSGD[43]方法通过每轮对本地梯度的混洗,实现了对高斯机制的隐私放大,$\varepsilon_l>1$情况下其放大效果达到$O\left(\sqrt{\mathrm{e}^{\varepsilon_l}(\log 1/\delta)/n}\right)$. FLAME[88]方法结合文献[97]中的二次抽样技术均匀抽取梯度中部分维度进行本地与混洗,并利用哑元填充实现了联邦模型θ的双倍隐私放大. 然而,该方法没有指出每轮用户的状态,仅假设每轮包括所有用户来实现隐私放大效果;若每轮的用户数由均匀采样获得,则该方法的隐私放大效果有可能不成立. 为了适应实际的联邦学习环境,CLDP-SGD[91]方法结合动态用户采样、数据采样以及梯度混洗实现了每轮迭代过程的隐私放大. 由于用户采用与数据采样可能采用非均匀的采样点,进而不能直接利用文献[97]与FLAME方法中的二次采样技术产生隐私放大效果. 根据文献[94]可知,上述基于(ε,δ)-SDP的隐私放大方法不太适合本地多步骤迭代的机器学习方法,其主要原因是隐私放大效果较差. 比起那些本地没有多步迭代方法的放大效果通常多出$O\left(\sqrt{(\log 1/\delta)}\right)$. 基于此,出现多种结合$(\varepsilon(\alpha),\alpha)$-Rényi差分隐私与混洗的分布式机器/联邦学习方法,并且获得较好的隐私放大效果. 文献[41]最早结合Rényi差分隐私与混洗实现了分布式学习下的隐私放大,然而该方法在$\varepsilon_l>1$时,放大效果却为$O(\alpha\mathrm{e}^{6\varepsilon_l}/n)$. RDP[76]方法利用克隆技术减少文献[41]的α-Rényi散度过大问题,该方法的隐私放大效果为$O(\alpha\mathrm{e}^{2\varepsilon_l}/n)$. 然而,这两种方法的放大效果均是渐进次优化. 基于此,DOPS[94]方法利用kRR[43]方法的克隆技术实现了隐私放大的最优效果,即$O(\alpha\mathrm{e}^{\varepsilon_l}/n)$. 由表9可知,无论是$(\varepsilon,\delta)$-SDP模型下的统计查询还是统计分析,混洗操作均可以实现隐私放大效果. 然而,如何在混洗方不可信,或者去掉混洗方的情况下,实现隐私放大是个挑战性问题.

表9 SDP框架下隐私放大方法对比分析

方法	放大条件	放大效果	应用
A_{sl}[41]	$n\geqslant 1\,000,\varepsilon_l<1/2,0<\delta<1/100$	$\varepsilon_c=O\left(\mathrm{e}^{2\varepsilon_l}(\mathrm{e}^{\varepsilon_l}-1)\sqrt{\log(1/\delta)/n}\right)$	二进制计数查询
PBS[27]	$\varepsilon_l>1,0<\delta<1$	$\varepsilon_c=O\left(\mathrm{e}^{\varepsilon_l/2}\sqrt{\log(1/\delta)/n}\right)$	二进制计数查询
Blanket[4]	$1<\varepsilon_l\leqslant\log\left(n/\log(1/\delta)\right)/2$	$\varepsilon_c=O\left(\mathrm{e}^{\varepsilon_l}\sqrt{\log(1/\delta)/n}\right)$	直方图/频率估计
kRR[43]	$\delta\in[0,1],1<\varepsilon_l\leqslant\log\left(n/16\log(2/\delta)\right)$	$\varepsilon_c=O\left((1-\mathrm{e}^{-\varepsilon_l})\left(\sqrt{\mathrm{e}^{\varepsilon_l}\log(1/\delta)/n}\right)\right)$	直方图/频率估计
DOSP[94]	$\delta\in[0,1],1<\varepsilon_l\leqslant\log\left(n/16\log(2/\delta)\right)$	$\varepsilon_c=O\left((1-\mathrm{e}^{-\varepsilon_l})\left(\sqrt{\mathrm{e}^{\varepsilon_l}\log(1/\delta)/n-t}\right)\right)$	直方图/频率估计
MURS[35,36]	$\delta\in(0,1),1<\varepsilon_l\leqslant\log\left((n-1)/\log(2/\delta)+1-g\right)$	$\varepsilon_c=O\left(\sqrt{\mathrm{e}^{\varepsilon_l/2}\log(1/\delta)/n-1}\right)$	直方图/频率估计
$\mathrm{P}_{\mathrm{FLIP}}$[44]	$\delta\in(0,1),\varepsilon_l\leqslant\log\left(\varepsilon_c^2 n/256\log(4/\delta)\right)$	$\varepsilon_c=O\left(\sqrt{\mathrm{e}^{\varepsilon_l}\log(1/\delta)/n}\right)$	直方图/频率估计
Variation-ratio[95]	$\delta\in(0,1),\varepsilon_l=\Theta(1)$	$\varepsilon_c=O\left(\sqrt{\beta(\mathrm{e}^{\varepsilon_l}-1)\log(1/\delta)/n}\right)$	分布式机器学习
$\mathrm{A}_{\mathrm{swap}}$[96]	$\delta\in(0,1),\varepsilon_l\leqslant 1$	$\varepsilon_c=O\left(\varepsilon_l\mathrm{e}^{1.5\varepsilon_l}\sqrt{\log(1/\delta)/n}\right)$	分布式机器学习
$\mathrm{A}_{\mathrm{swap}}$[96]	$\delta\in(0,1),\varepsilon_l>1$	$\varepsilon_c=O\left(\mathrm{e}^{2.5\varepsilon_l}\sqrt{\log(1/\delta)/n}\right)$	分布式机器学习
DPSGD[43]	$\delta\in[0,1],1<\varepsilon_l\leqslant\log\left(n/16\log(2/\delta)\right)$	$\varepsilon_c=O\left((1-\mathrm{e}^{-\varepsilon_l})\left(\sqrt{\mathrm{e}^{\varepsilon_l}\log(1/\delta)/n}\right)+\mathrm{e}^{\varepsilon_l}/n\right)$	分布式机器学习
FLAME[88]	$\varepsilon_l\leqslant\beta d\log\left(n/\log\left((2d\beta^2+\beta)/\delta\right)\right)/2$	$\varepsilon_c=O\left((1\wedge\varepsilon_l/\beta d)\mathrm{e}^{\varepsilon_l/\beta d}\beta^{1.5}\left(\sqrt{d\log(\beta d+\beta^2 d/\delta)/n}\right)\right)$	联邦学习
CLDP-SGD[91]	$\delta>0,\varepsilon_l\leqslant\log\left(qn/\log(1/\delta)\right)/2$	$\varepsilon_c=O\left(\min\{\varepsilon_l,1\}\mathrm{e}^{\varepsilon_l}\left(\sqrt{\log(1/\delta)/qn}\right)\right)$	联邦学习
RDP[76]	$\alpha\leqslant c_0(n/\mathrm{e}^{5\varepsilon_l})^{1/4},\varepsilon_l>0,c_0>0$	$\varepsilon_c(\alpha)=O\left(\alpha\mathrm{e}^{2\varepsilon_l}/n\right)$	分布式机器/联邦学习
DOSP[94]	$\alpha\leqslant c_0 n/\varepsilon_l\mathrm{e}^{\varepsilon_l},\varepsilon_l>1,c_0>0$	$\varepsilon_c(\alpha)=O\left(\alpha\mathrm{e}^{\varepsilon_l}/n\right)$	分布式机器/联邦学习

8 未来工作

虽然SDP是CDP与LDP之间的桥梁模型,但是基于该模型的研究仍是一个新方向,很多挑战性问题亟待解决.

8.1 SDP模型下高维向量的聚集查询分析

由5.1节与5.2节可知,目前基于(ε_c,δ)-SDP模型的聚集查询通常聚焦于$\{0,1\}$或者$[0,1]$值域上的单维查询. 然而由$\{0,1\}$或者$[0,1]$构成的高维向量在实际应用

中比较常见. 例如,用户的网页浏览记录、购物记录、电影打分等均可以由{0,1}或者[0,1]构成高维向量. 如何响应由{0,1}或者[0,1]组合而成的高维向量上的聚集查询是个大的挑战. 每个用户拥有高维向量 $\boldsymbol{v}_i=(v_i^1,v_i^2,\cdots,v_i^d)\in[0,1]^d$. 收集者响应高维向量聚集查询的主要挑战来自5个方面:①如何基于图2中3种混洗模型,设计高效的本地扰动机制与混洗机制;②如果用户的高维向量是稀疏的,如何设计高效的压缩机制、转换机制或者抽样机制克服此类情况;③如果利用抽样技术克服高维向量的稀疏性,如何使抽样技术与混洗技术同时达到隐私放大效果;④如何把表4与表5的方法有效拓展到高维数据;⑤如何设计高效的本地编码机制减少通信代价.

针对上述挑战性问题,我们认为可以采用图2(a)与图2(b)中的模型来设计高维向量的聚集查询方法. 最直接的方法是直接利用表4与表5中的方法处理 $\boldsymbol{v}_i$ 的每个维度. 例如,基于图2(a)模型利用Blanket[4]方法对每个维度所在的[0,1]值域进行离散化分割,然后利用GRR[5]方法的线性分解方式扰动离散化结果. 然而,这样的聚集结果误差大,可用性较低. 均方误差通常为 $O(d^{8/3}n^{-5/3})$. 基于此,有以下可行方法可以有效解决高维向量聚集查询的诸多挑战.

(1) 针对向量 $\boldsymbol{v}_i$ 的稀疏性、通信代价以及本地编码问题,可以利用哈希压缩、Hadamard转换、傅里叶转换、Count Min结构转换,或者向量的量化与稀疏化处理技术来压缩高维向量 $\boldsymbol{v}_i$,进而保证较低的通信代价实现聚集查询,例如,可以结合文献[98]中的哈希压缩技术把 d 维的向量 $\boldsymbol{v}_i$ 压缩成 b 维的哈希地址来克服向量的稀疏性与通信代价问题.

(2) 针对向量 $\boldsymbol{v}_i$ 的本地编码、扰动问题以及高效混洗机制问题,可以利用离散化技术处理高维区间 $[0,1]^d$,每个用户利用定点编码技术处理自身向量 $\boldsymbol{v}_i$,利用(1)中的哈希压缩与转换技术处理编码之后的向量,再结合Blanket[4]、CESS[50]、PAFAM[29]等方法本地扰动. 此外,设计高效的混洗机制同样是高维向量聚集查询的挑战. 针对如Fisher-Yates[45]混洗机制的效率较低问题,我们可以采用堆排序技术来提高混洗效率.

(3) 针对向量 $\boldsymbol{v}_i$ 的稀疏性与通信代价问题,除了(1)中的方法,可以采用二次采样技术从高维向量 $\boldsymbol{v}_i$ 中抽取 b 维进行扰动并且混洗. 二次采样与混洗可以实现聚集查询结果的隐私双倍放大. 而要实现该过程的隐私放大,每个 $\boldsymbol{v}_i$ 中的数据需要均匀采样.

8.2　SDP模型下多维关系范围查询分析

人工智能技术的迅速发展,使得高维关系数据的获取与收集变得尤为容易. 令Tmall_Schema(age, salary, state, purchase)为一个购物APP的关系模式,Tmall为该模式的关系表. 收集者通过多维范围查询分析Tmall中的元组,能够提升服务方的产品与服务质量. 以Tmall为例,多维关系范围查询的SQL语句通常包括3种情况:

Q_1: SELECT COUNT (*) FROM Tmall WHERE state = 'CHINA' And salary∈[1,3];

Q_2: SELECT SUM(purchase) FROM Tmall WHERE age∈[2,4] And salary∈[1,3];

Q_3: SELECT AVG(purchase) FROM Tmall WHERE age∈[1,4] And salary∈[1,2].

目前,HIO[99]、EHIO[100]、AHIO[100]是基于(ε, δ)-LDP模型响应类似于Q_1、Q_2与Q_3的代表性方法. 这些方法采用层次树索引用户数据,利用HIO哈希并扰动用户本地多维元组. 尽管这些方法在(ε, δ)-LDP模型范畴内优于同类方法,然而它们无法避免收集者对用户身份的重甄别攻击. 多维范围查询的响应精度无法逼近(ε,δ)-CDP模型下的精度. 因此,基于(ε,δ)-SDP模型下的多维范围查询更具有应用价值. 由于Tmall中元组是多维数据,不能直接利用表4、表5与表7中的方法,其主要原因包括:这些方法假设用户手中拥有单个值,例如PBS[27]、ZSUM[40]、DDM[55]、Blanket[4]、P^{matrix}[101]等方法仅处理{0,1}值或者[0,1]间的实数值. 高维元组中每个属性的值域类别多样,通常以类别属性与数值属性最为常见. 而这些方法如何拓展到值域类别多样的多维范围查询是个挑战性问题.

我们认为,可以采用图2(b)或者图2(c)中的SDP模型解决该类问题. 对于关系表中每个类别属性的值域,利用层次树对其索引. 对于数值型值域,利用离散化处理,例如采用Blanket[4]方法的[0,1]值域分割方式,将其转换成离散型值域. 结合离散后的值域,再利用层次树进行索引. 所有的层次树利用笛卡尔积操作进行层次组合. 对于多维关系范围查询语句中谓词属性与WHERE子句中谓词按照层次树进行查询重写. 例如,Q_2语句中purchase属性、age属性以及salary属性按照其对应的值域层次树进行分割重写. 结合重写结构与层次组合,利用GRR[5]机制的线性分解结构(如式(14)所示)进行本地扰动,以多消息的方式发送给单个混洗方或者多个混洗方进行随机混洗. 收集者结合混洗结果即可响应多维范围查询. 此外,如果属性值域比较稀疏,可以利用哈希编码与Hadamard转换进行本地压缩,然后再利用WRR[22]或者GRR扰动.

8.3　SDP模型下异构数据上的直方图查询

根据表6中的方法对比分析可知,目前基于(ε,δ)-SDP模型下的直方图查询方法通常假设每个用户拥有同构的单个类别数据. 而实际应用中,每个用户可能贡献多个异构的类别数据,例如,某个病人在一家医院可能产

生多条健康医疗记录,或者通过该医院APP预定挂号记录等,而该病人希望自己的所有记录得到保护.此外,每个用户贡献的数据大小也不尽相同,有些用户贡献的数据大小无法控制,此类情况会导致直方图每个桶上的噪声值过大,通信代价过高.基于此类情况,我们不能直接采用表6中的方法处理这种异构类别数据.主要原因是表6中的方法通常是在统一值域中对用户的单一数据进行编码与扰动,例如One-Hot编码.如果编码值域过大或者稀疏,可以利用哈希与Hadamard转换进行压缩.而对于异构数据上的直方图查询,如何控制每个用户添加噪声大小、控制每个用户的贡献度以及控制用户与收集者之间的通信代价是该问题的主要挑战.针对此类问题,我们给出的解法如下:

(1)为了控制每个用户贡献数据的大小,我们可以采取最优截断长度l来裁剪每个用户的数据.而如何在(ε,δ)-SDP模型下计算l是个挑战性问题.在(ε,δ)-SDP模型下,我们可以利用隐私预算分割与用户分组策略探索用户贡献数据的长度分布情况,结合贡献数据的分布估计出最优截断长度.为了减少用户、混洗方以及收集者之间的通信代价,可以对裁剪后的用户数据进行采样,例如采用文献[14]中的采样技术可以实现$O(1)$的通信代价.

(2)用户数据异构情况下的最优截断长度l估计更具有挑战性.l的选择直接决定着直方图的噪声误差与桶计数的偏差.l过大导致误差大而偏差小,l过小导致偏差大而误差小.基于此,为了有效控制每个用户贡献数据的大小,可以基于图2(a)与图2(b)模型利用DP-SGD[8]方法来学习最优截断长度.每个用户把自身的数据长度本地添加离散高斯噪声后发送给混洗方,收集者结合混洗后的数据可以学习出l.

8.4 SDP模型下复杂图数据查询与分析

基于复杂无向图的子图模式(三角形、k-星、4-环等)查询、平均度查询,以及聚类系数分析等可以有效获得无向图的结构特征与连接趋势.例如,聚类系数可以度量结点之间聚成团的程度.然而,通过查询子图模式信息,可以推理出结点之间的敏感联系,例如通过三角形子图模式查询,可以知道某个结点拥有几个朋友结点.基于CDP与LDP模型的图数据查询与分析得到了较为充分研究.例如,RHMS[102]方法结合边-CDP实现了系列子图模式查询,θ-Cumulative[103]方法利用结点-CDP实现了度序列发布问题.而LF-GDPR[104]方法与LocalRR[105]方法分别基于边-LDP模型利用邻接矩阵与WRR[22]机制实现了子图模式查询.为了克服无向图数据的稀疏性,这两种方法首先利用拉普拉斯机制估计结点最大度.每个用户分别利用结点最大度与WRR机制裁剪扰动邻接向量.收集者收集所有用户的邻接向量,重构邻接矩阵后即可响应子图模式查询.相比于LF-GDPR[104]方法,LocalRR[105]方法给出子图模式查询的上下误差边界.边-CDP与边-LDP模型下的子图模式查询误差大小差别非常大,例如三角形查询在边-CDP下的误差为$O\left(d_{\max}/\varepsilon^2\right)$,而边-LDP下的误差为$O\left(d_{\max}^3 n/\varepsilon^2\right)$,其中$d_{\max}$为结点最大度.由于图数据自身的复杂性,如何在$(\varepsilon,\delta)$-SDP模型下分析该类数据是个大的挑战.

我们认为,可以采用图2中的(ε,δ)-SDP模型来处理无向图的查询与分析问题.一个最直接的做法是把LF-GDPR与LocalRR方法中的WRR机制按照式(18)进行重写,再扰动邻接行向量的每个位.然后结合图2(a)或者图2(b)中的混洗模型进行混洗.这样操作存在2种不足:若无向图有n个结点,邻接行向量是n维向量,WRR机制扰动n维向量的结果满足$(n\varepsilon,n\delta)$-LDP,进而导致$n\varepsilon$值非常大,弱化保护效果;其次这种做法忽视了邻接矩阵行向量过大与稀疏性问题.过大与稀疏的行向量直接会导致分析与查询结果的可用性低.我们建议的解法:结合无向图的邻接矩阵对称性,可以利用非对称的WRR机制扰动该矩阵的上三角或者下三角部分即可.收集者收到混洗方发送的消息后重构无向图.收集者发送噪声无向图副本给每个用户,每个用户结合噪声无向图获得相应的的子图模式,然后利用拉普拉斯机制保护子图后发送给混洗者.然而上述方法由于用户、混洗方与收集者之间的多轮交换,通信代价会很高.基于此,我们建议利用无向图自身的结构特征设计混洗方法.例如文献[106]利用结点之间的楔子结构设计了楔子混洗协议.每个用户利用楔子结构编码自身信息,此时编码结果值域为$\{0,1\}$,再利用重写后的WRR机制扰动即可.楔子混洗协议能够减少通信代价,然而如何把该协议拓展到最大团与k-跳路径查询是个挑战.

8.5 SDP模型下分布式机器/联邦学习

SDP差分隐私保护下的分布式机器/联邦学习研究主要集中于全局模型的可用性、隐私性与通信代价之间的平衡性.由表7与表8可知,目前大多数方法采用高斯机制与WRR机制保护敏感模型的隐私.而这两种机制在基于SDP模型为本地梯度添加噪声时,其噪声大小容易受裁剪阈值与隐私预算的影响,也直接影响全局模型的可用性;通信代价容易受梯度维度的影响,维度过大则通信代价过高.针对裁剪阈值问题,有许多工作直接把阈值设置为定值,而启发式定值的问题直接带来全局模型的可用性较低.由于没有充分的先验知识,我们很难设置合适的裁剪阈值.裁剪阈值设置过小导致摈弃梯度的信息过多,偏差过大;而裁剪阈值设置过大会导致本地添加的噪声误差过大.(ε,δ)-CDP与(ε,δ)-LDP模型下已有部分工作研究自适应裁剪阈值

的设置问题[85,86]. 然而,如何在(ε,δ)-SDP模型设置自适应裁剪阈值是个很大挑战. 我们给出的解法是:在(ε,δ)-SDP模型下牺牲部分隐私预算来学习出梯度或者梯度更新的分布. 由于梯度的分布能够较好地反映出梯度的变化,可以利用分布的数字特征(例如分位数、中位数等)来自适应地裁剪梯度或者梯度更新.

目前SDP模型下求解全局模型参数时,很多方法通常利用迭代轮数均分隐私预算来确保每轮更新都受到SDP模型的保护,再利用强组合定理确保整个学习过程满足差分隐私. 这些方法很少考虑到当前轮梯度与历史梯度之间的关联性或者相似性(例如,采用cosine计算梯度之间的相似性). 如果我们能够有效地利用历史梯度或者历史梯度更新,则可以节省部分隐私预算,再把节省的隐私预算用在将来的迭代中去. 其次,目前SDP下分布式机器/联邦学习方法通常假设混洗方是可信的,然而如果混洗方是诚实且好奇或者不可信的,需要借助密码技术实现混洗效果. 在此情况下,连续型高斯机制产生的噪声无法适应于密码技术. 因此我们可以采取离散高斯机制[107]、Skellman机制[108]、二项机制以及泊松-二项机制[109]产生离散噪声来保护本地梯度.

SDP模型下求解全局模型时如何控制通信代价与用户掉线也是其主要挑战. 我们通常采用梯度压缩、量化与稀疏化处理,以及二次抽样技术来控制通信代价. 需要特别指出的是,如果我们采用二次抽样技术压缩梯度时,如何保持抽样技术与混洗技术同时达到隐私放大效果是个大的挑战. 其原因是如果先对用户采样后对用户数据采样,可能造成训练数据的非均匀性,进而无法实现隐私放大效果. 在控制用户掉线方面,我们可以采取随机签入技术允许用户以一定概率加入每轮训练.

9 结论

SDP是一种通用且具有坚实的数学理论支持的隐私保护框架. 针对基于SDP的数据分析与查询方法,本文对最近几年来国内外在对该领域的主要研究工作进行了回顾,总结了SDP保护下二进制/实数域上的聚集查询、直方图查询、范围查询以及分布式机器/联邦学习的研究现状,对各种查询与分析方法进行深入地对比和分析,并指出了当前仍然存在的问题和可能的解决思路. 总体来说,SDP下的数据查询与分析研究仍处于初级阶段,仍然有大量的问题需要做深入的研究.

参考文献

[1] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C]//Theory of Cryptography, Third Theory of Cryptography Conference. Heidelberg, Berlin: Springer, 2006: 265-284.

[2] BEIMEL A, NISSIM K, OMRI E. Distributed private data analysis: Simultaneously solving how and what[M]//Advances in Cryptology - CRYPTO 2008. Berlin, Heidelberg: Springer, 2008: 451-468.

[3] CHAN H H T, SHI E, SONG D. Optimal lower bound for differentially private multi-party aggregation[C]//20th Annual European Symposium on Algorithms. Cham: Springer, 2012: 277-288.

[4] BALLE B, BELL J, CASCÓN A, et al. The privacy blanket of the shuffle model[C]//39th Annual International Cryptology Conference. Cham: Springer, 2019: 638-667.

[5] KAIROUZ P, BONAWITZ K, RAMAGE D. Discrete distribution estimation under local privacy[C]//Proceedings of the 33rd International Conference on Machine Learning - Volume 48. New York: ACM, 2016: 2436-2444.

[6] LUO Q Y, WANG Y L, YI K. Frequency estimation in the shuffle model with almost a single message[C]//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2022: 2219-2232.

[7] DWORK C, ROTH A. The alogorithmic foundations of differential privacy[J]. Foundations and Trends in Theoretical Computer Science, 2014, 9(3/4): 211-407.

[8] ABADI M, CHU A, GOODFELLOW I J, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2016: 308-318.

[9] GOTZ M, MACHANAVAJJHALA A, WANG G Z, et al. Publishing search logs: A comparative study of privacy guarantees[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 24(3): 520-532.

[10] CORMODE G, PROCOPIUC C, SRIVASTAVA D, et al. Differentially private spatial decompositions[C]//Proceedings of the 2012 IEEE 28th International Conference on Data Engineering. New York: ACM, 2012: 20-31.

[11] KELLARIS G, PAPADOPOULOS S, XIAO X K, et al. Differentially private event sequences over infinite streams[J]. Proc. VLDB Endow, 2014, 7(12): 1155-1166.

[12] ERLINGSSON Ú, PIHUR V, KOROLOVA A. RAPPOR: Randomized aggregatable privacy-preserving ordinal response[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2014: 1054-1067.

[13] WANG T H, BLOCKI J, LI N H, et al. Locally differen-

tially private protocols for frequency estimation[C]//Proceedings of the 26th USENIX Conference on Security Symposium. New York: ACM, 2017: 729-745.

[14] BASSILY R, SMITH A D. Local, private, efficient protocols for succinct histograms[C]//Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing. New York: ACM, 2015: 127-135.

[15] Differential Privacy Team: Learning with privacy at scale. Apple, December 2017[EB/OL]. [2025-01-03]. https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html.

[16] CORMODE G, KULKARNI T, SRIVASTAVA D. Marginal release under local differential privacy[C]//Proceedings of the 2018 International Conference on Management of Data. New York: ACM, 2018: 131-146.

[17] DUCHI J C, JORDAN M I, WAINWRIGHT M J. Local privacy and statistical minimax rates[C]//54th Annual IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE, 2013: 429-438.

[18] WANG N, XIAO X K, YANG Y, et al. Collecting and analyzing multidimensional data with local differential privacy[C]//IEEE 35th International Conference on Data Engineering. Piscataway: IEEE, 2019: 638-649.

[19] BITTAU A, ERLINGSSON Ú, MANIATIS P, et al. Prochlo: Strong privacy for analytics in the crowd[C]//Proceedings of the 26th Symposium on Operating Systems Principles. New York: ACM, 2017: 441-459.

[20] MCSHERRY F, TALWAR K. Mechanism design via differential privacy[C]//48th Annual IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE, 2007: 94-103.

[21] MCSHERRY F. Privacy integrated queries: An extensible platform for privacy-preserving data analysis[C]//Proceedings of the 2009 International Conference on Management of Data. New York: ACM, 2009: 19-30.

[22] WARNER S L. Randomized response: A survey technique for eliminating evasive answer bias[J]. Journal of the American Statistical Association, 1965, 60(309): 63-69.

[23] 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.

[24] WU X, LI F G, KUMAR A, et al. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics[C]//Proceedings of the 2017 International Conference on Management of Data. New York: ACM, 2017: 1307-1322.

[25] LI Z T, WANG T H, LOPUHAÄ-ZWAKENBERG M, et al. Estimating numerical distributions under local differential privacy[C]//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2020: 621-635.

[26] STEMMER U. Locally private *k*-means clustering[M]// Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020: 548-559.

[27] CHEU A, SMITH A, ULLMAN J, et al. Distributed differential privacy via shuffling[C]//Advances in Cryptology - EUROCRYPT 2019. New York: ACM, 2019: 375-403.

[28] GHAZI B, PAGH R, VELINGKER A. Scalable and differentially private distributed aggregation in the shuffled model[EB/OL]. (2019-12-02)[2025-01-03]. https://arxiv.org/abs/1906.08320.

[29] GHAZI B, MANURANGSI P, PAGH R, et al. Private aggregation from fewer anonymous messages[C]//39th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham: Springer, 2020: 798-827.

[30] GIRGIS A M, DATA D, DIGGAVI S N, et al. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs[J]. IEEE Journal on Selected Areas in Information Theory, 2024, 2(1): 464-478.

[31] CHEN L J, GHAZI B, KUMAR R, et al. On distributed differential privacy and counting distinct elements[C]// 12th Innovations in Theoretical Computer Science Conference. New York: ACM, 2021: 56:1-56:18.

[32] CHEU A, ULLMAN J R. The limits of pan privacy and shuffle privacy for learning and estimation[C]//53rd Annual ACM SIGACT Symposium on Theory of Computing. New York: ACM, 2021: 1081-1094.

[33] NISSIM K, YAN C. The sample complexity of distribution-free parity learning in the robust shuffle model[J]. Journal of Privacy and Confidentiality, 2022, 12(2). DOI: 10.29012/jpc.805.

[34] EVANS D, KOLESNIKOV V, ROSULEK M. A pragmatic introduction to secure multi-party computation[J]. Foundations and Trends in Privacy and Security, 2018, 2(2/3): 70-246.

[35] FRANKLIN M, YUNG M. Communication complexity of secure computation (extended abstract) [C]//Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing. New York: ACM, 1992: 699-710.

[36] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]//Advances in Cryptology - EUROCRYPT'99. Berlin: Springer, 1999: 223-238.

[37] BALCER V, CHEU A, JOSEPH M, et al. Connecting robust shuffle privacy and pan-privacy[C]//Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms. New York: ACM, 2021: 2384-2403.

[38] WANG T H, XU M, DING B L, et al. Practical and robust privacy amplification with multi-party differential privacy[EB/OL]. (2020-08-02)[2025-01-03]. https://arxiv.org/abs/1908.11515v1.

[39] WANG T H, DING B L, XU M, et al. Improving utility and security of the shuffler-based differential privacy[J]. Proceedings of the VLDB Endowment, 2020, 13(13): 3545-3558.

[40] BALCER V, CHEU A. Separating local & shuffled differential privacy via histograms[C]//1st Conference on Information-Theoretic Cryptography. Cham: Springer, 2020. DOI: 10.4230/LIPIcs.ITC.2020.1.

[41] ERLINGSSON Ú, FELDMAN V, MIRONOV I, et al. Amplification by shuffling: From local to central differential privacy via anonymity[C]//Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019: 2468-2479.

[42] ALLEN J, DING B L, KULKARNI J, et al. An algorithmic framework for differentially private data analysis on trusted processors[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, 2019: 13635-13646.

[43] FELDMAN V, MCMILLAN A, TALWAR K. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling[C]//62nd IEEE Annual Symposium on Foundations of Computer Science. Piscataway: IEEE, 2021: 954-964.

[44] CHEU A, ZHILYAEV M. Differentially private histograms in the shuffle model from fake users[C]//2022 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2022: 440-457.

[45] EBERL M. Fisher-yates shuffle[EB/OL]. (2016-10-05)[2025-01-03]. https://www.wikiwand.com/en/articles/Fisher%E2%80%93Yates_shuffle.

[46] GHAZI B, GOLOWICH N, KUMAR R, et al. On the power of multiple anonymous messages[EB/OL]. (2020-05-19)[2025-01-03]. https://eprint.iacr.org/2019/1382.pdf.

[47] BASSILY R, NISSIM K, STEMMER U, et al. Practical locally private heavy hitters[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 2285-2293.

[48] ACHARYA J, SUN Z T, ZHANG H Y. Hadamard response: Estimating distributions privately, efficiently, and with little communication[C]//The 22nd International Conference on Artificial Intelligence and Statistics. New York: ACM, 2019: 1120-1129.

[49] ACHARYA J SUN Z T. Communication complexity in locally private distribution estimation and heavy hitters[C]//Proceedings of the 36th International Conference on Machine Learning. New York: ACM, 2019: 51-60.

[50] BALLE B, BELL J, GASCÓN A, et al. Private summation in the multi-message shuffle model[C]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2020: 657-676.

[51] BALLE B, BELL J, GASCÓN A, et al. Differentially private summation with multi-message shuffling[EB/OL]. (2019-08-21)[2025-01-03]. https://arxiv.org/abs/1906.09116.

[52] BALLE B, BELL J, GASCÓN A, et al. Improved summation from shuffling[EB/OL]. (2019-09-24) [2025-01-03]. https://arxiv.org/abs/1909.11225.

[53] ERLINGSSON Ú, FELDMAN V, MIRONOV I, et al. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation[EB/OL]. (2019-01-10)[2025-01-03]. https://arxiv.org/abs/2001.03618.

[54] GHAZI B, GOLOWICH N, KUMAR R, et al. Pure differentially private summation from anonymous messages[C]//1st Conference on Information-Theoretic Cryptography. Cham: Springer, 2022. DOI: 10.4230/LIPIcs.ITC.2020.15.

[55] GHAZI B KUMAR R, MANURANGSI P, et al. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead[C]//Proceedings of the 37th International Conference on Machine Learning. New York: ACM, 2020: 3505-3514.

[56] ISHAI Y, KUSHILEVITZ E, OSTROVSKY R, et al. Cryptography from anonymity[C]//2006 47th Annual IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE, 2006: 239-248.

[57] CHEU A. Differential privacy in the shuffle model: A survey of separations[EB/OL]. (2022-05-24) [2025-01-03]. https://arxiv.org/abs/2107.11839.

[58] XU S Q, ZHENG Y F, HUA Z Y. Camel: Communication-efficient and maliciously secure federated learning in the shuffle model of differential privacy[C]//Proceedings of the 2024 on ACM SIGSAC Conference on Computer

and Communications Security. New York: ACM, 2024: 243-257.

[59] GHAZI B, KUMAR R, MANURANGSI P, et al. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message[C]//Proceedings of the 39th International Conference on Machine Learning. New York: ACM, 2023: 3692-3701.

[60] GHAZI B, GOLOWICH N, KUMAR R, et al. On the power of multiple anonymous messages: Frequency estimation andSelection in the shuffle model of Differential privacy[C]//Advances in Cryptology - EUROCRYPT 2021. Cham: Springer, 2021: 463-488.

[61] GORMODE G, MUTHUKRISHNAN S. An improved data stream summary: The count-min sketch and its applications[J]. Journal of Algorithms, 2005, 55(1): 58-75.

[62] LI X C, LIU W R, CHEN Z Y, et al. DUMP: A dummy-point-based framework for histogram estimation in shuffle model[EB/OL]. (2021-07-31)[2025-01-03]. https://arxiv.org/abs/2009.13738v1.

[63] VADHAN S. The complexity of differential privacy[M]// Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich. Cham: Springer International Publishing, 2017: 347-450.

[64] BOTTOU L. Large-scale machine learning with stochastic gradient descent[C]//Proceedings of COMPSTAT'2010. Heidelberg: Springer, 2010: 177-186.

[65] CHAUDHURI K, MONTELEONI C, SARWATE A D. Differentially private empirical risk minimization[J]. Journal of Machine Learning Research, 2011, 12: 1069-1109.

[66] BASSILY R, SMITH A, THAKURTA A. Private empirical risk minimization: Efficient algorithms and tight error bounds[C]//2014 IEEE 55th Annual Symposium on Foundations of Computer Science. Piscataway: IEEE, 2014: 464-473.

[67] AGARWAL N, SURESH A T, YU F X, et al. CpSGD: Communication-efficient and differentially-private distributed SGD[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: ACM, 2018: 7575-7586.

[68] SURESH A T, YU F X, SANJIV K H, et al. Distributed mean estimation with limited communication[C]//Proceedings of the 34th International Conference on Machine Learning. New York: ACM, 2017: 3329-3337.

[69] MCMAHAN H B, RAMAGE D, TALWAR K, et al. Learning differentially private recurrent language models[C]//International Conference on Learning Representations. ICLR, 2017. https://openreview.net/forum?id=BJ0hF1Z0b.

[70] LOWY A, RAZAVIYAYN M. Locally differentially private federated learning: Efficient algorithms with tight risk bounds[EB/OL]. (2024-11-24) [2025-01-03]. https://arxiv.org/abs/2106.09779v1.

[71] DUCHI J C, JORDAN M I, WAINWRIGHT M J. Minimax optimal procedures for locally private estimation[J]. Journal of the American Statistical Association, 2018, 113(521): 182-201.

[72] YANG Q, LIU Y, CHEN T J, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1-19.

[73] CHEU A, JOSEPH M, MAO J M, et al. Shuffle private stochastic convex optimization[C]//10th International Conference on Learning Representations. New York: ACM, 2022: 209.

[74] GIRGIS A M, DATA D, DIGGAVI S N. Rényi differential privacy of the subsampled shuffle model in distributed learning[C]//Proceedings of the 35th International Conference on Neural Information Processing Systems. New York: ACM, 2021: 29181-29192.

[75] MIRONOV I. Rényi differential privacy[C]//30th IEEE Computer Security Foundations Symposium. Piscataway: IEEE, 2017: 263-275.

[76] GIRGIS A M, DATA D, DIGGAVI S, et al. On the rényi differential privacy of the shuffle model[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2021: 2321-2341.

[77] LIEW S P, HASEGAWA S, TAKAHASHI T. Shuffled check-in: Privacy amplification towards practical distributed learning[EB/OL]. (2023-07-04)[2025-01-03]. https://arxiv.org/abs/2206.03151v1.

[78] BASSILY R, FELDMAN V, TALWAR K, et al. Private stochastic convex optimization with optimal rates[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, 2019: 11279-11288.

[79] ZHU L G, LIU Z J, HAN S. Deep leakage from gradients[C]// Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, 2019: 14747-14756.

[80] WEI W Q, LIU L. Gradient leakage attack resilient deep learning[J]. IEEE Transactions on Information Forensics and Security, 2021, 17: 303-316.

[81] GEYER R C, KLEIN T, NABI M. Differentially pri-

vate federated learning: A client level perspective[EB/OL]. (2018-03-01) [2025-01-03] https://arxiv.org/abs/1712.07557.

[82] TRUEX S, LIU L, CHOW K H, et al. LDP-Fed: federated learning with local differential privacy[C]//Proceedings of the 3rd International Workshop on Edge Systems, Analytics and Networking. New York: ACM, 2020: 61-66.

[83] WANG L, JIA R X, SONG D. D2P-Fed: Differentially private federated learning with efficient communication[EB/OL]. (2021-01-02)[2025-01-03]. https://arxiv.org/abs/2006.13039.

[84] WEI K, LI J, DING M, et al. User-level privacy-preserving federated learning: Analysis and performance optimization[J]. IEEE Transactions on Mobile Computing, 2022, 21(9): 3388-3401.

[85] ANDREW G, THAKKAR O, MCMAHAN B, et al. Differentially private learning with adaptive clipping[C]//Proceedings of the 35th International Conference on Neural Information Processing Systems. New York: ACM, 2021: 17455-17466.

[86] ZHANG X W, CHEN X Y, HONG M Y, et al. Understanding clipping for federated learning: Convergence and client-level differential privacy[C]//Proceedings of the 39th International Conference on Machine Learning. New York: ACM, 2022: 26048-26067.

[87] SUN L C, QIAN J W, CHEN X. LDP-FL: Practical private aggregation in federated learning with local differential privacy[C]//Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. Cham: Springer, 2021: 1571-1578.

[88] LIU R X, CAO Y, CHEN H, et al. FLAME: Differentially private federated learning in the shuffle model[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(10): 8688-8696.

[89] SCOTT M, CORMODE G, MAPLE C. Aggregation and transformation of vector-valued messages in the shuffle model of differential privacy[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 612-627.

[90] 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. New York: ACM, 2010: 735-746.

[91] GIRGIS A, DATA D, DIGGAVI S, et al. Shuffled model of differential privacy in federated learning[C]//Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. PMLR, 2021: 2521-2529.

[92] WANG S W, PENG Y, LI J, et al. Privacy amplification via shuffling: Unified, simplified, and tightened[J]. Proceedings of the VLDB Endowment, 2024, 17(8): 1870-1883.

[93] GIRGIS A M, DATA D, DIGGAVI S. Differentially private federated learning with shuffling and client self-sampling[C]//2021 IEEE International Symposium on Information Theory. Piscataway: IEEE, 2021: 338-343.

[94] FELDMAN V, MCMILLAN A, TALWAR K. Stronger privacy amplification by shuffling for rényi and approximate differential privacy[EB/OL]. (2023-10-30) [2025-01-03]. https://arxiv.org/abs/2208.04591.

[95] CHEN W N, ÖZGÜR A, KAIROUZ P. The poisson binomial mechanism for unbiased federated learning with secure aggregation[C]//Proceedings of the 41st International Conference on Machine Learning. New York: ACM, 2024: 3490-3506.

[96] BALLE B, KAIROUZ P, MCMAHAN B, et al. Privacy amplification via random check-ins[C]//Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, 2020: 23-31.

[97] BALLE B, BARTHE G, GABOARDI M. Privacy amplification by subsampling: Tight analyses via couplings and divergences[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: ACM, 2018: 6280-6290.

[98] CORMODE G, KULKARNI T, SRIVASTAVA D. Answering range queries under local differential privacy[J]. Proceedings of the VLDB Endowment, 2019, 12(10): 1126-1138.

[99] WANG T H, DING B L, ZHOU J R, et al. Answering multi-dimensional analytical queries under local differential privacy[C]//Proceedings of the 2019 International Conference on Management of Data. New York: ACM, 2019: 159-176.

[100] XU M, DING B L, WANG T H, et al. Collecting and analyzing data jointly from multiple services under local differential privacy[J]. Proceedings of the VLDB Endowment, 2020, 13(12): 2760-2772.

[101] GHAZI B, COLOWICH N, KUMAR R, et al. Private heavy hitters and range queries in the shuffled model[EB/OL]. (2020-08-15)[2025-01-03]. https://dblp.org/rec/journals/corr/abs-1908-11358.html.

[102] KARWA V, RASKHODNIKOVA S, SMITH A, et al. Private analysis of graph structure[J]. Proceedings of the

VLDB Endowment, 2011, 4(11): 1146-1157.
[103] DAY W Y, LI N H, LYU M. Publishing graph degree distribution with node differential privacy[C]//Proceedings of the 2016 International Conference on Management of Data. New York: ACM, 2016: 123-138.
[104] YE Q Q, HU H B, AU M H, et al. LF-GDPR: A framework for estimating graph metrics with local differential privacy[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(10): 4905-4920.
[105] IMOLA J, MURAHAMI T, CHAUDHURI K. Locally differentially private analysis of graph statistics[C]//Proceedings of the 30th USENIX Security Symposium. Berkeley: USENIX, 2021: 983-1000.
[106] IMOLA J, MURAKAMI T, CHAUDHURI K. Differentially private triangle and 4-cycle counting in the shuffle model[C]//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2022: 1505-1519.
[107] KAIROUZ P LIU Z Y, STEINKE T. The distributed discrete gaussian mechanism for federated learning with secure aggregation[C]//Proceedings of the 40th International Conference on Machine Learning. New York: ACM, 2023: 5201-5212.
[108] AGARWAL N, KAIROUZ P, LIU Z Y. The skellam mechanism for differentially private federated learning[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: ACM, 2022: 5052-5064.
[109] WANG S W, YU S Y, REN X J, et al. Differentially private numerical vector analyses in the local and shuffle model[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(1): 1-15.

作者简介

张啸剑 男,1980年9月出生于河南省周口市. 现为河南财经政法大学计算机与信息工程学院教授、硕士生导师. 主要研究方向为数据安全与隐私、差分隐私、数据库,以及图数据管理等. 中国电子学会会员编号:E190087355M.
E-mail: xjzhang82@alu.ruc.edu.cn

王浩锋 男,1999年9月出生于河南省许昌市. 现为河南财经政法大学计算机与信息工程学院硕士研究生. 主要研究方向为数据安全与隐私、差分隐私.
E-mail: 13140075039@163.com

傅继彬 男,1975年9月出生于河南省许昌市. 现为河南财经政法大学计算机与信息工程学院副教授、硕士生导师. 主要研究方向为机器学习、数据安全与隐私保护.
E-mail: fujibin@huel.edu.cn