

基于本地化差分隐私的数据立方体发布方法

张啸剑^{1,2}, 王浩锋¹, 曹小杰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: 基于本地化差分隐私的数据立方体发布已经得到研究者的广泛关注。针对现有方法在数据立方体发布过程中未能完全利用方体格结构的问题, 提出了有效且满足本地差分隐私的数据立方体发布策略 Full-Pub、Base-Pub 和 Partial-Pub。这3种策略利用格结构自身蕴含的偏序关系进行方体发布。为了有效利用方体格结构的特性, Full-Pub、Base-Pub 和 Partial-Pub 分别从全方体、基本方体和部分方体的角度发布方体。其中 Partial-Pub 是在用户分组策略的基础上, 利用贪心算法与集合覆盖技术在整个方体集合中寻找部分发布方体。Full-Pub、Base-Pub 和 Partial-Pub 中的每个用户首先把自身数据进行本地编码, 然后利用 Cuboid-OUE、Cuboid-HHT 和 Cuboid-CMS 对编码数据进行本地扰动, 最后汇报给收集者。收集者结合每个用户的报告值重构噪音方体进行发布。在真实与合成数据集上进行实验, 结果表明所提的数据立方体发布方法优于现有数据立方体发布方法。

关键词: 本地差分隐私; 数据立方体; 格结构; 方体发布; 随机应答机制

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2025048

Data cubes publication under local differential privacy

ZHANG Xiaojian^{1,2}, WANG Haofeng¹, CAO Xiaojie¹, FU Jibin¹

1. College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450046, China

2. Engineering Technology Research Center of Henan Province IoT Big Data Processing and Security, Zhengzhou 450121, China

Abstract: The release of data cubes based on local differential privacy has been widely studied by researchers. To address the problem that existing methods failed to fully utilize the lattice structure of cuboids during data cube publication, three efficient data cube publishing methods—Full-Pub, Base-Pub, and Partial-Pub—were proposed in this paper. All three algorithms made use of the partial order relationships among cuboids within the lattice struc-

收稿日期: 2025-01-19; 修回日期: 2025-04-22

通信作者: 张啸剑, xjzhang82@alu.ruc.edu.cn

基金项目: 国家自然科学基金(62072156); 河南省自然科学杰出青年科学基金(252300421061); 河南省重点科研项目计划基础研究专项(25ZX012)

Foundation Items: The National Natural Science Foundation of China (62072156), The Natural Science Outstanding Youth Science Foundation Project of Henan Province (252300421061), The Basic Research Special Projects of Key Research Projects in Higher Education Institutes in Henan Province (25ZX012)

引用格式: 张啸剑, 王浩锋, 曹小杰, 等. 基于本地化差分隐私的数据立方体发布方法[J]. 网络与信息安全学报, 2025, 11(4): 141-159.

Citation Format: ZHANG X J, WANG H F, CAO X J, et al. Data cubes publication under local differential privacy[J]. Chinese Journal of Network and Information Security, 2025, 11(4): 141-159.

ture for data cube publication. To effectively leverage the characteristics of the cuboid lattice, Full-Pub, Base-Pub, and Partial-Pub published the cuboids from the perspectives of the entire data cube, the base cuboid, and a part of the data cube, respectively. In particular, based on a user-grouping strategy, the Partial-Pub method integrated a greedy algorithm and set coverage to identify a subset of all cuboids. In Full-Pub, Base-Pub, and Partial-Pub, each user's tuple was first encoded, and then perturbed using Cuboid-OUE, Cuboid-HHT, and Cuboid-CMS, respectively. Based on the reports collected from all users, the noise cuboids were reconstructed by the collector to answer OLAP queries. Extensive experiments were conducted on multiple real-world and synthetic datasets, and the results demonstrated the superiority of the proposed solutions.

Keywords: local differential privacy, data cube, lattice structure, cube release, randomized response

0 引言

人工智能与大数据技术的迅速发展,使得第三方收集与分析个人数据(例如年龄、收入、喜好等)变得尤为容易。第三方收集者利用多维数据模型对个人数据进行重构与建模可以高效响应聚集查询与联机分析处理(online analytical processing, OLAP)。多维数据模型通常把数据看作数据立方体,收集者从多个维度对个人数据进行建模与观察。数据立方体由外键约束下的维表与事实表定义。例如,图1(a)所示的事实表包含 A 、 B 、 C 3个维度。结合这3个维度可以形成23种不同维度组合的方体。每个方体表示维度不同等级的汇总。这些不同维度的方体形成格结构,进

而形成数据立方体。结合数据立方体可以响应如上卷、下钻、切块等OLAP操作。

然而,直接收集事实表中个人多维数据并构建数据立方体来响应OLAP查询,与个人关联的敏感信息很有可能被泄露。例如,图1(d)中的3维方体 $\{A, B, C\}$, $A \in [1 \sim 20]$, $B \in [3\ 000 \sim 5\ 000]$ 且属性 $C(C=F)$ 的用户个数为2。若攻击者已知Alice的年龄为20且为女性,则可推断出Alice的属性 B 在3 000~5 000内。造成个人敏感信息泄露的主要原因是数据收集过程中,用户无法掌控自己的隐私数据。本地化差分隐私^[1-2](local differential privacy, LDP)技术的出现能够帮助用户编码与扰动自身隐私数据,进而有效解决了收集过程

Uid	A	B	C
u_1	1~20	1 000~3 000	M
u_2	1~20	3 000~5 000	F
u_3	21~40	7 000~9 000	M
u_4	41~60	5 000~7 000	F
u_5	60+	9 000+	M
u_6	41~60	5 000~7 000	F
u_7	21~40	5 000~7 000	M
u_8	21~40	7 000~9 000	M
u_9	1~20	3 000~5 000	F
u_{10}	21~40	7 000~9 000	F

(a) 事实表 T

A	B	C	c
1~20	*	F	2
1~20	*	M	1
21~40	*	F	1
...	*	...	...

(c) 2维方体 $\{A, C\}$

A	B	C	c
1~20	*	*	3
21~40	*	*	4
41~60	*	*	2
60+	*	*	1

(b) 1维方体 $\{A\}$

A	B	C	c
1~20	1 000~3 000	F	0
1~20	3 000~5 000	F	0
1~20	5 000~7 000	F	0
1~20	7 000~9 000	F	0
1~20	9 000+	F	0
21~40	1 000~3 000	F	0
21~40	3 000~5 000	F	0
21~40	5 000~7 000	F	0
21~40	7 000~9 000	F	0
...	...	...	...

(d) 3维方体 $\{A, B, C\}$

图1 事实表与基于事实表的数据立方体

Figure 1 Fact table and data cube based on the fact table

中隐私泄露问题。目前基于LDP响应多维属性查询的方法主要包括InputHT (input Hashing technique)^[3]、CALM (consistent adaptive local marginal)^[4]等。InputHT采用构建满足LDP的全列联表响应 k -way边缘查询。该方法采用哈达玛技术转换原始属性值,再利用RR (randomized response) 机制^[5,6]扰动转换结果,然而该方法的查询误差随着维度个数 d 与边缘大小 k 的增加而增加。CALM采用用户分组策略与覆盖集技术响应 k -way边缘查询。该方法根据 d 与 k 的大小,选择 w 个大小为 l 的边缘表,再自适应地选择GRR (generalized randomized response)^[7]或OUE (optimized unary encoding)^[8]机制扰动边缘表,响应任何 k -way边缘表的查询,然而该方法的查询误差随着 d 与属性值域的增加而增加。上述方法均是基于响应 k -way边缘表查询为目标,边缘表只是属性之间不同组合方式,未考虑数据立方体本身所蕴含的偏序关系,无法直接应用于立方体发布和OLAP查询。数据立方体发布过程中存在的挑战包括:所发布方体如何能响应任何外部方体查询;如何利用集合覆盖技术与格结构寻找部分方体来响应任何方体查询。基于此,本文基于LDP技术提出了一种有效算法LDPCUBE来响应任何方体查询与OLAP操作。

本文主要贡献如下:

1) 从全方体、基本方体和部分方体的角度设计了3种数据立方体发布策略Full-Pub、Base-Pub和Partial-Pub。在这3种发布策略中,分别基于最优一元编码、哈达玛转化编码以及计数均值草图(count mean sketch, CMS)结构设计了3种扰动机制Cuboid-OUE、Cuboid-HHT和Cuboid-CMS。

2) Partial-Pub发布策略利用方体格结构本身蕴含的偏斜关系搜索部分方体进行发布。当事实表维度 d 很大时,在 2^d 个方体中搜索部分方体为一个NP问题。因此,Partial-Pub利用集合覆盖技术与方体格结构蕴含的偏斜关系搜索部分方体进行发布。

3) 理论分析了所提出算法满足 ϵ -LDP,以及响应方体查询的误差边界。通过真实数据实验分析,本文所设计的数据立方体发布算法具有较高的可用性和查询准确性。

1 相关工作

目前基于中心化差分隐私保护的数据立方体发布已经得到了广泛关注。文献[9]是中心化差分隐私(center differential privacy, CDP)^[10-12]下数据立方体发布的早期代表。该方法通过在单元格度量值上添加拉普拉斯^[12]噪音发布方体,并依赖噪音方体响应OLAP查询。然而,该方法无法直接应用于LDP环境,其原因是文献[9]中的算法在LDP环境下发布误差很大。PriView^[13]基于CDP利用覆盖技术寻找大小固定的边缘表,并结合拉普拉斯机制^[12]响应 k -way边缘表查询。PriView假设属性之间彼此独立,试图寻找所有边缘表。然而,当维度大于12时,该算法生成的噪音会变得很大,从而影响查询的准确性。与PriView不同,PrivSyn^[14]结合了满足CDP的高斯机制计算出任意两个依赖属性之间的差异分值。利用属性差异分值与团合并技术寻找高可用的边缘表进行发布。上述算法均假设收集者可信,无法直接应用于LDP环境。

基于LDP的边缘表与立方体发布同样得到广泛的关注。文献[15]把LDP下边缘表发布方分为直接发布与转换发布。InputHT对用户元组数据进行编码,利用RR机制对哈达玛转换^[16-18]结果进行本地扰动,然而该算法只能处理二进制属性,处理非二进制属性时维度会增加。此外,响应 k -way边缘表查询的误差会随着维度增加而增大。CALM针对类别属性利用扰动误差、重构误差与采样误差之间的平衡关系直接寻找 w 个大小为 l 的边缘表集合,并响应 k -way边缘表查询。然而,该方法的误差随着维度 d 与值域 m 的增加而增加。尽管目前在LDP下有多种响应 k -way边缘表的方法,然而边缘表只是属性之前的组合形式,这些方法未能考虑到数据立方体的格结构本身的特性。因此,针对上述方法存在的不足,本文提出了基于方体格结构的数据立方体发布算法LDPCUBE,该算法从不同角度对数据立方体进行发布,具有较高的发布精度。

2 定义与概念

2.1 本地化差分隐私

LDP技术允许每个用户本地编码与本地扰动

自身数据后发送给数据收集者, 实现隐私不被泄露。收集者通过聚集用户的报告值来响应查询, LDP 技术的形式化定义如下。

定义 1 ε -LDP。设 v 与 v' 为值域 D 中任意两个值, 给定随机算法 M , $\mathcal{O}$ 为 M 的输出域, 若 M 在 v 与 v' 上得到相同输出结果 $o(o \in \mathcal{O})$ 的概率满足

$$\Pr[M(v) = o] \leq \exp(\varepsilon) \times \Pr[M(v') = o] \quad (1)$$

则 M 满足 ε -LDP。其中, ε 为隐私预算, ε 越小, M 的隐私保护程度越高。

2.2 随机应答机制

由 Wanner 提出的 RR 机制是实现本地隐私保护的常用技术。该机制思想是用户在响应布尔敏感问题时, 以 p 的概率真实回答, 以 $1-p$ 的概率给出错误的回答。这种随机性使得攻击者无法确定用户的真实值。SUE 机制^[8]与 OUE 机制是 RR 机制的拓展。

定义 2 本地扰动机制 UE。给定值 v_i 且 $v_i \in D$, 将值 v_i 编码成长度为 $|D|=m$ 的 0-1 向量 B , 且 B 中仅第 v_i 位置为 1。则 UE 机制可以表示为:

$$\Pr[B[i] = 1] = \begin{cases} p, & B[i] = 1 \\ q, & B[i] = 0 \end{cases} \quad (2)$$

结合 $p+q=1$, UE 机制变成了对称一元编码 SUE 机制, 可以表示为:

$$\Pr[B[i] = 1] = \begin{cases} p = \frac{e^{e/2}}{e^{e/2} + 1}, & B[i] = 1 \\ q = \frac{1}{e^{e/2} + 1}, & B[i] = 0 \end{cases} \quad (3)$$

原始值经过编码后变成长度为 m 的向量, 且有 $m-1$ 位值为 0。 m 很大时导致 0 值过多, 若 0 位分配全部 ε , 会使得方差最小。此时 UE 机制变成 OUE 机制, 可以表示为:

$$\Pr[B[i] = 1] = \begin{cases} p = \frac{1}{2}, & B[i] = 1 \\ q = \frac{1}{e^\varepsilon + 1}, & B[i] = 0 \end{cases} \quad (4)$$

定义 3 哈达玛转换^[18]。 $\theta = H_l t$ 为向量 $t \in \mathbb{R}^l$ 经过哈达玛矩阵转换后的表示, H_l 为 $l \times l$ 的对称矩阵, 且 $H_l \in \left\{ -1/\sqrt{l}, 1/\sqrt{l} \right\}^{l \times l}$ 。

2.3 数据立方体

给定包含 d 个维度 $\{A_1, A_2, \dots, A_d\}$ 的事实表 T , 每个维度的值域大小为 c_i 。 n 个用户 $U = \{u_1, u_2, \dots, u_n\}$ 。事实表 T 的数据立方体由方体格组成, 而每个方格是由单元格 a 与度量值构成。单元格 a 可以表示为 $(a[1], a[2], \dots, a[d])$, 其中 $a[i] \in (A_i \cup \{*\})$ 表示单元格 a 在第 i 个维度中的取值。单元格 a 的度量值表示为 $c(a)$, 通常为事实表 T 内聚集在 a 中的元组个数。 k 维方体 C 可以看作事实表 T 在维度集合 $[C]$ 上的投影。 d 维方体称为基本方体。 L 表示所有方体的集合, 则 $(L, \leq)$ 形成方体的格, 其中 $\leq$ 表示偏序关系。给定两个方体 C_1 与 C_2 , $[C_1]$ 与 $[C_2]$ 分别表示二者的维度集合。如果 $[C_1] \subseteq [C_2]$, 可表示为 $C_1 \leq C_2$, 则 C_1 为 C_2 的祖先, C_2 为 C_1 的后代。例如, 图 1(a) 事实表 T 包含 $A: \{1 \sim 20, 21 \sim 40, 41 \sim 60, 60+\}$, $B: \{1\ 000 \sim 3\ 000, 3\ 000 \sim 5\ 000, 5\ 000 \sim 7\ 000, 7\ 000 \sim 9\ 000, 9\ 000+\}$, $C: \{F, M\}$ 3 个维度。事实表 T 的格结构如图 2(b) 所示。最底层为基本方体, 自底向上依次为 3 维方体、2 维方体、1 维方体、0 维方体 (也称为 all 方体)。例如, 2 维方体 $\{A, B\}$ 所对应的表示如图 2(a) 所示。由图 2 可知方体 $\{A\} \subseteq \{A, C\}$, 则 $\{A\}$ 是 $\{A, C\}$ 的祖先, 即 $\{A\} \leq \{A, C\}$ 。

2.4 问题描述

给定查询 Q , $P(Q)$ 与 $P'(Q)$ 分别表示查询 Q 的真实结果与噪音结果。本文要解决的问题是如何在 LDP 下收集事实表中的用户数据, 结合用户噪音数据如何全部或者部分发布数据立方体, 且能够获得高可用的响应结果 $P'(Q)$ 。采用均方误差度量查询 Q 的可用性, 如式(5)所示:

$$\text{MSE}(P(Q)) = \mathbb{E} \left[(P(Q) - P'(Q))^2 \right] \quad (5)$$

3 数据立方体发布方法

在设计满足 LDP 的数据立方体发布算法时需考虑 3 条原则: ①所发布方体能够响应任何方体查询; ②设计发布算法时, 要充分利用方体格结构特征; ③发布算法尽可能返回高精度的查询结果。针对上述 3 种原则, 本文设计了数据立方体发布算法 LDPCUBE, 具体描述见算法 1。

算法 1 LDPCUBE

输入 用户的元组数据 $t_i = (t_i[A_1], t_i[A_2], \dots, t_i[A_d])$

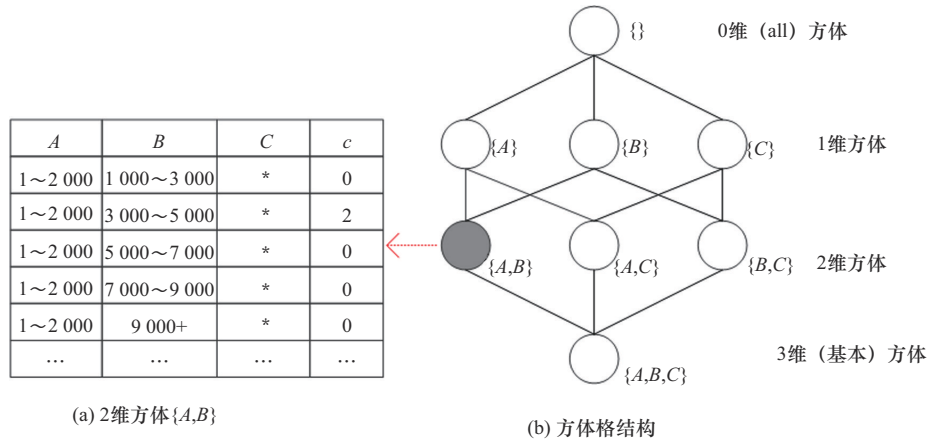


图2 数据立方体的格结构

Figure 2 Lattice structure of the data cube

$[A_d])$, 隐私预算 ε , 由格结构控制的方体集合 L

输出 噪音方体集 $\tilde{L}_{\text{pub}}$

- 1) $L_{\text{pub}} \leftarrow \text{Publication-Mechanism}(L)$;
- 2) **for each** user u_i **do**
- 3) u_i 选择 $C \in L_{\text{pub}}$ 且 $[C] = \{A_{i_1}, A_{i_2}, \dots, A_{i_m}\}$;
- 4) $y_i \leftarrow R_{\text{client}}(t_i, C, \varepsilon)$; //本地扰动
- 5) u_i 发送 y_i 给收集端;
- 6) **end for**
- 7) $\tilde{L}_{\text{pub}} \leftarrow \emptyset$; //存放噪音方体
- 8) **for each** cuboid $C \in L_{\text{pub}}$ **do**
- 9) 收集端根据报告值 y_i 重构噪音方体 $\tilde{C}$;
- 10) $\tilde{L}_{\text{pub}} \leftarrow \tilde{L}_{\text{pub}} \cup \tilde{C}$;
- 11) **end for**
- 12) **return** $\tilde{L}_{\text{pub}}$. //噪音方体集合

LDPCUBE 首先根据数据立方体发布方体策略 Publication-Mechanism 得到发布方体集合 L_{pub} 。用户 u_i 从发布方体集合 L_{pub} 中选择一个方体 C , 利用本地扰动机制 R_{client} 保护隐私数据 (步骤2至步骤6)。初始化空集用于存放噪音方体 (步骤7)。收集端收集到 n 个用户的噪音数据, 计算方体集合 L_{pub} 中的每一个噪音方体。收集端将方体集合 $\tilde{L}_{\text{pub}}$ 进行发布 (步骤9至步骤11)。

3.1 Publication-Mechanism 策略的实现

由算法1可知, Publication-Mechanism 策略与本地扰动算法 R_{client} 是 LDPCUBE 的核心步骤。为了实现 Publication-Mechanism, 本文从全方位、基本方体与部分方体的角度设计了 Full-Pub、Base-Pub 和 Partial-Pub 3 种 L_{pub} 的生成策略。

(1) Full-Pub 全方位 L_{pub} 生成策略

Full-Pub 策略从发布全方位的角度生成集合 L_{pub} 。该策略首先计算出全部方体并构建出格结构。若事实表中包含 d 个维度, 则方体总数为 2^d 。设由 2^d 个方体组成的集合为 L , 方体间通过偏序关系组成方体的格。格结构中每个方体蕴含着不同属性组合的聚集值。Full-Pub 策略采用格结构对所有用户进行分组。首先根据格的层次结构进行第1次用户分组, 然后再根据每一层方体个数进行第2次分组。由于0维方体不包含任何属性值, 无须利用 LDP 策略保护, 该方体不参与用户分组。例如, 图2(b)中的格结构把 n 个用户按照3维、2维、1维层次分为3组, 每层用户个数为 $n/3$ 。而每层用户再根据该层的方体个数进行随机分组。例如, 图2(b)1维层次中包含3个方体, 随机分到 $\{A\}$ 方体中的用户为 $n/9$ 。分到方体中的用户采用 $R_{\text{client}}(\cdot)$ 对自身数据进行本地扰动, 再把扰动值报告给收集者。

(2) Base-Pub 基本方体 L_{pub} 生成策略

Base-Pub 策略从发布基本方体的角度生成 L_{pub} 。基本方体为 d 维方体, 是所有方体的后代。因此, 可从基本方体计算出任何 $0, 1, \dots, d-1$ 维方体。该策略的思想是每个用户把 d 个维度的取值作为一个整体, 利用 $R_{\text{client}}(\cdot)$ 扰动后发送给收集者, 收集者只构建基本方体并将其发布。如图2(b)中3维方体为基本方体。收集者根据 n 个用户的报告值构建噪音基本方体 $\{A, B, C\}$ 即可。

由上述分析可知, Full-Pub 和 Base-Pub 是方

体发布的两个极端策略，二者造成的发布误差不同。给定格结构中任意方体 C ，该方体包含的属性为 $[C]$ 。设 Var_{cell} 为 $R_{\text{client}}(\cdot)$ 扰动 1 个单元格产生的误差，利用 Full-Pub 策略发布方体 C 产生的误差如式(6)所示。

$$\text{err}_{\text{Full}}(C) = n(C) \times \text{Var}_{\text{cell}} \times \prod_{A_i \in [C]} |A_i| \times g \quad (6)$$

其中， $n(C)$ 表示方体 C 中用户个数， $|A_i|$ 表示属性 A_i 的值域大小， g 表示用户分组数。

利用 Base-Pub 策略发布方体 C 产生的误差如式(7)所示。

$$\text{err}_{\text{Base}}(C) = n \times \text{Var}_{\text{cell}} \times \prod_{A_i \in [C]} |A_i| \quad (7)$$

例如，假设图 2(a) 中 $\{A, B, C\}$ 3 个维度的值域大小分别为 4、5、2。利用 Full-Pub 策略发布方体 $\{B\}$ 所产生的误差为： $n/9 \times \text{Var}_{\text{cell}} \times 5 \times 9 = 5n \times \text{Var}_{\text{cell}}$ 。利用 Base-Pub 策略发布基本方体 $\{A, B, C\}$ ，每个单元格的误差为 $n \times \text{Var}_{\text{cell}}$ 。则方体 $\{B\}$ 的误差为 $n \times \text{Var}_{\text{cell}} \times 8 \times 5 = 40n \times \text{Var}_{\text{cell}}$ 。

尽管 Full-Pub 策略发布方体越多查询性能越好，但是方体越多，导致分配到每个方体的用户个数会越少，进而造成查询精度较低；Base-Pub 策略把 n 个用户分配到基本方体，能够准确计算基本方体的聚集值。然而从基本方体推导出其他方体时，误差会累加。

(3) Partial-Pub 部分方体 L_{pub} 生成策略

由于格结构蕴含方体之间的偏斜关系，若能找到那些蕴含全部方体的部分方体进行发布，则发布误差会大幅度降低。而如何在整个格结构中寻找覆盖全部方体的部分方体是 NP 问题。基于此，利用集合覆盖技术提出了 Partial-Pub 策略寻找部分方体。该策略的思想是利用 βn 个用户寻找 L_{pub} ，剩余用户响应外部查询。利用 Partial-Pub 发布方体 C 产生的误差如式(8)所示。

$$\text{err}_{\text{Partial}}(C) = \frac{(1 - \beta)n}{|L_{\text{part}}|} \times \text{Var}_{\text{cell}} \times \prod_{A_i \in [C]} |A_i| \quad (8)$$

例如，结合图 2(b) 中的格结构，利用 $\beta=20\%$ 的用户寻找 $L_{\text{pub}} = \{\{A, B, C\}, \{A, C\}, \{B, C\}, \{A\}\}$ 。剩余 80% 用户按照方体个数分组来响应查询。发布部分方体集合中每个方体单元格的误差为 $0.8 \times n/4 \times \text{Var}_{\text{cell}}$ 。部分方体之外的方体均可由 L_{pub} 集合推导出。例如，由于 $\{B\} \leq \{B, C\}$ 、 $\{B\} \leq \{A, B,$

$C\}$ ，则方体 $\{B\}$ 可以由误差最小的方体 $\{B, C\}$ 生成。此时方体 $\{B\}$ 由方体 $\{B, C\}$ 生成带来的误差为 $0.8 \times n/4 \times \text{Var}_{\text{cell}} \times 5 \times 2 = 2n \times \text{Var}_{\text{cell}}$ 。

Partial-Pub 算法具体描述见算法 2。

算法 2 Partial-Pub 算法

输入 用户的数据 $t_i = (t_i[A_1], t_i[A_2], \dots, t_i[A_d])$ ，
全方体 L ，隐私预算 ε

输出 噪音方体集合 $\tilde{L}_{\text{pub}}$

1) **for each** user u_i **do** // βn 个用户

2) u_i 随机选择格结构中任意一层；

3) 在该层选择方体 $C \in L[C]$ ， $L[C] = \{A_{i_1}, A_{i_2}, \dots, A_{i_m}\}$ ；

4) $y_i \leftarrow R_{\text{client}}(t_i, C, \varepsilon)$ ；

5) u_i 发送 y_i 给收集端；

6) **end for**

7) 收集端计算 $\{A_1, A_2, \dots, A_d\}$ 的值域笛卡儿积

$D = \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_d)$ ；

8) $\theta_{\text{left}} \leftarrow 0$ ， $\theta_{\text{right}} \leftarrow \text{Var}_{\text{base}} \times |D|$ ；

9) **while** $|\theta_{\text{right}} - \theta_{\text{left}}| > 1$ **do**

10) $\theta = (\theta_{\text{right}} + \theta_{\text{left}})/2$ ；

11) **for each** $S=1, 2, \dots, |L|$ **do**

12) **if** Search-Partial(L, θ, S) = No **then**

13) $\theta_{\text{left}} \leftarrow \theta$ ；

14) **else**

15) $\theta_{\text{right}} \leftarrow \theta$ ；

16) **end for**

17) 收集端计算得到 θ 与 L_{pub} ；

18) **for each** user u_i **do** // $(1-\beta)n$ 个用户

19) u_i 选择方体 $C \in L_{\text{pub}}$ ；

20) $y_i \leftarrow R_{\text{client}}(t_i, C, \varepsilon)$ ；

21) **end for**

22) $\tilde{L}_{\text{part}} \leftarrow \emptyset$ ；

23) **for each** $C \in L_{\text{pub}}$ **do**

24) 收集端根据 $(1-\beta)n$ 个用户的报告值，计算噪音方体 $\tilde{C}$ ；

25) $\tilde{L}_{\text{pub}} \leftarrow \tilde{L}_{\text{pub}} \cup \tilde{C}$ ；

26) **end for**

27) **return** $\tilde{L}_{\text{pub}}$ ；

Partial-Pub 策略结合用户分组与集合覆盖技术在格结构中寻找部分方体。首先收集者采用 Full-Pub 策略的用户分组技术对 βn 个用户进行分

组。每个用户随机选择一个方体，利用 $R_{\text{client}}(\cdot)$ 对该方体的维度取值进行扰动后报告给收集者（步骤1至步骤6）。收集者计算 d 个维度值域的笛卡儿积 D ，得到最小误差 θ_{left} 与最大误差 θ_{right} 。利用折半查找寻找最小 θ 值。获得 θ 值后，遍历1到 $|L|$ 个方体，利用 Search-Partial 函数寻找到满足条件的部分方体集合 L_{pub} 。若返回结果为 No，代表 θ 值设置过小，令 $\theta = \theta_{\text{left}}$ ；否则，代表 θ 值设置过大，令 $\theta = \theta_{\text{right}}$ 。（步骤7至步骤16）。收集者寻找到满足条件的 L_{pub} 后，将其广播给 $(1-\beta)n$ 个用户（步骤17）。 $(1-\beta)n$ 中每个用户随机选择一个方体 $C \in L_{\text{pub}}$ ，利用 $R_{\text{client}}(\cdot)$ 对 C 的维度取值扰动，将扰动值报告给收集者（步骤18至步骤21）。收集者根据 $(1-\beta)n$ 个用户的报告值，计算 L_{pub} 中的噪音方体，并将噪音方体集合 $\tilde{L}_{\text{pub}}$ 发布（步骤23至步骤26）。

在 Search-Partial 函数中，对于全方体 L 中的每个方体，根据格结构得到每个方体 C 的祖先方体，以及从方体 C 得到的对应误差。

Search-Partial 函数具体描述如下。

```

1) for each  $C \in L$  do
2) 根据格结构得到方体  $C$  的祖先方体集合与对应误差;
3) 将误差大于  $\theta$  的祖先方体删除，得到方体  $C$  的覆盖集  $\text{cov}(C)$ ;
4) end for
5)  $L_{\text{pub}} \leftarrow \emptyset$ ,  $\text{COV} \leftarrow \emptyset$ ; //  $L_{\text{pub}}$  存储每次所选方体  $C$ ,  $\text{COV}$  存储已覆盖方体
6) for each  $s \in S$  do // 重复  $S$  次迭代
7) 选择方体  $C$ , 使得  $|\text{cov}(C) - \text{COV}|$  最大化;
8)  $L_{\text{pub}} \leftarrow L_{\text{pub}} \cup \{C\}$ ;
9)  $\text{COV} \leftarrow \text{COV} \cup \text{cov}(C)$ ;
9) 从  $L$  中删除当前所选方体  $C$ ;
10) 根据  $\text{COV}$  中已覆盖方体，利用未选择的方体  $\text{cov}(C)$  删除已覆盖的方体;
11) end for
12) if  $\text{COV} = L$  then
13) return  $L_{\text{pub}}$ ;
14) else
15) return No
将误差大于  $\theta$  值的方体，从祖先方体集合中

```

删除，得到方体 C 的覆盖集 $\text{cov}(C)$ （步骤1至步骤4）。COV 用来存储已覆盖的方体（步骤5）。选择使 $|\text{cov}(C) - \text{COV}|$ 最大化的方体 C 添加到 L_{pub} 中。将当前轮未覆盖方体放入集合 COV（步骤7和步骤8）。将当前已选方体从 L 中删除，更新每个方体的 $\text{cov}(C)$ 。在剩余方体的 $\text{cov}(C)$ 中删除当前已覆盖方体（步骤9和步骤10）。若 COV 已覆盖的方体等于全方体 L ，则返回 L_{pub} ；否则返回 No（步骤12至步骤15）。

例如，给定事实表 $T = \{A:3, B:4, C:3\}$ ，对应方体数为 2^3 。给定10万个用户，20%的用户寻找部分方体，80%的用户响应查询。以扰动方法 OUE 为例，给定 $\varepsilon = 1.0$ ，由 Partial-Pub 策略可知 $\theta_{\text{right}} = 883\,890.84$ 。可计算出除 all 方体外每个方体中单元格的误差为 $\{A: 32\,731.78, B: 32\,735.47, C: 32\,735.47, AB: 32\,735.47, AC: 32\,735.47, BC: 32\,735.47, ABC: 98\,206.41\}$ 。根据格蕴含的偏序关系，可以获得每个方体的祖先集合： $\{A: [A, \text{all}], B: [B, \text{all}], C: [C, \text{all}], AB: [AB, A, B, \text{all}], AC: [AC, A, C, \text{all}], BC: [BC, B, C, \text{all}], ABC: [ABC, AB, AC, BC, A, B, C, \text{all}], \text{all}: [\text{all}]\}$ 。若给定 $\theta = 441\,928.84$ ，把误差值大于 θ 的方体从集合中删除。若寻找 $S=3$ 的部分方体，第1轮迭代选择方体 ABC， $\text{Cov}(ABC) = \{ABC, AB, AC, BC\}$ 覆盖最多方体。第2轮迭代选择方体 AB， $\text{Cov}(AB) = \{A, B, \text{all}\}$ 覆盖最多方体。第3轮迭代选择方体 A， $\text{Cov}(BC) = \{C\}$ 。因此， $S=3$ 时所选方体集合为 $L_{\text{pub}} = \{ABC, AB, BC\}$ ，能覆盖全部方体。

Partial-Pub 策略的时间复杂度来自多个方面，首先是二分查找的时间复杂度。算法2中利用二分查找来确定最小阈值 θ ，直到算法满足条件 $|\theta_{\text{right}} - \theta_{\text{left}}| > 1$ 。由于每次迭代需要搜索区间缩小为原区间的一半，则最大迭代次数为 $O(\log|L|)$ 。其次是 Search-Partial(L, θ, S) 函数的时间复杂度。在每次二分查找迭代中需要调用 Search-Partial(L, θ, S) 函数 $|L|$ 次。计算每个方体 C 的覆盖集 $\text{cov}(C)$ 需要进行 S 次选择，最多选择 $|L|$ 次。每个方体 C 的覆盖集 $\text{cov}(C)$ 的上界为 $|L|$ ，总共有 2^d 个方体，因此 $\sum_C |\text{cov}(C)| \leq 2^d |L|$ 。结合上述分析可知，Partial-Pub 策略的时间复杂度为 $O(2^d |L|^2 \log|L|)$ 。

3.2 $R_{\text{client}}(\cdot)$ 本地扰动机制的实现

由算法1与算法2可知, 每个用户利用 $R_{\text{client}}(\cdot)$ 本地扰动自身数据。本文结合式(2)–式(4)设计了 Cuboid-OUE、Cuboid-HHT 和 Cuboid-CMS 3种扰动机制来实现 $R_{\text{client}}(\cdot)$ 。其中 Cuboid-OUE 的具体描述见算法3。

算法3 Cuboid-OUE

输入 用户 u_i 的数据 $t_i = (t_i[A_1], t_i[A_2], \dots, t_i[A_d])$, 方体 C , 隐私预算 ε

输出 B'_i

1) u_i 计算 $\{A_1, A_2, \dots, A_m\}$ 的值域笛卡儿积 $D = \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_m)$;

2) u_i 利用 C 包含的属性筛选 t_i 的属性值 $\{a[i_1], a[i_2], \dots, a[i_m]\}$;

3) u_i 初始化长度为 $|D|$ 的全0向量 B_i ;

4) u_i 根据筛选的属性值把 B_i 中对应位置1;

5) u_i 本地扰动 B_i 中的每一位, 得到 B'_i , 形如

$$\Pr[B'_i[j] = 1] = \begin{cases} \frac{1}{2}, & B_i[j] = 1 \\ \frac{1}{e^\varepsilon + 1}, & B_i[j] = 0 \end{cases};$$

6) return B'_i .

用户 u_i 根据方体 C 的维度 $\{A_1, A_2, \dots, A_m\}$ 计算维度值域的笛卡儿积 D , 再利用维度 $\{A_1, A_2, \dots, A_m\}$ 筛选 t_i 的属性值, 并初始化长度为 $|D|$ 的全0向量。 u_i 把 t_i 的属性值编码成长度为 $|D|$ 的向量 B_i , 且只有一位为1。利用式(4)扰动 B_i 中每一位, 得到噪音向量 B'_i 。以图2(b)为例, 假设用户 u_1 所选方体为 $\{A, C\}$ 。 u_1 需要向收集者报告方体 $\{A, C\}$ 对应的维度取值。维度 A 与 C 值域的笛卡儿积组合为 $D = \{(1 \sim 20, F), (21 \sim 40, F), (41 \sim 60, F), (60+, F), (1 \sim 20, M), (21 \sim 40, M), (41 \sim 60, M), (60+, M)\}$, 总共有8种情况。 u_1 的元组 t_1 对应该方体的维度取值为 $(1 \sim 20, M)$ 。首先对值 $(1 \sim 20, M)$ 进行编码, 则 $B_1 = [0, 0, 0, 0, 1, 0, 0, 0]$, 然后对 B_1 中的每一位进行扰动即可。

定理1 Cuboid-OUE 满足 ε -LDP。

证明 假设方体 C 对应维度值域的笛卡儿积为 D , 由算法3可知, u_i 将元组 t_i 中维度 $[C]$ 对应的值 v_i 编码成长度为 $|D|$ 的0-1向量, 扰动概率有4种情况: $\Pr[1 \rightarrow 1] = 1/2$, $\Pr[0 \rightarrow 1] = 1/e^\varepsilon + 1$, $\Pr[0 \rightarrow 0] = e^\varepsilon/e^\varepsilon + 1$, $\Pr[1 \rightarrow 0] = 1/2$ 。由定义1可知, Cuboid-OUE 满足 ε -LDP。

定理2 给定格中任意一个方体 C , 该方体中任意一个单元格 a 的真实用户计数为 $c(a)$, $\tilde{c}(a)$ 表示单元格 a 的估计计数, 则 $\mathbb{E}[\tilde{c}(a)] = c(a)$ 成立。

证明 令 $c'(a)$ 表示单元格 a 中用户计数的观测值, $n(C)$ 为方体 C 中用户计数。观测值 $c'(a)$ 可表示为 $c'(a) = c(a)p + (n(C) - c(a))q$, 其中 $p = 1/2$, $q = 1/e^\varepsilon + 1$ 。由极大似然定理可知, $c(a)$ 的估计值 $\tilde{c}(a)$ 可表示为: $\tilde{c}(a) = c'(a) - n(C)q/p - q$ 。则 $\mathbb{E}[\tilde{c}(a)] = \mathbb{E}[c'(a) - n(C)q/p - q]$ 成立。把 $c'(a)$ 代入 $\mathbb{E}[\tilde{c}(a)] = \mathbb{E}[c'(a) - n(C)q/p - q]$, 得 $\mathbb{E}[\tilde{c}(a)] = c(a)$ 成立。

定理3 给定任意一个由 Cuboid-OUE 产生的方体 C , C 中任意一个单元格 a 的估计计数 $\tilde{c}(a)$ 所引起的误差为:

$$\text{Var}[\tilde{c}(a)] = n(C) \times \frac{4e^\varepsilon}{(e^\varepsilon - 1)^2} \quad (9)$$

证明 由定理2中 $\tilde{c}(a) = c'(a) - n(C)q/p - q$ 可知, $\text{Var}[\tilde{c}(a)] = \text{Var}[c'(a) - n(C)q/p - q]$ 。

则

$$\begin{aligned} \text{Var}[\tilde{c}(a)] &= \text{Var}\left[\frac{c'(a) - n(C)q}{p - q}\right] = \frac{\text{Var}[c'(a)]}{(p - q)^2} = \\ &= \frac{c(a)p(1 - p) + (n(C) - c(a))q(1 - q)}{(p - q)^2} \approx \\ &= \frac{n(C)q(1 - q)}{(p - q)^2} \end{aligned} \quad (10)$$

将 $p = 1/2$, $q = 1/e^\varepsilon + 1$ 代入式(10)中可得:

$$\text{Var}[\tilde{c}(a)] \approx \frac{n(C)q(1 - q)}{(p - q)^2} = n(C) \times \frac{4e^\varepsilon}{(e^\varepsilon - 1)^2} \quad (11)$$

由式(11)可知定理3成立。

定理4 Cuboid-OUE 产生的估计值 $\tilde{c}(a)$ 与 $c(a)$

之间的最大偏差 $|\tilde{c}(a) - c(a)| = O\left(\frac{\sqrt{n(C)\ln(1/\beta)}}{\varepsilon}\right)$

至少以 $1 - \beta$ 的概率成立。

证明 设 $\tilde{c}(a) - c(a)$ 为一个随机变量, 由定理2可知, 其均值为0。

该随机变量可表示 $\tilde{c}(a) - c(a) = \sum_{i=1}^{n(C)} \frac{z_i - q}{p - q} - w_i$

其中, $z_i \in \{0, 1\}$ 、 $w_i \in \{0, 1\}$ 分别表示第 i 个用

户的观测值与真实值。由定理3可知，该随机变量的方差为： $\text{Var}\left[\frac{z_i - q}{p - q} - w_i\right] = \frac{4e^\varepsilon}{(e^\varepsilon - 1)^2}$ 。

随机变量 $\frac{z_i - q}{p - q} - w_i$ 的取值分为4种情况，分别为 $\frac{1 - q}{p - q} - 0, \frac{1 - q}{p - q} - 1, \frac{0 - q}{p - q} - 0, \frac{0 - q}{p - q} - 1$ 。可得到最大值为 $M = \frac{1 - q}{p - q} - 0$ ，将 $p=1/2, q=1/(e^\varepsilon + 1)$ 代入可得 $M = \frac{2e^\varepsilon}{e^\varepsilon - 1}$ 。

根据伯恩斯不等式可得：

$$\begin{aligned} \Pr[\tilde{c}(a) - c(a) \geq \lambda] &= \\ \Pr\left[\sum_{i=1}^{n(C)} \left|\frac{z_i - q}{p - q} - w_i\right| \geq \lambda\right] &\leq \\ 2\exp\left[-\frac{\lambda^2}{2 \sum_{i=1}^{n(C)} \frac{4e^\varepsilon}{(e^\varepsilon - 1)^2} + \frac{2}{3} \times \frac{2e^\varepsilon}{e^\varepsilon - 1} \lambda}\right] &= \\ 2\exp\left[O\left(-\frac{\lambda^2}{n(C) \frac{1}{\varepsilon^2} + \frac{1}{\varepsilon} \lambda}\right)\right] & \\ \text{令 } \beta = \exp\left[O\left(-\frac{\lambda^2}{n(C) \frac{1}{\varepsilon^2} + \frac{1}{\varepsilon} \lambda}\right)\right], \text{ 则 } \lambda = & \\ O\left(\frac{\sqrt{n \ln(1/\beta)}}{\varepsilon}\right) \text{ 使 } |\tilde{c}(a) - c(a)| \text{ 至少以 } 1 - \beta \text{ 的概率成立。} & \end{aligned}$$

上述 Cuboid-OUE 采用 One-Hot 编码^[19]对用户的原始值进行处理，且只有一位值为1。由算法3步骤1可知，当笛卡儿积很大时，用户所拥有的向量会很稀疏，进而导致 Cuboid-OUE 扰动结果的可用性较低。哈达玛矩阵是对稀疏向量进行转换的常用技术，该技术要求哈达玛矩阵的大小必须为2的整数次幂。基于此，本文提出了 Cuboid-HHT 本地扰动机制，该机制首先利用哈希函数将维度值域压缩到2的整数次幂，再对其进行哈达玛转换并扰动。Cuboid-HHT 描述见算法4。

算法4 Cuboid-HHT

输入 用户 u_i 的数据 $t_i = (t_i[A_1], t_i[A_2], \dots, t_i[A_d])$ ，方体 C ，隐私预算 ε ，哈希函数簇 $H = \{h_1, h_2, \dots, h_k\}$

输出 $\theta'_i[j], j$

1) $v_i \leftarrow u_i$ 利用 C 包含的属性筛选 t_i 的属性值 $\{a[i_1], a[i_2], \dots, a[i_m]\}$;
2) 初始化 $\mathbf{B}_i \in \{0\}^l$;
3) $\mathbf{B}_i[h_j[v_i]] = 1$; // u_i 随机采样哈希函数 h_j
4) $\theta_i \leftarrow \mathbf{H}_i \mathbf{B}_i$; // 哈达玛矩阵对向量 $\mathbf{B}_i$ 进行转换
5) $\theta_i[j] \leftarrow$ 从 $[\theta_i]$ 中随机采样索引 j 并取出其对应的值;

6) $\theta'_i[j] \leftarrow \text{RR}[\theta_i[j]]$ ，扰动概率形如：

$$\Pr[\theta'_i[j]] = \begin{cases} \frac{e^\varepsilon}{e^\varepsilon + 1}, & \theta_i[j] = \theta_i[j] \\ \frac{1}{e^\varepsilon + 1}, & \theta_i[j] \neq \theta_i[j] \end{cases};$$

7) return $\theta'_i[j], j$

用户 u_i 根据方体 C 的维度，筛选出元组 t_i 中的属性值 v_i 。初始化长度为 l 的全0向量 $\mathbf{B}_i$ ，利用函数 h_j 将值 v_i 哈希到 l 中某一位，将其置为1（步骤3）。结合定义3，将哈希向量 $\mathbf{B}_i$ 与哈达玛矩阵 $\mathbf{H}_i$ 相乘，得到转换后的向量 θ_i （步骤4）。从 θ_i 中随机采样索引 j ，对值 $\theta_i[j]$ 进行 RR 扰动，返回扰动值与哈希索引 j （步骤5至步骤7）。

定理5 Cuboid-HHT 满足 ε -LDP。

证明 假设方体 C 维度值域的笛卡儿积为 D ，用户 u_i 将元组 t_i 中维度 $[C]$ 对应的值 v_i 经过哈希后，编码成长度为 l 的0-1向量，扰动概率存在如下：

$$\begin{aligned} \Pr[1 \rightarrow 1] &= \frac{e^\varepsilon}{e^\varepsilon + 1} = p, \Pr[0 \rightarrow 1] = \frac{1}{e^\varepsilon + 1} = q \\ \Pr[0 \rightarrow 0] &= \frac{e^\varepsilon}{e^\varepsilon + 1} = p, \Pr[1 \rightarrow 0] = \frac{1}{e^\varepsilon + 1} = q \end{aligned} \quad (12)$$

根据定义1可知，Cuboid-HHT 满足 ε -LDP。

定理6 给定用户 u_i 所采样索引 j 对应的真实值为 $\theta_i[j]$ ，估计值为 $\theta^*_i[j]$ ，则 $\mathbb{E}[\theta^*_i[j]] = \theta_i[j]$ 成立。

证明 u_i 的值 v_i 经过编码与哈达玛转换后的向量为 θ_i 。从该向量中随机采样一位索引 j ，并对值 $\theta_i[j]$ 进行扰动。假设 $\theta'_i[j]$ 为观察值， $\theta^*_i[j]$ 为估计值。 $\theta_i[j]$ 不考虑哈达玛系数时，取值为1或-1。

1) u_i 以 $1/l$ 的概率采样索引 j 。若观察值 $\theta'_i[j]=1$, 则 $\theta'_i[j]=p \times 1 \times 1/l + (1-p) \times (-1) \times 1/l = 1/l \times 2p-1$ 。可获得校正因子 $l/2p-1$ 。若观察值 $\theta'_i[j]=-1$, 则 $\theta'_i[j]=p \times (-1) + (1-p) \times 1 = -(2p-1) \times 1/l$, 可获得校正因子 $-l/(2p-1)$ 。 $\theta_i[j]=1$ 时, 以 p 的概率等式 $\theta^*[j]=l/(2p-1)$ 成立, 否则以 $1-p$ 的概率等式 $\theta^*[j]=-l/(2p-1)$ 成立, 则 $\mathbb{E}[\theta^*[j]]=\theta_i[j]$ 。 $\theta_i[j]=-1$ 时, 以 p 的概率等式 $\theta^*[j]=-l/(2p-1)$ 成立, 否则以 $1-p$ 的概率等式 $\theta^*[j]=l/(2p-1)$ 成立, 则 $\mathbb{E}[\theta^*[j]]=\theta_i[j]$ 成立。

2) u_i 以 $1-1/l$ 的概率采样索引 j , 则 $\theta^*[j]=0$ 。

由上述分析可知, $\mathbb{E}[\theta^*[j]]=\theta_i[j]$ 成立。

定理 7 给定 Cuboid-HHT 中任意用户 u_i , u_i 从 $[l]$ 中采样到的索引 j 所对应的真实值与估计值分别为 $\theta_i[j]$ 和 $\theta^*[j]$ 。二者之间的误差为:

$$\text{Var}[\theta_i^*[j] - \theta_i[j]] = \frac{4le^\epsilon}{(e^\epsilon - 1)^2} + \left(1 - \frac{1}{l}\right) \quad (13)$$

证明 由定理 6 可知, $\mathbb{E}[\theta^*[j]]=\theta_i[j]$, 则

$$\begin{aligned} \text{Var}[\theta_i^*[j] - \theta_i[j]] &= \\ \mathbb{E}[(\theta_i^*[j] - \theta_i[j])^2] &- \\ \mathbb{E}[\theta_i^*[j] - \theta_i[j]]^2 &= \\ \mathbb{E}[(\theta_i^*[j] - \theta_i[j])^2] & \end{aligned} \quad (14)$$

其中, $\theta^*[j]$ 与 $\theta_i[j]$ 的取值包括: $\theta_i[j]=1$ 时, $\theta^*[j]=l/(2p-1)$ 或 $\theta^*[j]=-l/(2p-1)$; $\theta_i[j]=1$ 或 -1 时, $\theta^*[j]=0$; $\theta^*[j]=-l/(2p-1)$ 或 $\theta^*[j]=l/(2p-1)$ 。

$|\theta^*[j]-\theta_i[j]|$ 存在两种情况: $|l/(2p-1)-1/l|$ 与 $|l/(2p-1)+l|$ 。

根据上述描述以及期望的性质可知, 式(14)可以表示为式(15)。

$$\begin{aligned} \text{Var}[\theta_i^*[j] - \theta_i[j]] &= \mathbb{E}[(\theta_i^*[j] - \theta_i[j])^2] = \\ \frac{1}{l} pl^2 \left(\frac{1}{2p-1} - 1 \right)^2 &+ \frac{1}{l} (1-p) l^2 \left(\frac{1}{2p-1} + 1 \right)^2 + \\ (0 - \theta_i[j])^2 \left(1 - \frac{1}{l} \right) &= \\ \frac{p(2-2p)^2 + (1-p)(2p)^2}{(2p-1)^2} l &+ \left(1 - \frac{1}{l} \right) = \\ \frac{4p(1-p)l}{(2p-1)^2} + \left(1 - \frac{1}{l} \right) & \end{aligned} \quad (15)$$

将式(12)中的概率 p 代入式(15)可得采样索引 j 的真实值 $\theta_i[j]$ 与估计值 $\theta^*[j]$ 之间的误差为

$$\frac{4e^\epsilon l}{(e^\epsilon - 1)^2} + \left(1 - \frac{1}{l}\right), \text{ 则定理 7 成立。}$$

定理 8 Cuboid-HHT 产生的估计值 $\tilde{c}(a)$ 与 $c(a)$

的最大偏差 $|\tilde{c}(a) - c(a)| = O\left(\frac{\sqrt{n(C)l \ln(1/\beta)}}{\epsilon}\right)$ 至少以 $1-\beta$ 的概率成立。

证明 由定理 7 可知, 对于任何单元格 a 的估计计数与真实计数之间的差值可表示为:

$$\tilde{c}(a) - c(a) = \sum_{i=1}^{n(C)} \theta_i^*[j] - \theta_i[j]. \text{ 此时 } \theta_i^*[j] - \theta_i[j]$$

为一个随机变量, 根据定理 6 可知, 其均值为 0, 根据定理 7 可知, 其方差为:

$$\text{Var}[\theta_i^*[j] - \theta_i[j]] = \frac{4e^\epsilon l}{(e^\epsilon - 1)^2} + \left(1 - \frac{1}{l}\right) \leq \frac{1}{\epsilon^2} \quad (16)$$

由定理 7 可知, $|\theta_i^*[j] - \theta_i[j]|$ 的最大值为 $M = l \times \left| \frac{1}{2p-1} + 1 \right| = \frac{2le^\epsilon}{e^\epsilon - 1}$ 。

根据伯恩斯坦不等式与式(16)可知:

$$\Pr[|\tilde{c}(a) - c(a)| \geq \lambda] =$$

$$\Pr\left[\sum_{i=1}^{n(C)} |\theta_i^*[j] - \theta_i[j]| \geq \lambda\right] \leq$$

$$2 \exp \left[- \frac{\lambda^2}{2 \times \sum_{i=1}^{n(C)} \text{Var}[\theta_i^*[j] - \theta_i[j]] + \frac{2}{3} \times M \lambda} \right] \leq$$

$$\begin{aligned} & 2 \exp \left[O \left(- \frac{\lambda^2}{\frac{n(C)l}{\epsilon^2} + \frac{l}{\epsilon} \lambda} \right) \right] \\ \text{令 } \beta &= \exp \left[O \left(- \frac{\lambda^2}{\frac{n(C)l}{\epsilon^2} + \frac{l}{\epsilon} \lambda} \right) \right], \text{ 则 } \lambda = \\ & O \left(\frac{\sqrt{n(C)l \ln(1/\beta)}}{\epsilon} \right) \text{ 使 } |\tilde{c}(a) - c(a)| \text{ 至少以 } 1-\beta \end{aligned}$$

的概率成立。

每个方体包含的属性维度不同，方体的值域取值不同，对于值域大且稀疏的方体会导致 Cuboid-OUE 机制的传输代价增加与 Cuboid-HHT 机制的误差增加。不同于哈达玛矩阵，计数均值草图^[20]是转换稀疏向量另外一种技术，该技术能够精确高效地估计方体单元格中的计数。基于此，本文提出了 Cuboid-CMS 机制，具体描述见算法5。

算法5 Cuboid-CMS

输入 用户的数据 $t_i = (t_i[A_1], t_i[A_2], \dots, t_i[A_d])$ ，方体 C ，隐私预算 ε ，哈希函数簇 $H = \{h_1, h_2, \dots, h_k\}$ ，向量长度 l

输出 B'_i, j

- 1) $v_i \leftarrow u_i$ 利用 C 包含的属性筛选 t_i 的属性值 $\{a[i_1], a[i_2], \dots, a[i_m]\}$;
- 2) u_i 随机选择一个哈希函数 $h_j \in H$;
- 3) u_i 初始化 $B_i \in \{-1\}^l$; // l 且全为-1 的向量
- 4) $B_i[h_j(v_i)] = 1$; // 哈希 v_i 将其索引位置 1
- 5) u_i 本地扰动 B_i ，得到 B'_i ，形如：

$$\Pr[B'_i[j] = 1] = \begin{cases} \frac{e^{\varepsilon/2}}{e^{\varepsilon/2} + 1}, & B_i[j] = 1 \\ \frac{1}{e^{\varepsilon/2} + 1}, & B_i[j] = 0 \end{cases};$$

- 6) return B'_i, j ; // B'_i 与采样哈希函数索引 j

用户 u_i 根据方体 C 的维度，筛选出元组 t_i 中的属性值 v_i 。从簇 H 中随机选择一个哈希函数 h_j ，初始化长度为 l 且全为-1 的向量 B_i 。使用 h_j 将值 v_i 哈希到长度为 l 的向量中，只有一位为1。用户 u_i 利用 SUE 机制扰动向量中的每一位，将噪音向量与哈希函数索引 j 发送给收集者。

定理9 Cuboid-CMS 满足 ε -LDP。

证明 假设方体 C 维度值域的笛卡儿积为 D ，给定 D 内任意两个不同的值 v_i 与 v_j 。 $j \in [k]$ 为随机选择的哈希函数索引。 v_i 与 v_j 经过哈希后形成的向量分别为 $B_i, B_j \in \{-1, 1\}^l$ 。存在两种情况：

- ① 若 $h_j(v_i) = h_j(v_j)$ ，即向量 B_i 与 B_j 中置1的索引位相同。假设输出向量为 B' ，则：

$$\frac{\Pr[B'B_i]}{\Pr[B'B_j]} = \prod_{l=1}^m \frac{\Pr[B'[l]|B_i[l]]}{\Pr[B'[l]|B_j[l]]} = 1 \quad (17)$$

- ② 若 $h_j(v_i) \neq h_j(v_j)$ ，即向量 B_i 与 B_j 中置1的索引位不相同，则有两个位置不同：

$$\begin{aligned} \frac{\Pr[B'B_i]}{\Pr[B'B_j]} &= \prod_{l=1}^m \frac{\Pr[B'[l]|B_i[l]]}{\Pr[B'[l]|B_j[l]]} \leq \\ &\frac{\Pr[B[h_j(v_i)] = 1|B_i[h_j(v_i)] = 1]}{\Pr[B[h_j(v_i)] = 1|B_j[h_j(v_j)] = 0]} \times \\ &\frac{\Pr[B[h_j(v_j)] = 0|B_i[h_j(v_i)] = 0]}{\Pr[B[h_j(v_j)] = 0|B_j[h_j(v_j)] = 1]} = \\ &\left(\frac{e^{\varepsilon/2}}{e^{\varepsilon/2} + 1} / \frac{1}{e^{\varepsilon/2} + 1} \right)^2 \leq e^\varepsilon \end{aligned} \quad (18)$$

根据定义1可知，Cuboid-CMS 满足 ε -LDP。

定理10 设 C 是由 Cuboid-CMS 产生的任意方体。 C 的任意单元格 a 中真实用户计数为 $c(a)$ ， $\tilde{c}(a)$ 为对应的估计数，则 $\mathbb{E}[\tilde{c}(a)] = c(a)$ 成立。

证明 由算法5可知， v_i 值经过编码后形成向量 B_i ，其中只有一位为1，其余位为-1。利用 SUE 机制对 B_i 的每一位进行扰动。观察到值1的情况为：

$$1 \times \frac{e^{\varepsilon/2}}{e^{\varepsilon/2} + 1} + (-1) \times \frac{1}{e^{\varepsilon/2} + 1} = \frac{e^{\varepsilon/2} - 1}{e^{\varepsilon/2} + 1}.$$

观察到值-1的情况为：

$$(-1) \times \frac{e^{\varepsilon/2}}{e^{\varepsilon/2} + 1} + 1 \times \frac{1}{e^{\varepsilon/2} + 1} = \frac{e^{\varepsilon/2} - 1}{e^{\varepsilon/2} + 1} \times (-1), \text{ 其中 } \frac{e^{\varepsilon/2} + 1}{e^{\varepsilon/2} - 1} \text{ 表示为}$$

校正因子。令 $\sigma_\varepsilon = \frac{e^{\varepsilon/2} + 1}{e^{\varepsilon/2} - 1}$ ，则每个用户经过校正后的向量可表示为：

$$\tilde{B}_i = \frac{k\sigma_\varepsilon}{2} \cdot B'_i + \frac{k}{2} \cdot \mathbf{1}.$$

令 $b \in \{-1, 1\}$ 表示扰动值，则值1以 $\frac{e^{\varepsilon/2}}{e^{\varepsilon/2} + 1}$ 的概率被抽样到。给定方体 C 中的单元格 a ， $B_i[h_j(a)] = 1$ 所表示向量 B_i 的第 $h_j(a)$ 个索引位为1。若要估计单元格 a 的计数值，则存在以下两种情况：

- ① 用户 u_i 所拥有的值 $a^{(i)} = a$ ， $h_j(a^{(i)}) = h_j(a)$ ，则：

$$\begin{aligned} \mathbb{E}[\tilde{B}_i(a)] &= \mathbb{E}\left[\frac{k\sigma_\varepsilon b B_i[h_j(a)] + 1}{2} + \frac{k}{2}\right] = \\ \Pr[J = j] \times \frac{k}{2} \times \mathbb{E}[\sigma_\varepsilon b B_i[h_j(a)] + 1] &= \end{aligned} \quad (19)$$

$$\frac{1}{k} \times \frac{k}{2} \times \frac{e^{\varepsilon/2} - 1}{e^{\varepsilon/2} + 1} \times \frac{e^{\varepsilon/2} + 1}{e^{\varepsilon/2} - 1} + \frac{1}{k} \times \frac{k}{2} = 1$$

- ② 用户 u_i 所拥有的值 $a^{(i)} \neq a$ ，存在 $h_j(a^{(i)}) = h_j(a)$ 与 $h_j(a^{(i)}) \neq h_j(a)$ ，则：

$$\begin{aligned}\mathbb{E}[\tilde{B}_i(a)] &= \mathbb{E}\left[\frac{k\sigma_\varepsilon b\mathbf{B}_i[h_j(a)] + 1}{2} + \frac{k}{2}\right] = \\ \Pr[J=j] \times \frac{k}{2} \times \mathbb{E}[\sigma_\varepsilon b\mathbf{B}_i[h_j(a)] + 1] &= \\ \frac{1}{2} \times \frac{1}{l} \times \mathbb{E}[\sigma_\varepsilon b + 1] + \frac{1}{2} \times \left(1 - \frac{1}{l}\right) \times & \\ \mathbb{E}[\sigma_\varepsilon b(-1) + 1] = & \\ \frac{1}{2} \times \frac{1}{l} \times 1 + \frac{1}{2} \times \left(1 - \frac{1}{l}\right) \times 0 = \frac{1}{l} &\end{aligned}\quad (20)$$

由以上分析可知:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{k} \sum_{j=1}^k M_{j,h_j(a)}\right] &= \mathbb{E}\left[\frac{1}{k} \sum_{i=1}^{n(C)} \sum_{j=1}^k \tilde{B}_i(a)\right] = \\ \frac{1}{k} \sum_{i=1}^{n(C)} \sum_{j=1}^k \mathbb{E}[\tilde{B}_i(a)] &= c(a) + (n(C) - c(a)) \frac{1}{l}\end{aligned}\quad (21)$$

$n(C)$ 为 C 中的总用户数, 由式(22)可知, $c(a)$ 的估计数为:

$$\tilde{c}(a) = \left(\frac{l}{l-1}\right) \left(\frac{1}{k} \sum_{j=1}^k \sum_i \tilde{B}_i - \frac{n(C)}{l}\right) \quad (22)$$

则估计值 $\tilde{c}(a)$ 的期望为:

$$\begin{aligned}\mathbb{E}[\tilde{c}(a)] &= \mathbb{E}\left[\left(\frac{l}{l-1}\right) \left(\frac{1}{k} \sum_{j=1}^k \sum_i \tilde{B}_i - \frac{n(C)}{l}\right)\right] = \\ \frac{l}{l-1} \times \mathbb{E}\left[\frac{1}{k} \sum_{j=1}^k \sum_i \tilde{B}_i\right] - \frac{n(C)}{l-1} &= \\ \frac{l}{l-1} \times \left(c(a) \times \left(1 - \frac{1}{l}\right) + \frac{n(C)}{l}\right) - \frac{n(C)}{l-1} &= \\ \frac{l}{l-1} c(a) - \frac{1}{l-1} = c(a) &\end{aligned}\quad (23)$$

由式(23)可知, $\mathbb{E}[\tilde{c}(a)] = c(a)$, 定理 10 成立。

定理 11 给定 Cuboid-CMS 产生的方体 C , C 中任意单元格 a 的估计值 $\tilde{c}(a)$ 产生的误差为:

$$\text{Var}[\tilde{c}(a)] = \left(\frac{l}{l-1}\right)^2 \times n(C) \times \left(\frac{e^{\varepsilon/2}}{(e^{\varepsilon/2} - 1)^2} + \frac{1}{l}\right)$$

证明 估计值 $\tilde{c}(a)$ 所引起的误差为:

$$\begin{aligned}\text{Var}[\tilde{c}(a)] &= \text{Var}\left[\left(\frac{l}{l-1}\right) \left(\frac{1}{k} \sum_{j=1}^k \sum_i \tilde{B}_i - \frac{n(C)}{l}\right)\right] = \\ \left(\frac{l}{l-1}\right)^2 \times \text{Var}\left[\frac{1}{k} \sum_{j=1}^k \sum_i \tilde{B}_i\right] &\end{aligned}\quad (24)$$

首先计算式(24)中右侧的 Var 部分:

$$\begin{aligned}\text{Var}\left[\frac{1}{k} \sum_{j=1}^k \sum_{i=1}^{n(C)} \tilde{B}_i\right] &= \frac{1}{k^2} \sum_{i=1}^{n(C)} \text{Var}\left[\sum_{j=1}^k \tilde{B}_i(a)\right] + \\ \frac{1}{k^2} \sum_{i_1 \neq i_2} \text{Cov}\left(\sum_{j=1}^k \tilde{B}_{i_1}(a), \sum_{j=1}^k \tilde{B}_{i_2}(a)\right) &\end{aligned}\quad (25)$$

根据方差的性质和协方差来计算式(25)求和的第一部分:

$$\begin{aligned}\frac{1}{k^2} \sum_{i=1}^{n(C)} \text{Var}\left[\sum_{j=1}^k \tilde{B}_i(a)\right] &= \\ \frac{1}{k^2} \sum_{i=1}^{n(C)} \left(\sum_{j=1}^k \text{Var}[\tilde{B}_i(a)] + \frac{1}{k^2} \sum_{j \neq j'} \text{Cov}(\tilde{B}_{i_1}(a), \tilde{B}_{i_2}(a))\right) &= \\ \frac{1}{k^2} \sum_{i=1}^{n(C)} \sum_{j=1}^k \left[\mathbb{E}[\tilde{B}_i(a)^2] - \mathbb{E}[\tilde{B}_i(a)]^2\right] &= \\ \frac{n(C)}{4} (\sigma_\varepsilon^2 - 1) + \frac{n(c) - c(a)}{l} \left(1 - \frac{1}{l}\right) &\end{aligned}\quad (26)$$

利用协方差计算式(25)求和的第二部分:

$$\begin{aligned}\frac{1}{k^2} \sum_{i_1 \neq i_2} \text{Cov}\left(\sum_{j=1}^k \tilde{B}_{i_1}(a), \sum_{j=1}^k \tilde{B}_{i_2}(a)\right) &= \\ x \frac{1}{k} \sum_{i_1 \neq i_2} \text{Cov}(\tilde{B}_{i_1}(a), \tilde{B}_{i_2}(a)) &= \\ \frac{1}{k} \left(\frac{1}{l} - \frac{1}{l^2}\right) \sum_{a' \neq a, a' = a'} (c(a') - 1) &= \\ \frac{1}{k} \left(\frac{1}{l} - \frac{1}{l^2}\right) \sum_{a' \neq d} (c(a'))^2 - (n - c(a)) &\end{aligned}\quad (27)$$

由式(26)与式(27)可知, 式(25)的结果为:

$$\begin{aligned}\text{Var}\left[\frac{1}{k} \sum_{j=1}^k \sum_{i=1}^{n(C)} \tilde{B}_i\right] &= \\ \frac{n(C)}{4} (\sigma_\varepsilon^2 - 1) + \frac{n(c) - c(a)}{l} \left(1 - \frac{1}{l}\right) + \\ \frac{1}{k} \left(\frac{1}{l} - \frac{1}{l^2}\right) \sum_{a' \neq d} (c(a'))^2 - (n - c(a)) &\leq \\ \frac{n(C)}{4} (\sigma_\varepsilon^2 - 1) + \frac{n(C)}{l} + \frac{1}{kl} \sum_{a' \in D} (c(a'))^2 &\end{aligned}\quad (28)$$

将式(28)代入式(24)中可得:

$$\begin{aligned}\text{Var}[\tilde{c}(a)] &\approx \left(\frac{l}{l-1}\right)^2 \left(\frac{n(C)}{4} (\sigma_\varepsilon^2 - 1) + \right. \\ \frac{n(C)}{l} + \frac{1}{kl} \sum_{a' \in D} (c(a'))^2 &\leq \\ \left.\left(\frac{l}{l-1}\right)^2 \left(\frac{n(C)}{4} (\sigma_\varepsilon^2 - 1) + \frac{n(C)}{l}\right)\right) &\end{aligned}\quad (29)$$

由式(29)可知定理11成立。

定理12 Cuboid-CMS产生的 $\tilde{c}(a)$ 与 $c(a)$ 之间

的最大偏差 $|\tilde{c}(a) - c(a)| = O\left(\frac{\sqrt{n(C)\ln(1/\beta)}}{\varepsilon}\right)$ 至少以 $1-\beta$ 的概率成立。

证明 由定理10可知, $\tilde{c}(a) - c(a) = \sum_{i=1}^{n(C)} (\tilde{B}_i - B_i)$ 成立。此时 $\tilde{B}_i - B_i$ 为一个随机变量。根据定理10可知, 其均值为0。由定理11可知, $\tilde{B}_i - B_i$ 的方差为:

$$\text{Var}[\tilde{B}_i - B_i] \leq \left(\frac{l}{l-1}\right)^2 \times \left(\frac{e^{\varepsilon/2}}{(e^{\varepsilon/2} - 1)^2} + \frac{1}{l}\right) \quad (30)$$

$\tilde{B}_i - B_i$ 可表示为:

$$\tilde{B}_i - B_i = \frac{k\sigma_\varepsilon}{2} \cdot B'_i + \frac{k}{2} \cdot \mathbf{1} - B_i \quad (31)$$

由式(31)可知, $\tilde{B}_i - B_i$ 的最大值为 $M = \frac{ke^{\varepsilon/2}}{e^{\varepsilon/2} - 1}$ 。由伯恩斯坦不等式与式(30)可知:

$$\begin{aligned} \Pr[|\tilde{c}(a) - c(a)| \geq \lambda] &= \Pr\left[\left|\sum_{i=1}^{n(C)} \tilde{B}_i - B_i\right| \geq \lambda\right] \leq \\ &2\exp\left[-\frac{\lambda^2}{2 \times \sum_{i=1}^{n(C)} \text{Var}[\tilde{B}_i - B_i] + \frac{2}{3} \times M\lambda}\right] \leq \\ &2\exp\left[O\left(-\frac{\lambda^2}{\frac{n(C)l^2}{\varepsilon^2(l-1)^2} + \frac{k}{\varepsilon}\lambda}\right)\right] \leq \\ &2\exp\left[O\left(-\frac{\lambda^2}{\frac{n(C)}{\varepsilon^2} + \frac{k}{\varepsilon}\lambda}\right)\right] \\ &\text{令 } \beta = \exp\left[O\left(-\frac{\lambda^2}{\frac{n(C)}{\varepsilon^2} + \frac{k}{\varepsilon}\lambda}\right)\right], \text{ 则存在 } \lambda = \\ &O\left(\frac{\sqrt{n(C)\ln(1/\beta)}}{\varepsilon}\right) \text{ 使 } |\tilde{c}(a) - c(a)| \text{ 至少以 } 1-\beta \text{ 的} \end{aligned}$$

概率成立。

4 实验结果与分析

4.1 实验平台与数据集

实验平台是4核Intel CPU(4GHz), 8 GB内存, Win7系统, 代码采用Python实现。采用Adult、Kosarak、SYN和IPUMS数据集, 其统计信息如表1所示。

表1 数据集统计信息
Table 1 Dataset statistics

数据集	用户数	维度数
Adult	32 561	8
Kosarak	65 536	16
SYN	1 000 000	8
IPUMS	3 210 231	3

为了与现有立方体发布方法进行充分对比分析, 在Full-Pub、Base-Pub和Partial-Pub 3种策略的基础上, 结合Cuboid-OUE、Cuboid-HHT和Cuboid-CMS 3种扰动机制, 设计了FC-OUE、FC-HHT、FC-CMS、BC-HHT、BC-OUE、BC-CMS、PC-HHT、PC-OUE、PC-CMS 9种扰动机制, 并与InputHT、CALM对比响应OLAP查询均方误差, 均方误差的表达式如式(32)所示:

$$\text{MSE}(P(Q)) =$$

$$\frac{1}{|Q_C|} \sum_{C \in Q_C} \left(\frac{1}{|C(a)|} \sum_{a \in C(a)} (\tilde{c}(a) - c(a))^2 \right) \quad (32)$$

其中, Q_C 为查询 Q 所涉及方体的集合, $C(a)$ 为方体 $C \in Q_C$ 中所有单元格 a 的集合。

4.2 实验结果比较

参数 ε 的取值为0.5、1.0、1.5、2.0。从响应方体查询与发布方体的精度进行比较, 每种情况的结果为5轮平均值。

(1) 基于Adult数据集的MSE比较

图3(a)~(d)描述了9种扰动机制与CALM响应方体查询的实验结果。固定维度 $d=8$, 分别响应 $k=3, 4, 5, 6$ 维方体的查询。随着 k 从3变化到6, 所有机制的MSE值均呈下降趋势。图3(e)~(g)描述了9种扰动机制与CALM发布方体的实验结果。CALM发布方体大小以及个数为(2, 28)、(2,

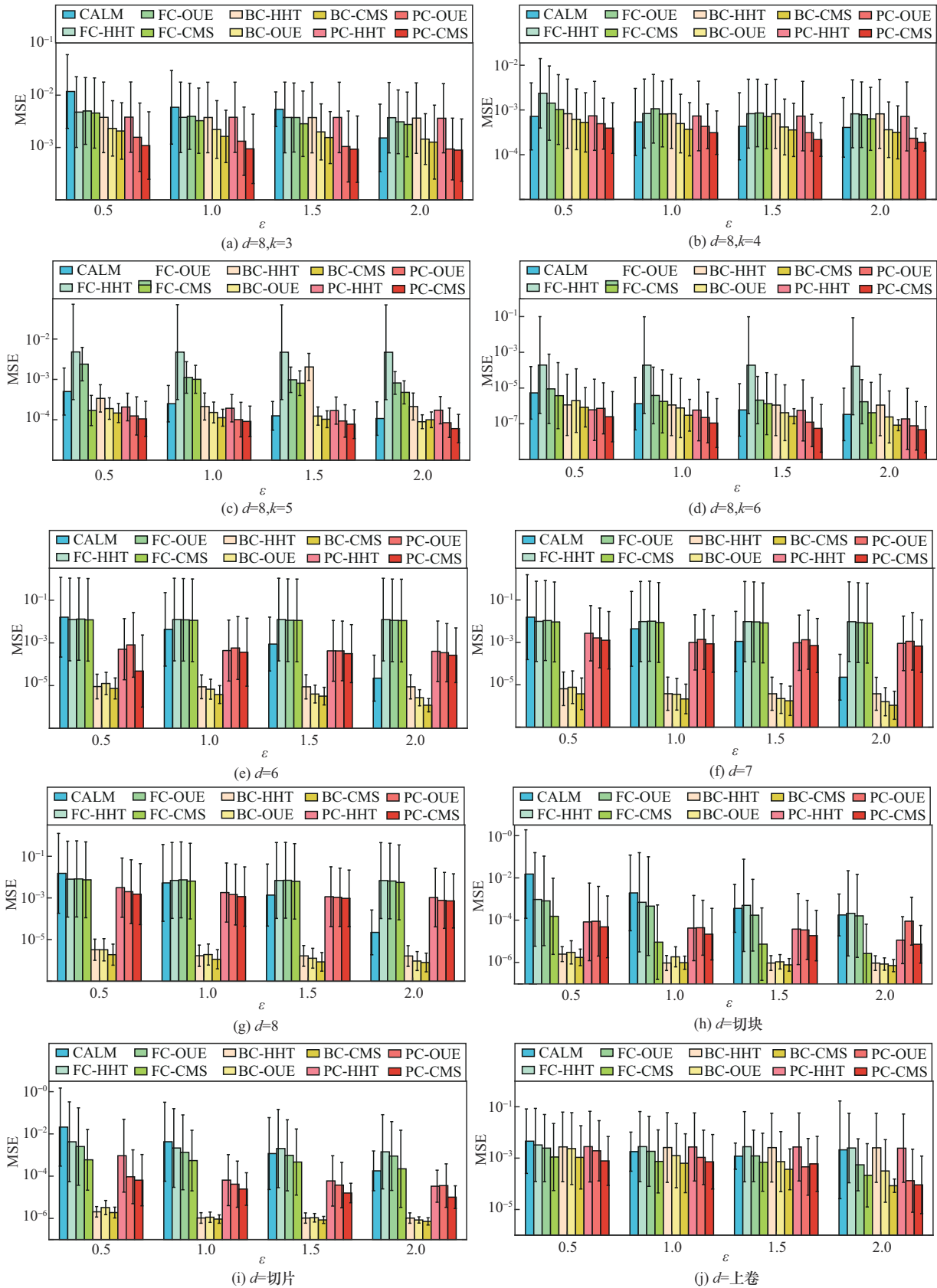


图3 基于 ADULT 数据集响应发布方体查询与 OLAP 操作结果

Figure 3 Results of responding to data cube queries and olap operations based on the ADULT dataset

28)、(2, 28)、(1, 8); FC-HHT、FC-OUE、FC-CMS 发布 2^8 个方体, BC-HHT、BC-OUE、BC-CMS 发布一个基本方体, PC-HHT、PC-OUE、PC-CMS 发布方体数约为总方体的 $1/8$ 。参数 $\varepsilon=0.5, 1.0, 1.5$ 时, BC-HHT、BC-OUE、BC-CMS、PC-HHT、PC-OUE、PC-CMS 在发布方体的性能上优于 CALM; 在 $\varepsilon=2.0$ 时, BC-HHT、BC-OUE、BC-CMS 在发布方体的性能上也优于 CALM。其原因是上述 9 种本地扰动机制均采用用户分组策略来提高立方体发布精度。基于哈达玛矩阵与计数均值草图结构的本地扰动机制能够较好地避免向量的稀疏性。图 3(h)~(j) 描述了 9 种扰动机制响应 OLAP 操作的 MSE 值比较结果。切块是在两个或多个维度上形成的子立方体, 切片是在一个维度上进行的条件查询。上卷是沿着一个或多个维度在数据立方体上聚集, 执行上卷操作的 MSE 值高于切块与切片操作的 MSE 值, 其原因是上卷操作造成了噪音结果的累加, 进而造成 MSE 增加。

(2) 基于 Kosarak 数据集的 MSE 比较

图 4(a)~(d) 描述了 9 种扰动机制与 InputHT、CALM 响应方体查询的 MSE 值比较。同样固定 $d=8$, 响应 $k=3, 4, 5, 6$ 维方体的查询。 ε 从 0.5 变化到 2.0 时, 所有机制的 MSE 值呈下降趋势, 其原因是噪音值与 ε 成反比, ε 值越大噪音值越少。从图 4(a)~(d) 中发现当维度 $d=8$ 且 ε 的取值固定时, 随着 k 值从 3 变化到 6, 所有机制的 MSE 值呈下降趋势, 其原因是随着 k 值增加, 聚集到立方体中的用户数据增加, 进而提升立方体查询精度。图 4(e)~(g) 描述了 9 种扰动机制与 InputHT、CALM 发布方体的 MSE 结果。当维度 $d=8$ 时, 总方体数为 2^8 。InputHT、BC-HHT、BC-OUE 与 BC-CMS 均发布基本方体。PC-HHT、PC-OUE 与 PC-CMS 发布方体个数约为总方体数的 $1/16$; CALM 发布方体需能响应 $k=8$ 的方体查询。 ε 从 0.5 变化到 2.0, CALM 所发布方体大小以及个数分别为 (2, 28)、(2, 28)、(2, 28)、(3, 56)。FC-HHT、FC-OUE 和 FC-CMS 机制发布 2^8 个方体。从实验结果可知, BC-OUE 与 BC-CMS 发布方体的性能优于 InputHT 机制, PC-OUE 与 PC-CMS 发布方体的性能优于 CALM。其原因是 BC-OUE

与 BC-CMS 仅发布基本方体, 而 InputHT 仅能处理二进制属性, 处理其他属性时会增加维度, 进而导致 MSE 增加。FC-HHT、FC-OUE 和 FC-CMS 发布全方体导致发布精度最低。图 4(h)~(j) 描述了 9 种扰动机制与 InputHT 响应切块、切片以及上卷操作的 MSE 值结果。随着 ε 增加, 所有算法的 MSE 值呈递减趋势, 其原因是 ε 增加导致噪声减少。所有算法在切片与切块操作上的精度相似, 因为这两种操作均是给定维度的条件查询。BC-CMS 在任何 OLAP 操作上的 MSE 均优于 InputHT。

(3) 基于 SYN 数据集的 MSE 比较

图 5(a)~(d) 描述了 9 种扰动机制与 CALM 响应方体查询角度对比的实验结果。固定 $d=8$ 来响应 $k=3, 4, 5, 6$ 维方体的查询。 k 从 3 变化到 6 时, 所有机制的 MSE 值均呈下降趋势。图 5(e)~(g) 描述了 9 种扰动机制与 InputHT 响应切块、切片以及上卷操作的 MSE 值的比较结果。从 SYN 中随机采样维度来执行 OLAP 操作。随着 ε 增加, 所有机制的 MSE 值同样呈下降趋势, 其原因是大的 ε 值产生较小的噪音值。BC-CMS 在任何 OLAP 操作上的 MSE 结果优于 InputHT。图 5(h)~(j) 描述了 9 种扰动机制响应切块、切片以及上卷操作的 MSE 结果。BC-HHT、BC-OUE 与 BC-CMS 在切片与切块操作上的 MSE 结果优于其他机制, PC-CMS 在上卷操作上的 MSE 结果表现最优。其原因是 PC-CMS 采用最优搜索技术与计数均值草图生成部分方体来响应 OLAP 操作。

(4) 基于 IPUMS 数据集的 MSE 比较

图 6(a)~(c) 描述了 9 种扰动机制与 CALM 发布方体的对比结果。由于 IPUMS 数据仅包含 3 个属性, 该组实验从每个属性值域大小进行对比 MSE。从每个属性值域中随机抽取 3 个维度基数 $c \in \{2, 3, 4\}$ 。随着 ε 的取值增加, 维度基数从 2 变化到 4 时, 所有机制的 MSE 值整体呈下降趋势。基于计数均值草图的扰动机制明显优于 CALM, 其原因是 IPUMS 数据属性的稀疏性, 从而 CALM 的发布精度下降。图 6(d)~(f) 描述了上述 10 种机制响应切块、切片与上卷操作的 MSE 值比较。在 $d=3, c=4$ 时, 随着 ε 增加, 所有机制的 MSE 值都呈递减趋势, 其中 FC-CMS、BC-CMS 与 PC-CMS 的 OLAP 操作精度优于 CALM。

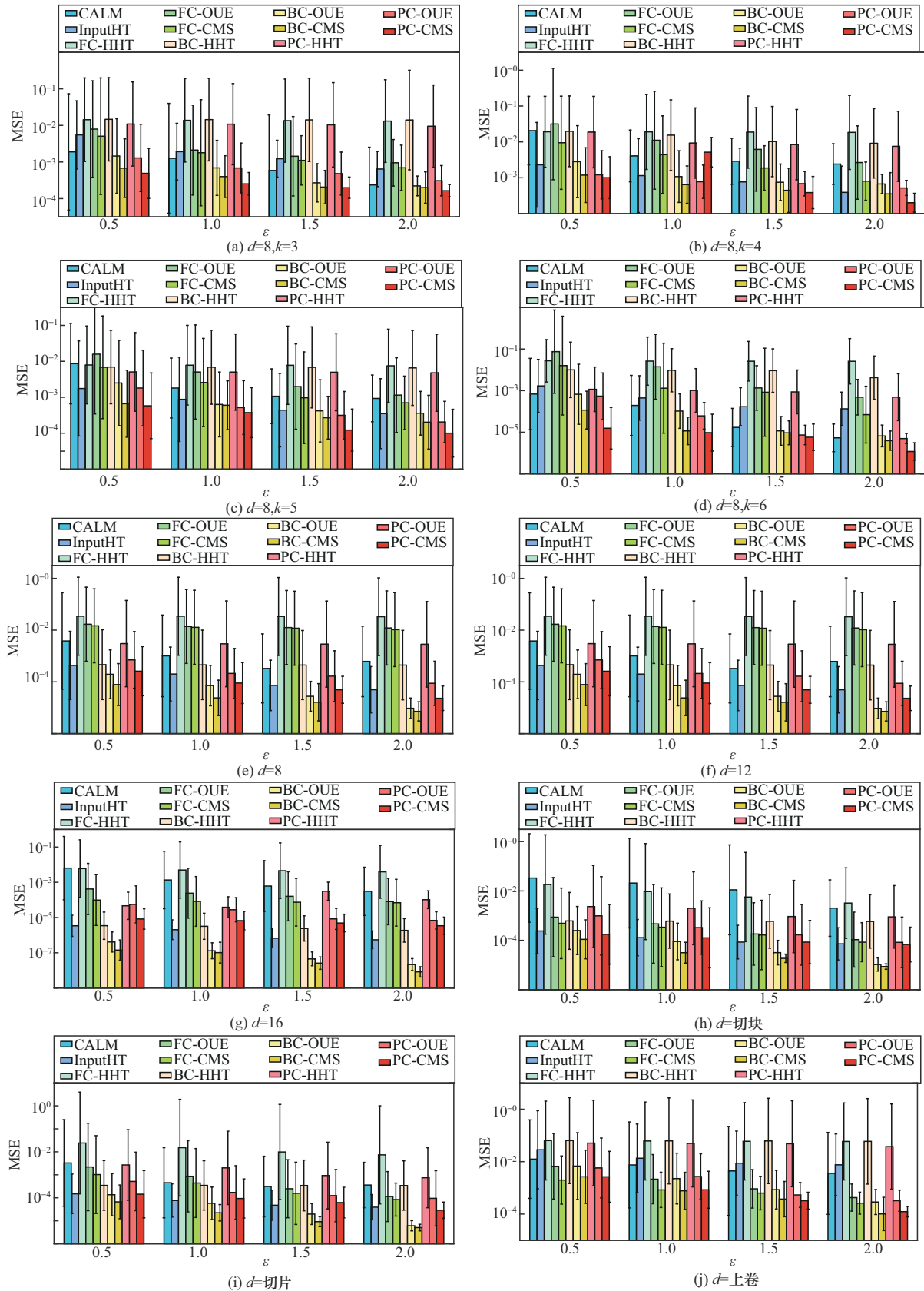


图4 基于 Kosarak 数据集响应发布方体查询与 OLAP 操作结果

Figure 4 Results of cube query and OLAP operations based on the Kosarak dataset

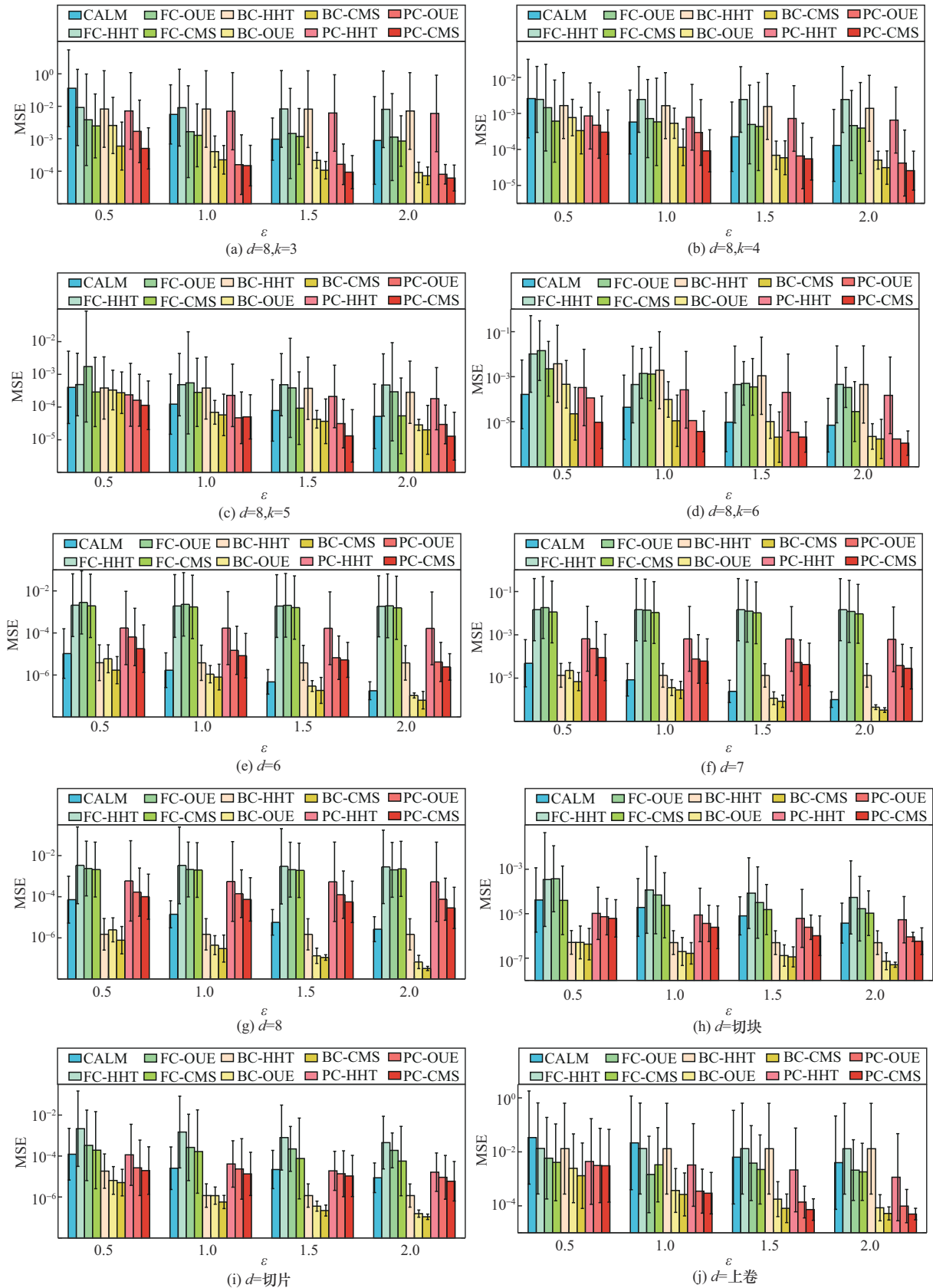


图5 基于SYN数据集响应发布方体查询与OLAP操作结果

Figure 5 Results of responding to data cube queries and OLAP operations based on the SYN dataset

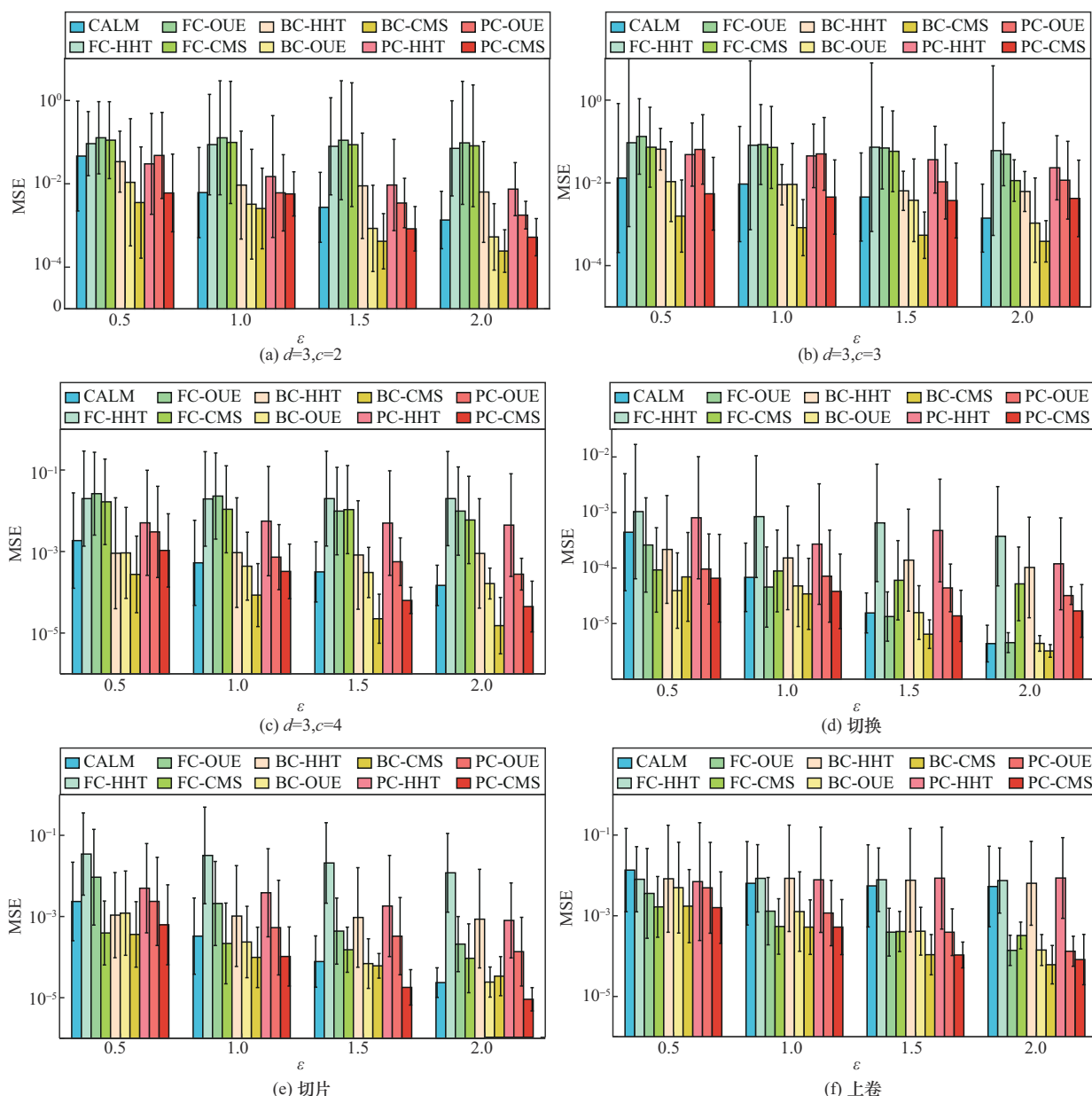


图6 基于IPUMS数据集响应发布方体查询与OLAP操作结果
Figure 6 Results of responding to data cube queries and OLAP operations based on the IPUMS dataset

5 结束语

针对数据立方体的发布问题,提出了3种基于本地差分隐私的数据立方体发布策略。3种策略分别从全方体、基本方体以及部分方体的角度进行方体发布。并在此基础上设计了3种本地扰动机制Cuboid-OUE、Cuboid-HHT和Cuboid-CMS用于保护用户的原始数据。收集者根据用户的报告值构建噪音方体,并对其发布。从理论角度分析了本文算法的隐私性与可用性,并从实验角度验证了它们优于同类算法。

参考文献:

- [1] DUCHI J C, JORDAN M I, MARTIN J. Wainwright: local privacy and statistical minimax rates[Z]. 2013: 429-438.
- [2] CHATTERJEE S, DUCHI J C, LAFFERTY J D, et al. Local minimax complexity of stochastic convex optimization[C]// NIPS, 2016: 3423-3431.
- [3] CORMODE G, KULKARNI T, SRIVASTAVA D. Marginal release under local differential privacy[C]// SIGMOD, 2022. 131-146.
- [4] ZHANG Z K, WANG T H, LI N H, et al. CALM: Consistent adaptive local marginal for marginal release under local differential privacy[C]// CCS, 2018: 212-229.

- [5] 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.
- [6] ERLINGSSON U, PIHUR V, KOROLOVA A. RAPPOR: randomized aggregatable privacy-preserving ordinal response[C]//CCS, 2014: 1054-1067.
- [7] KAIROUZ P, BONAWITZ, K A, RAMAGE D. Discrete distribution estimation under local privacy[C]//ICML, 2016: 2436-2444.
- [8] WANG T, BLOCI J, LI N, et al. Locally differentially private protocols for frequency estimation [C]//Proc of the 26th USENIX Security Symposium (S&P), 2017: 729-745.
- [9] DING B, WINSLETT M, HAN J W, et al. Differentially private data cubes: optimizing noise sources and consistency[C]//SIGMOD Conference, 2011: 217-228.
- [10] DWORK C. Differential privacy[C]// Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006), Springer Verlag, 2006.
- [11] DWORK, C, JING L. Differential privacy and robust statistics[C]// Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC 2009), 2009: 71-380.
- [12] DWORK C. Differential privacy in new settings[C]//Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), Society for Industrial and Applied Mathematics, 2010: 174-183.
- [13] QARDAJI W, YANG W N, LI N H. PriView: practical differentially private release of marginal contingency tables[C]//Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2014: 1435-1446.
- [14] ZHANG Z K, WANG T H, LI N H, et al. PrivSyn: Differentially private data synthesis[C]//USENIX Security Symposium, 2021: 929-946.
- [15] YANG M M, GUO T L, ZHU T Q, et al. Local differential privacy and its applications: A comprehensive survey[J]. Computer Standards & Interfaces, 2024, 89: 103827.
- [16] ACHARYA J, SUN Z T, ZHANG H Y. Hadamard response: estimating distributions privately, efficiently, and with little communication[C]//AISTATS, 2019: 1120-1129.
- [17] ACHARYA J, SUN Z T, ZHANG H Y. Communication efficient, sample optimal, linear time locally private discrete distribution estimation[J]. arXiv preprint arXiv:1802.04705.
- [18] BARAK B, CHAUDHURI K, DWORK C, et al. Privacy, accu-

racy, and consistency too: a holistic solution to contingency table release[C]//PODS, 2007: 273-282.

- [19] BALLE B, BELL J, GASC'ON A. The privacy blanket of the shuffle model[C]//CRYPTO, 2019: 638-667.

- [20] Differential Privacy Team. Learning with privacy at scale, Apple Inc[EB].2017.

[作者简介]



张啸剑 (1980-), 男, 河南周口人, 博士, 河南财经政法大学教授、硕士生导师, 主要研究方向为差分隐私与机器学习、AI增强型数据管理、图数据管理。



王浩锋 (1999-), 男, 河南许昌人, 硕士, 主要研究方向为差分隐私。



曹小杰 (1999-), 女, 河南周口人, 硕士, 主要研究方向为差分隐私。



傅继彬 (1975-), 男, 河南许昌人, 博士, 河南财经政法大学副教授、硕士生导师, 主要研究方向为人工智能、计算机视觉。