PRIMA: Privacy preserving Multi-dimensional Analytic Approach

ABSTRACT

Sum query is an important and fundamental operator for online analytical processing. In this paper, we focus on the process of answering sum queries over data cube, each of which consists of a collection of cuboids, while satisfying differential privacy (DP). Existing works fail to process the sum queries in online analytical processing with high utility due to sum queries' high sensitivity and the noise aggregation: constructing a base cuboid requires the data curator to answer a workload of linear sum queries under DP in advance, whose sensitivity will result in a large amount of DP noise, and the noise will finally be aggregated when constructing the remaining cuboids. To this end, we present a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA). In PRIMA, we propose a Symmetric Bounded Sum Query Processing Method (SBS) which reduces the sensitivity of sum queries by bounding both the maximum and minimum contribution of each record in the data table in a symmetricaly manner. Moreover, we propose a Hypothesis Testing based Prefix Sum Computing Method (SCOPE) to compute a base prefix-sum cuboid based on hypothesis testing. By employing the base prefix-sum cuboid, any remaining cuboid can be constructed with constant pieces of DP noise aggregated. We conduct experiments on both real-world and synthetic datasets. Experimental results confirm the effectiveness of PRIMA over existing works.

KEYWORDS

Differential privacy, Sum queries, Data cube, Prefix sum

1 INTRODUCTION

In On-Line Analytical Processing (OLAP), sum query is an important and fundamental operator. By constructing a data cube that consists of a collection of cuboids in advance, sum queries can be answered over massive amounts of data, which are stored in a data warehouse, quickly and efficiently. In the state-of-the-art analytical data warehouse for big data, the data curator first computes a base cuboid based on the data table via pre-querying. By aggregating the base cuboid in different dimensions, the data curator then constructs the remaining cuboids. Finally, any sum queries can be answered over the constructed data cube. However, since the data table usually contains private data, constructing data cubes based on such data may endanger the privacy of individuals. Therefore, the data curator must provide a rigorous privacy guarantee on how sum queries in OLAP are processed.

Differential Privacy (DP) has been increasingly accepted as the state-of-the-art standard for protecting individual privacy in the central setting where a trusted data curator processes the sensitive data of all users, and has been studied extensively in both literature [12, 16, 17, 23, 24, 34, 40, 41] and industry [3, 13, 15, 26, 36]. DP guarantees that the presence or absence of any particular individual's record has a negligible impact on the likelihood that a particular result is returned to a query. Thus, an adversary cannot make meaningful inferences about any individual's record value, or even whether the record is present.

In this paper, we first focus on the processing of answering sum queries over data cubes under DP. Since sum queries typically have large sensitivities, answering sum queries under DP directly will lead to excessive noise. A trivial approach for processing sum queries with both privacy and utility guarantee is bounding users' maximum contribution with a fixed threshold []. However, this approaches faces a bias-variance trade-off: a loose bounding threshold cannot reduce the DP noise caused by the high sensitivity; while a tight bounding threshold will inevitably introduce systematic bias into the query results. Several existing works [] attempt to handle this problem by bounding both users' maximum and minimum contributions. By setting a upper threshold, the sensitivity of sum queries is limited and less DP noise is added; by setting a lower threshold, the systematic bias in query results is alleviated. However, in practical applications, these methods fail to identify an effective lower threshold. Unlike the upper threshold which is carefully determined by using elaborate algorithms, the lower threshold is usually determined in a trivial way. Specifically, the lower thresholds in these works are typically set to 0.

Besides the high sensitivity of sum queries, the noise aggregation is another tough problem. To limit the DP noise added to the cuboids when constructing remaining cuboids, Ding et al. [6] propose a Noise Control Framework, which formulates an optimization problem and minimizes the utility loss. However, this framework cannot cope well with the problem of noise aggregation, i.e., the problem that the DP noise in each cell of the base cuboid will be aggregated when constructing the remaining cuboids. As the domains of attributes increase, the numbers of noisy cells aggregated in the Noise Control Framework will grow in polynomial degrees, resulting in massive DP noise aggregated in the constructed cube.

Both the bias-variance trade-off in answering sum queries and the noise aggregation in constructing data cubes significantly affect the accuracy of queries' results. To deal with these problem, we propose a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA). Its main idea is that bounding users' contribution in a symmetric manner and processes sum queries based on the prefix-sum cube which is constructed via hypothesis testing.

In PRIMA, we first put forward a **S**ymmetrical **B**ounded **S**um Query Processing Method (SBS) for answering sum queries under DP while alleviating the systematic bias. Different from the existing works that determines the upper and lower threshold using different algorithm, SBS carefully chooses a symmetric bounding range $[\Phi - \tau, \Phi + \tau]$ and rewrites the data table. In SBS, the data curator first formulates optimization problems for choosing the optimal bounding axis Φ and radio τ and then rewrites the records whose measures' value are out of the bound range $[\Phi - \tau, \Phi + \tau]$.

Next, for constructing the remaining cuboids accurately, the data curator alleviates noise aggregation by constructing the prefix-sum cubes [37]. Unfortunately, computing the prefix sums require multiple accesses to any single record in the data table, which will introduce massive DP noise. Therefore, we design a Hypothe**S**is Testing based Prefix Sum **C**o**P**uting M**E**thod (SCOPE), which avoids multiple accesses to the records and computes the prefix

sums based on hypothesis testing. In SCOPE, the data curator first acquires a noisy sum vector by accessing each record once under DP, then predicts the distribution of the noisy sum vector based on hypothesis testing. Next, the data curator computes the prefix sums based on both the noisy sum vector and its predicted distribution. Finally, the data curator boosts the accuracy of the prefix sums by enforcing consistency. Based on the prefix sums, the data curator can compute a prefix sum base cuboid and constructs the remaining cuboids without noise aggregation.

The main contributions of this work are summarized as follows:

- We focus on the process of answering sum queries over data cubes under DP and present PRIMA which can bound the sensitivity of sum query while minimizing the systematic bias and alleviate noise aggregation when constructing data cubes under DP.
- We put forward a **S**ymmetrical **B**ounded **S**um Query Processing Method (SBS). By rewrites the records whose measures' value are out of the bound range $[\Phi - \tau, \Phi + \tau]$, SBS can answer sum queries under DP accurately.
- We design a Hypothe**S**is Testing based Prefix Sum **CO**m**P**uting M**E**thod (SCOPE). By computing prefix sums based on hypothesis testing and boosting the accuracy of prefix sums by enforcing consistency, SCOPE can improve the utility of the noisy prefix-sum cube and alleviate the problem of noisy aggregation.
- To evaluate the performance of PRIMA, we conduct extensive experiments on both real-world and synthesized datasets. The experimental results demonstrate that PRIMA outperforms the baseline approaches.

2 RELATED WORK

Since DP [11] was introduced, many works[10, 14, 21, 28, 29, 31, 42] have been proposed for preserving individuals' privacy under DP during answering queries. However, these approaches are all designed for processing linear count queries over a single relation with different predicates, which have a relatively small sensitivity. Different from the count queries, sum queries and average queries possess higher sensitivity. For answering sum queries, several works [1, 7, 22, 25, 33] bound the sensitivity by restricting the maximum contribution of each record, which rewrite the records whose attributes' values are more than a threshold τ before adding DP noise. Specifically, some of them [7, 22, 33] adaptively choose the threshold in a bottom-up manner. Others [1, 25] try to find the threshold which can strike a balance between bias and variance by formulating an optimization problem. Unfortunately, none of these works can alleviate DP noise and systematic bias simultaneously. Different from above works, Wilson et al. [38] attempt to bound both the maximum and minimum contribution of each user. However, in practice, the lower threshold is often set to a trivial value close to zero, which nearly fails to alleviate systemic bias. There are also a batch of works [4, 8, 20] consider how to answering average queries with high utility. Huang et al. [20] propose a binary search based algorithm to find a upper threshold that optimizes the bias-variance trade-off. Bun et al. [4] propose a Private Mean Estimation Method which discards the largest m records and the smallest m records from data table and then computes the mean of the remaining samples. Targeting the average queries over an infinite domain, Dong et al.[8] propose a empirical estimators which clip the domain in a symmetrical manner. However, thisi method cannot be extended to answering sum queries over a finite domain.

Since the interactive-based approaches can only process a limited number of count queries under a fixed privacy budget. A series of approaches [6, 18, 27, 32, 39] based on the non-interactive strategy are proposed. In these approaches, the data curator first constructs a differentially private data model that contains certain statistical information about the data table via a set of queries. Any subsequent queries can be then processed over the differentially private data model without spending a privacy budget. In the non-interactive strategy, noise aggregation is a major problem that leads to poor utility. To answer queries with both privacy and utility guarantees, Hay et al. [18] propose an approach that is able to deal with noise aggregation in single-dimensional range queries via hierarchical intervals. Qardaji et al. [32] optimize the hierarchical intervals by choosing a proper branching factor. Targeting the same problem, Xiao et al. [39] develop an approach called Privelet that provides accurate answers by using Haar wavelet. However, the data models in these approaches are limited to single-dimensional analysis. To tackle this problem, Ding et al. [6] present a general Noise Control Framework which carefully selects a subset of cells in a cuboid to inject Laplace noise and compute the rest cells from them. However, when the domains of attributes are large, the noise aggregation in Noise Control Framework will also cause a poor utility of query results.

3 PRELIMINARIES

3.1 Differential Privacy (DP)

DP requires the outputs of algorithms to be approximately the same even if any individual's record in the database is added or removed. Thus, the presence or absence of any individual's record has a statistically negligible effect on the outputs.

Formally, DP considers an instance data table T of relational schema $R(\mathbb{A}, M)$. The distance between two tables $T, T' \in D^n$ is $d(T, T') = |\{i | t_i \neq t'_i\}|$. Two tables T, T' are neighboring tables if $d(T, T') = 1$.

Definition 3.1 (Differential Privacy). A private algorithm $\mathcal{A}$ preserves ϵ-differential privacy is for any pair of neighboring tables T, T', and for all sets O of possible outputs:

$$Pr[\mathcal{A}(T) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(T') =\in O] + \delta,$$

where the probability is taken over the randomness of $\mathcal{A}$.

Dwork et al. [11] propose the Laplace mechanism to achieve DP. For a function whose outputs are real, they prove that DP can be achieved by adding noise drawn randomly from Laplace distribution. The Laplace distribution with magnitude γ follows probability density function $Pr[Lap(\gamma) = x] = \frac{1}{2\gamma} e^{-|x|/\gamma}$, where $\gamma = \frac{S}{\epsilon}$ is determined by the desired privacy budget ϵ and the sensitivity S of the function. In particular, the sensitivity S is used to measure the maximum change in the outputs of a function when any individual's record in the data table is changed.

One measure of sensitivity is the global sensitivity [11], which is the maximum difference in the query's result on any two neighboring tables. However, for many functions, the global sensitivity yields high noise, which leads to a poor utility of query results. To this end, a local measure of sensitivity called local sensitivity [10] is proposed, which is the maximum difference between the query's results on a given table and any neighboring table of it.

Definition 3.2 (Local Sensitivity). For an algorithm $\mathcal{A}$ and an instance data table T of relational schema $R(\mathbb{A}, M)$, the local sensitivity of $\mathcal{A}$ at T is

$$LS_{\mathcal{A}} = \max_{y:d(T,T')=1} \left\| \mathcal{A}(T) - \mathcal{A}(T') \right\|.$$

Since the local sensitivity is a property of the given table rather than the set of all possible tables, it is much often much smaller than global sensitivity. However, the local sensitivity is not safe enough and the corresponding noise magnitude may reveals information about the database. [29]. In this paper, for utilizing local sensitivity safely, we choose to release the noisy query via Propose-Test-Release (PTR) [11].

Definition 3.3 ((ϵ, δ)−Propose-Test-Release). Let $\mathcal{A}$ be the function of interest, which we wish to privately query on datasets T. A (ϵ, δ)−Propose-Test-Release consists of the following three steps:

1. Propose an upper bound β on the local sensitivity.
2. Compute the distance from T to the nearest dataset T' such that $\Delta_{LS}(T') \geq \beta$, name it γ. Distance is measured in terms of how many records must be changed to get from T to T'.
3. Compute $\hat{\gamma} = \gamma + Lap(1/\epsilon)$.
4. If $\hat{\gamma} \leq \ln(1/\delta)/\epsilon$, return $\perp$.
5. If $\hat{\gamma} > \ln(1/\delta)/\epsilon$, return $\mathcal{A}(T) + Lap(\beta/\epsilon)$.

In our work, all queries are processed via PTR to avoid the potental privacy leakage.

To support multiple differentially private queries, the sequential and parallel compositions of DP are extensively used.

Theorem 3.4 (Sequential Composition). *Given k random algorithms $\mathcal{A}_i (1 \leq i \leq k)$ that access the same record t, each of which satisfies (ϵ_i, δ_i)-DP, then the combination of their outputs satisfies (ϵ, δ)-DP, where $\epsilon = \left(\sum_{i=1}^{k} \epsilon_i\right)$, $\delta = \left(\sum_{i=1}^{k} \delta_i\right)$.*

Theorem 3.5 (Parallel Composition). *Given k random algorithms $\mathcal{A}_i (1 \leq i \leq k)$ that access different records $t_i (1 \leq i \leq n)$, each of which satisfies (ϵ_i, δ_i)-DP, then the combination of their outputs satisfies (ϵ, δ)-DP, where $\epsilon = \max(\epsilon_1, \cdots, \epsilon_k)$, $\delta = \max(\delta_1, \cdots, \delta_k)$.*

3.2 Data Cube

Data cube is a combination of cuboids. For a data table T which is an instance of relational schema $R(\mathbb{A}, M)$, a data cuboid can be viewed as the projection of T on a subset of attributes $\mathbb{B} \subseteq \mathbb{A}$, producing a multi-dimensional array. In the cuboid, each cell aggregates the measure M of records in T that match certain dimensions. Given a data cube among d attributes in T, there are $\binom{d}{\lambda}$ λ-dimensional ('λ-D') cuboids ($0 \leq \lambda \leq d$) in a data cube, representing all possible combinations of attributes in T. In these cuboids, the only d-D cuboid is called a base cuboid. Any other cuboids can be acquired by aggregating the base cuboid. In addition, for any two cuboids C_1 and C_2, if C_1 can be constructed from C_2 by directly aggregating one or more attributes in C_2, then C_1 is said to be an ancestor of C_2, and C_2 is a descendant of C_1. We show an example of constructing a data cube based on a table T in Fig. 1.

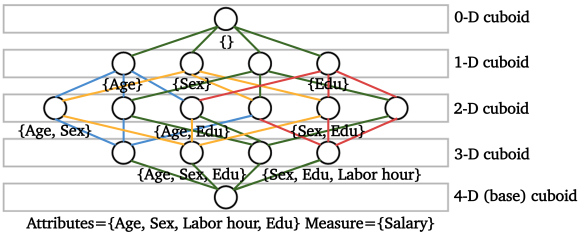

Figure 1: Data Cube

Example 3.6. Suppose there is a measure *Salary* and a subset of attributes $\mathbb{A} = \{a_1, a_2, a_3, a_4\}$, where a_1, a_2, a_3, and a_4 refer to age, gender, labor hours, and educational level, respectively. To analyze the relationship of workers' salary with their age, gender, labor hours, and educational level, the data curator can construct a data cube among these four attributes which aggregates the *Salary*. The data cube contains $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 16$ cuboids, in which the 4-D cuboid $C_{a_1 a_2 a_3 a_4}$ is the base cuboid. In particular, for the 2-D cuboid $C_{a_1 a_2}$, the 3-D cuboid $C_{a_1 a_2 a_4}$ is the descendant of $C_{a_1 a_2}$ while $C_{a_2 a_3 a_4}$ is not, since $C_{a_1 a_2}$ cannot be constructed by aggregating $C_{a_2 a_3 a_4}$.

3.3 Prefix-Sum Cube

Given λ attributes, a prefix-sum cuboid [19] is a λ-D array that has the same structure as the data cuboid among these attributes. The cells in the prefix-sum cuboid contain a kind of auxiliary information called prefix sum [19], which can be computed from the distributions among λ attributes. Similarly, based on the base prefix-sum cuboid, the remaining cuboids can be constructed through the computation between two cells in the base prefix-sum cuboid. Since each cell in the prefix-sum cuboid contains a piece of DP noise, which is introduced when computing the base prefix-sum cuboid under DP, there are only constant pieces of noise being aggregated, irrespective of the domain of the aggregated attributes.

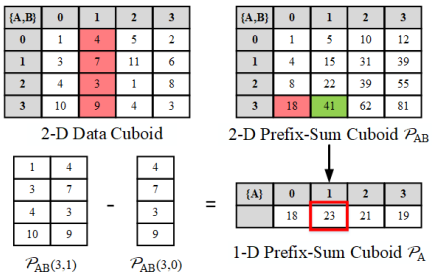

Figure 2: Computing a cell in an ancestor cuboid based on $\mathcal{P}$

Suppose there is a λ-D cuboid A of size $dom(A_1) \times dom(A_2) \times \cdots \times dom(A_\lambda)$. Prefix-sum cuboid $\mathcal{P}$ is another λ-d array which is used to store various precomputed prefix sums of A. For all

$0 \le x_j < dom(A_j)$ and $j \in [0, \lambda - 1]$,

$$\mathcal{P}[x_1, x_2, \cdots, x_\lambda] = Sum(0 : x_1, 0 : x_2, \cdots, 0 : x_\lambda) = \sum_{i_1=0}^{x_1} \cdots \sum_{i_\lambda=0}^{x_\lambda} A[i_1, \cdots, i_\lambda]. \quad (1)$$

Theorem 3.7 provides how a cell in an ancestor cuboid can be computed from up to 2^λ appropriate cells in the prefix sum base cuboid $\mathcal{P}$. The left-hand side of Eq. 2 specifies a cell to be computed. The right-hand side of Eq. 2 consists of 2^λ additive terms, each of which is from a cell of $\mathcal{P}$ with a sign '+' or '-' defined by the product of all $s(i)$. For notational convenience, let $\mathcal{P}[x_1, x_2, \cdots, x_\lambda] = 0$ if $x_j = -1$ for some $j \in [0, \lambda - 1]$.

Theorem 3.7. *For all $j \in [0, \lambda - 1]$, let*

$$s(j) = \begin{cases} 1, & if\, x_j = h_j \\ -1, & if\, x_j = l_j - 1 \end{cases}.$$

Then, for all $j \in [0, \lambda - 1]$,

$$Sum(l_1 : h_1, \cdots, l_d : h_d) = \sum_{\forall x_j \in \{l_j - 1, h_j\}} \left\{ \left[\prod_{i=1}^{d} s(i) * \mathcal{P}[x_1, x_2, \cdots, x_d] \right] \right\}. \quad (2)$$

We show an example of how to compute a cell in an ancestor cuboid based on a prefix-sum cuboid in Example 2.

Example 3.8. Fig. 2 shows an example of a prefix-sum cuboid $\mathcal{P}$ with $\lambda = 2$, $dom(M_A) = dom(M_B) = 4$, and the geometrical explanation for the process of computing the value of a certain cell in the 1-D prefix-sum cuboid $\mathcal{P}_A$ based on the 2-D prefix-sum cuboid $\mathcal{P}_{AB}$. Suppose there is a 2-D prefix-sum cuboid $\mathcal{P}_{AB}$ that is computed from the 2-D data cuboid between A and B, a certain cell in the 1-D prefix-sum cuboid $\mathcal{P}_A(1) = Sum(0 : 1, 0 : 3) - Sum(0 : 0, 0 : 3) = \mathcal{P}_{AB}(3, 1) - \mathcal{P}_{AB}(3, 0)$.

3.4 Linear Sum Queries and Query Matrix

The base cuboid can be constructed by answering a set of linear sum queries. Specifically, given a table T which is an instance of relational schema $R(\mathbb{A}, M)$, where $\mathbb{A}$ is a set of d attributes and M is a measure, we focus on the following class of multi-dimensional sum queries:

```
SELECT SUM(M) FROM T WHERE
A1 ∈ [l1, r1] AND ... AND Ad ∈ [ld, rd].
```

By denoting the cross-product of the domains of attributes in $\mathbb{A}$ as $dom(\mathbb{A})$. The data curator chooses a set of attributes $\mathbb{B} \subseteq \mathbb{A}$ relevant to the base cuboid. For example, if the data analyst is interested in a subset of two dimensional sum queries over attributes A_1 and A_2, they would set $\mathbb{B} = \{A_1, A_2\}$. The data curator then forms a sum vector $\mathbf{x}$ with one entry for each element of $dom(\mathbb{B})$. For simplicity, we assume $dom(\mathbb{B}) = \{1, 2, ..., n\}$ and for each $i \in dom(\mathbb{B})$, x_i is the sum of records' measures that satisfy the constraints that $|\mathbb{B}| = i$. We represent $\mathbf{x}$ as a column vector of sums: $\mathbf{x} = [x_1, x_2, ..., x_n]^t$. A linear query computes a linear combination of the counts in $\mathbf{x}$.

Definition 3.9 (Linear Sum Query). A linear query is a length-n row vector $\mathbf{q} = [q_1, ..., q_n]$ with each $q_i \in \mathbb{R}$. The answer to a linear sum query $\mathbf{q}$ on $\mathbf{x}$ is the vector product $\mathbf{qx} = q_1 x_1 + ... + q_n x_n$.

A series of linear sum queries can be organized into the rows of a pre-query matrix [23, 24] whose definition is given as follows:

Definition 3.10 (Pre-Query Matrix). A pre-query matrix is a collection of m linear sum queries, arranged by rows to form an $m \times n$ matrix.

If $\mathbf{Q}$ is an $m \times n$ pre-query matrix, the query answer for $\mathbf{Q}$ is a length m column vector of query results, which can be computed as the matrix product $\mathbf{Qx}$. The sensitivity of $\mathbf{Q}$ [23, 24] can be computed as follows.

Proposition 3.11 (Pre-Query Matrix Sensitivity). *The sensitivity of matrix $\boldsymbol{Q}$, denoted as $\Delta_{\boldsymbol{Q}}$, is*

$$\Delta_{\boldsymbol{Q}} = \max_{\|\boldsymbol{x} - \boldsymbol{x'}\|=1} \|\boldsymbol{Qx} - \boldsymbol{Qx'}\|_1 = \max_j \sum_{i=1}^{n} |q_{ij}|.$$

Thus the sensitivity of a pre-query matrix is the maximum L_1 norm of a column.

We show the pre-query matrices for computing a data cuboid and a prefix-sum cuboid in Example 3.

Example 3.12. Fig. 3 shows two query matrices. Given a subset of attributes $\mathbb{B} = A, B$, where $dom(M_A) = dom(M_B) = 4$, $dom(\mathbb{B}) = dom(M_A) \times dom(M_B) = 16$. $\mathbf{I}_{16}$ is the pre-query matrix for computing a 2-D data cuboid, and $\mathbf{P}_{16}$ is the pre-query matrix for computing a 2-D prefix-sum cuboid. Specifically, for a data table where the sensitivity of the sum query is LS_f, $\mathbf{I}_{16}$ is an identity matrix of size 16×16. This matrix consists of 16 queries and each row of $\mathbf{I}_{16}$ represents a linear sum query that asks for the value of a cell in the data cuboid. According to Proposition 3.11, the sensitivity of each linear sum query in $\mathbf{I}_{16}$ is LS_f. $\mathbf{P}_{16}$ is a lower triangular matrix of size 16×16, which consists of 16 queries, and each row of $\mathbf{P}_{16}$ represents a linear sum query that asks for the value of a cell in the prefix-sum cuboid. According to Proposition 3.11, the sensitivity of each linear sum query in $\mathbf{P}_{16}$ is $16 \cdot LS_f$.

$$\begin{bmatrix} LS_f & 0 & \dots & 0 \\ 0 & LS_f & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & LS_f \end{bmatrix} \quad \begin{bmatrix} LS_f & 0 & \dots & 0 \\ LS_f & LS_f & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ LS_f & LS_f & \dots & LS_f \end{bmatrix}$$

(a) $\mathbf{I}_{16}$ (b) $\mathbf{P}_{16}$

Figure 3: Two Query Matrices

4 SOLUTION

In this section, we present PRIMA for answering multi-dimensional analytic queries over data cubes under DP.

4.1 Overview

As shown in Section 2, the existing approaches fail to reduce the systematic bias while bounding the high sensitivity of sum queries and alleviate the problem of noise aggregation. To this end, we present a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA). PRIMA contains two key building blocks: **S**ymmetrical **B**ounded **S**um Query Processing Method (SBS) and Hypothe**S**is Testing based Prefix Sum **C**om**P**uting M**E**thod (SCOPE). We show

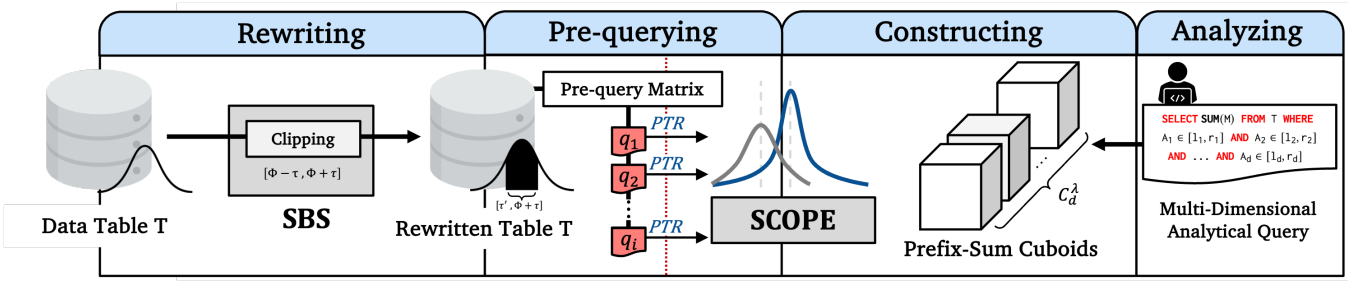

Figure 4: The overview of PRIMA

the overview of PRIMA in Fig. 4 and introduce the main phases of PRIMA as follows:

1. Data Rewriting Phase. Given a class of sum queries that associate with d attributes and a measure M. The data curator first determines the optimal bounding axis Φ and bounding radius τ via SBS, which forms the bounding range $[\Phi-\tau, \Phi+\tau]$. Then, the data curator traverses the data table and clips the measure M of each record whose value is beyond the bounding range $[\Phi-\tau, \Phi+\tau]$. The details of SBS are shown in Section 4.2.

2. Pre-querying Phase. The data curator then organizes a workload of linear sum queries into a pre-query matrix to acquire the noisy sum vectors for constructing the base prefix-sum cuboid. Since the sum queries in the pre-query matrix access different records in the table, according to the parallel composition, a piece of privacy budget ϵ can be shared when answering these queries. To prevent the privacy leakage, all queries are processed via PTR.

3. Cuboid Constructing Phase. Based on the noisy sum vectors, the data curator computes the base prefix-sum cuboid via SCOPE and constructs the remaining prefix-sum cuboids by aggregating the base prefix-sum cuboid in different dimensions. The details of SCOPE are given in Section 4.3.

4. Query Processing Phase. Finally, based on the constructed data cube, the sum queries can be processed privately and efficiently.

In the following, we first introduce SBS in Section 4.2, and then give the details of SCOPE in Section 4.3. Finally, we show the privacy guarantee and utility analysis of PRIMA in Section **??**.

4.2 Symmetrical Bounded Sum Query Processing Method

The high sensitivity of the sum query is the underlying cause of the poor result utility. A series of works [7, 22] bound the sensitivity of sum queries by bounding users' maximum contribution. Given records $\{t_i | 1 \le i \le n\}$ in a data table T, if each record t_i has a measure whose measure's value is $|M^{t_i}|$, the data curator deletes the record t_i that $|M^{t_i}| > \tau$ [7, 22] or rewrites its measure $|M^{t_i}| = \tau$ [1, 25, 38], where τ is a carefully chosen threshold. However, only bounding users' maximum contribution introduces systematic bias to the data table, leading to the poor utility of the query results. Although Wilson et al. try to bound the minimux contribution of users in a trivial way, their method is proved to be ineffective.

For answering sum queries under DP while alleviating the systematic bias, we put forward a Symmetrical Bounded Sum Query Processing Method (SBS). Rather than computing the upper threshold and lower bound with different algorithms, in SBS, the data curator carefully chooses a bounding range $[\Phi-\tau, \Phi+\tau]$, which is symmetrical about the bounding axis Φ, and rewrites $|M^{t_i}|$ such that any $|M^{t_i}|$ beyond the bounding range only contributes $\Phi-\tau$ or $\Phi+\tau$ to the query result. By carefully choosing the bounding axis and radius, SBS guarantees both the lower and upper threshold play an important role during the rewriting. In particular, by bounding the records whose measures' values larger than $\Phi+\tau$, the sensitivity of sum queries is bounded and less DP noise is added; by bounding the records whose measures' values smaller than $\Phi-\tau$, the systematic bias in query results is alleviated.

SBS takes a data table T which is an instance of relational schema $R(\mathbb{A}, M)$ as input and outputs a rewritten data table T'. The data curator first computes the optimal bounding axis Φ by traversing the records in the data table T, then determines the optimal bounding radius τ by formulating an optimization problem. Φ and τ form a bounding range $[\Phi-\tau, \Phi+\tau]$.

After determining the bounding range, the data curator rewrites the records in data table T such that the contribution of record t_i is $\max\min|M^{t_i}|, \Phi-\tau, \Phi+\tau$. We show the details of how to determine the optimal Φ and τ as follows.

4.2.1 Bounding Axis Φ. Given the original records $\{t_i | 1 \le i \le n\}$ and the bounded records $\{t_i' | 1 \le i \le n\}$, in order to reduce the systematic bias in the rewritten data table, we aim to find the optimal Φ that makes the mathematic expectation of the bounded records' measures converge to the mathematic expectation of the original records' measures. We show the optimal Φ in Eq. 3.

Theorem 4.1. *The optimal bounding axis is*

$$\Phi = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n}. \tag{3}$$

Due to space limitation, we put the proof of Theorem 4.1 in Appendix A.1.

4.2.2 Bounding Radius τ. To determine an optimal bounding radius τ, we consider two types of errors. Specifically, we denote the error caused by adding Laplace noise to the linear sum query results as the DP noise and denote the error caused by the deviation between the original values of the records and the bounded values of the records as the data distortion. A small τ indicates there are more values of records bounded, resulting in a large data distortion. While a large τ indicates the sum query has a large sensitivity, leading to a large DP noise. Therefore, we formalize an optimization problem. Before giving the details of the optimization problem, we first quantify the DP noise and the data distortion.

The magnitude of DP noise can be directly represented by the variance of the Laplace distribution:

$$2(\frac{\Delta f}{\epsilon})^2 = 2[min(\Phi + \tau, dom(M))/\epsilon]^2, \tag{4}$$

where $dom(M)$ is the domain of the measure M. As for the data distortion, we denote the data distortion as

$$\sum_{i=1}^{n} \frac{(|M^{t_i}| - |M^{t'_i}|)^2}{n}. \tag{5}$$

where t_i and t'_i refer to the original and bounded records, respectively.

In Eq. 5, the value of $|M^{t'_i}|$ depends on the relationship between the domain of measure $dom(M)$ and the bounding range $[\Phi-\tau, \Phi+\tau]$. For any original record t_i, the value of its measure $|M^{t_i}|$ belongs to either $[0, \Phi-\tau)$, $[\Phi-\tau, \Phi+\tau]$, or $(\Phi+\tau, dom(M)]$. After rewriting the data table, the value of the bounded record' measure $|M^{t'_i}|$ is $\Phi-\tau$, $|M^{t_i}|$, or $\Phi+\tau$. Therefore, we can convert Eq. 5 to

$$\sum_{i=1}^{\alpha} \frac{(\Phi - \tau - |M^{t_i}|)^2}{n} + \sum_{i=1}^{\beta} \frac{(\Phi + \tau - |M^{t_i}|)^2}{n}. \tag{6}$$

where α is the number of records that satisfy $|M^{t_i}| < \Phi - \tau$, and β is the number of records that satisfy $|M^{t_i}| > \Phi + \tau$.

Based on Eq. 4 and Eq. 6, the objective function is shown as follows:

$$\begin{aligned} &\arg\min \sum_{i=1}^{\alpha} \frac{(\Phi - \tau - |M^{t_i}|)^2}{n} + \sum_{i=1}^{\beta} \frac{(\Phi + \tau - |M^{t_i}|)^2}{n} \\ &+ 2[\frac{min(\Phi + \tau, dom(M))}{\epsilon}]^2. \\ &s.t. \qquad \tau > 0 \end{aligned} \tag{7}$$

The problem is a convex optimization problem. By taking the partial derivative of Eq. 7 with respect to τ, the optimal τ is shown in Theorem 4.2.

Theorem 4.2. *Suppose* $B = \sum_{j=1}^{\beta} |M^{t_j}| - \sum_{i=1}^{\alpha} |M^{t_i}|$. *The optimal bounding radius is*

$$\tau = \frac{\epsilon^2 B - [\epsilon^2(\beta - \alpha) + 2n]\Phi}{\epsilon^2(\alpha + \beta) + 2n}. \tag{8}$$

Due to space limitation, we put the proof of Theorem 4.2 in Appendix A.2.

4.2.3 Privacy Analysis. Although the bounding range determined by SBS can minimize the sum of DP noise and systematic bias, since both the bounding axis Φ and the radius τ is related to the sensitive data, taking the upper threshold $\Phi + \tau$ as the local sensitivity will inevitably cause privacy leakage! Fortunately, there are extensive works [11, 29] are proposed for calibrating the magnitude of DP noise while guaranteeing DP. In this paper, for taking the advantage of SBS while avoiding the possible leakage, we invoke the Propose-Test-Release mechanism (PTR), whose details are shown in Sec. 3.1, to guarantee the results of linear sum queries satisfy (ϵ, δ)-DP.

4.3 Hypothesis Testing based Prefix Sum Computing Method

After rewriting the data table via SBS, the data curator can acquire the noisy sum vector by answering the linear sum queries in the pre-query matrix via PTR. Based on the noisy sum vector, the data curator first computes the base cuboid and then constructs the remaining cuboids by aggregating the base cuboid in different dimensions. However, constructing the remaining cuboids based on the base cuboid directly will lead to noise aggregation, which refers to the problem that the DP noise in each cell of the base cuboid will be aggregated when constructing the remaining cuboids. As a baseline approach, the Noise Control Framework [6] alleviates noise aggregation by computing a subset of cuboids from the data table in a manner that reduces the overall noise and aggregates the remaining cuboids from the descendant cuboids in the subset. However, noise aggregation is still inevitable in this framework. Since the amount of aggregated DP noise is polynomial in the cardinality of $dom(M)$, a large $dom(M)$ will lead to a massive amount of DP noise aggregated in the constructed cuboids.

We consider alleviating noise aggregation by using the prefix-sum cube, which is a kind of structure where any ancestor cuboids can be constructed by the computation between pieces of prefix sums. However, the prefix sum cannot be applied to construct the base cuboid directly. As shown in Fig. 3(b), computing prefix sums by answering a workload of linear sum queries under DP will result in multiple DP noise. Therefore, to take the advantage of the prefix sum without introducing multiple DP noise, we propose a Hypothe**S**is Testing based Prefix Sum **Co**m**P**uting M**E**thod (SCOPE). Rather than directly computing the prefix sums under DP, SCOPE first computes the prefix sums based on the distribution of the noisy sum vector, which is predicted via hypothesis testing, and then boosts the accuracy of prefix sums by enforcing consistency.

We first describe how to compute prefix sums based on the noisy sum vector. The idea for computing prefix sums is based on an observation: for an attribute A in data table T, if its marginal follows a distribution $\mathcal{X}$ with probability density function f, then the corresponding prefix sums follows a distribution that is the integral of $\mathcal{X}$ with a probability distribution function F. Inspired by this observation, we can first predict the distribution $\mathcal{X}$ that the marginal of involved attributes follows by using hypothesis testing, and then compute the prefix sums based on the query results. Specifically, the hypothesis testing is a statistical method used to make inferences and draw conclusions about the distribution of a population based on a sample set drawn from them. For ease of illustration, we take a 1-D prefix-sum cuboid $\mathcal{P}$ of an attribute A as an example. Suppose the data curator acquires a noisy sum vector $\mathbf{x} = [x_1, x_2, ..., x_{dom(A)}]$, where $x_j = \sum_{|A^{t_i}|=j, 1\le i\le n} |M^{t_i}| (1 \le j \le dom(A))$, by answering a workload of linear sum queries in the pre-query matrix. Since the input of the hypothesis testing is a sample set, the data curator first converts the noisy sum vector $\mathbf{x}$ to a synthetic sample set $\mathbf{x'}$. The details of the conversion procedure are shown in Algorithm 2. In Algorithm 2, the data curator takes the noisy sum vector $\mathbf{x}$ as input and initials a synthetic sample set $\mathbf{x'}$. For any element x_i in $\mathbf{x}$, the data curator appends i into $\mathbf{x'}$ for x_i times. For $\mathcal{X}$ which represents the distribution of A, $\mathbf{x'}$

Algorithm 1: Converting

Input: The noisy sum vector $\mathbf{x}$.
Output: The synthetic sample set $\mathbf{x}'$.

```
Initialize a synthetic sample set x';
for i ∈ {1, 2, ..., dom(A)} do
    for j ∈ range(1, x_i) do
        Append i into x'
Return x'.
```

can be considered as a sample set drawn from the sample space $\Omega = \{1, 2, ..., dom(A)\}$ following a probability distribution P.

To compute the prefix-sum cuboid $\mathcal{P}$, the data curator predicts the probability distribution P of the random variable $\mathcal{X}$ through hypothesis testing. There are various hypothesis testing methods that can be used, such as Komogorov-Smirnov test [5], Anderson-Darling test [2], and Pearson's chi-squared test [30]. Without loss of generality, here we adopt the Komogorov-Smirnov test to predict the distribution P, which is one of the most useful and general hypothesis testing methods for quantifying the distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution. Suppose there exist a series of probability distributions in a hypothesis set $\mathbb{P} = P_1, P_2, ..., P_i, ...$ (e.g., normal distribution, Weibull distribution, exponential distribution, etc.). For any distribution P_i in $\mathbb{P}$, the data curator first computes the parameter of P_i based on $\mathbf{x}'$ and gets the corresponding cumulative distribution function $F^i_{exp}(x)$ of P_i. After getting $F^i_{exp}(x)$, the data curator then computes the empirical distribution function $F_{obv}(x)$:

$$F^i_{obv}(x) = \frac{1}{n}\sum_{j=1}^{n} 1_{(-\infty,x]}(X_j), \tag{9}$$

where X_j is the element in $\mathbf{x}'$ and $1_{(-\infty,x]}(X_j)$ is the indicator function:

$$1_{(-\infty,x]}(X_i) = \begin{cases} 1 & \text{if } X_j \le x \\ 0 & \text{if } X_j > x \end{cases}. \tag{10}$$

Given $F^i_{exp}(x)$ and $F^i_{obv}(x)$, the Kolmogorov–Smirnov statistic is $D_i = \max|F^i_{exp}(x) - F^i_{obv}(x)|$. and the p-value is $p_i = 2e^{-2(D_i)^2\frac{n}{2}}$. For all $p_i (1 \le i \le |\mathbb{P}|)$, the data curator chooses the largest p-value $p_{max} = max(p_1, p_2, ...)$ and identifies the corresponding probability distribution of random variable $\mathcal{X}$ as P_{max}.

We find that according to the definition, prefix-sum cuboid $\mathcal{P}$ can be regarded as the integral of the sum vector $\mathbf{x}$. If we denote the discretized probability density function and the discretized cumulative distribution function of P_{max} as $f(i)$ and $F(i)$ respectively, we have $\mathcal{P}(i) = \frac{\mathbf{x_i}}{f(i)}F(i)$. Based on the noisy sum vector $\mathbf{x}$ and the predicted distribution P_{max}, the prefix-sum cuboid $\mathcal{P}$ can be computed.

However, the process of computing the prefix sums will inevitably introduce errors: 1) the predicted probability distribution P_{max} may deviate from the actual distribution; 2) the sum vector $\mathbf{x}$ contains DP noise. These errors may make the noisy prefix sum $\mathcal{P}(i_2)$ smaller than $\mathcal{P}(i_1)$ $(0 \le i_i < i_2 \le dom|M|)$, which leads to inconsistency among cells and violates the prior knowledge that the values in the prefix-sum cube are in ascending order. To improve the utility, the data curator post-processes the noisy prefix-sum cube to remove the inconsistency. Specifically, for any cell $\tilde{P}(i_i)$ in the noisy prefix-sum cube $\tilde{P}$, the data curator makes sure $0 \le \mathcal{P}(i-1) \le \mathcal{P}(i) \le \mathcal{P}(i+1) \le n$.

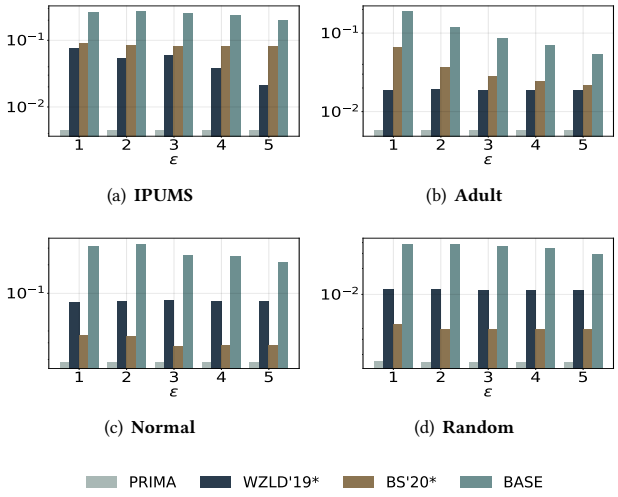

Figure 5: Varying ϵ on all datasets.

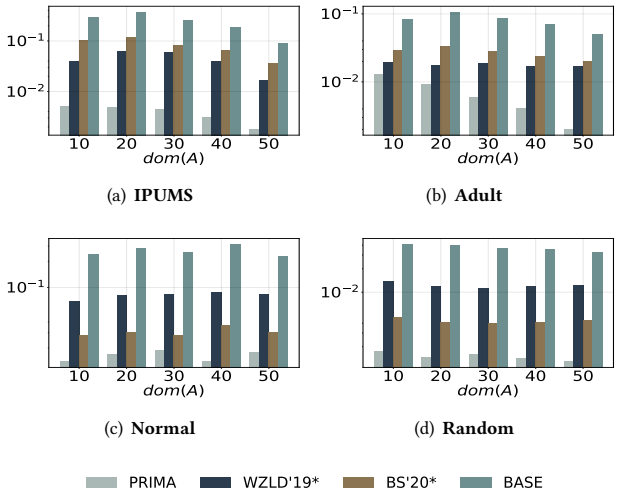

Figure 6: Varying $dom(A)$ on all datasets.

5 EXPERIMENTS

In this section, we first evaluate the performance of PRIMA and its building blocks by varying different parameters.

5.1 Experimental Settings

Datasets. We make use of two real datasets and two synthetic datasets in our experiments:

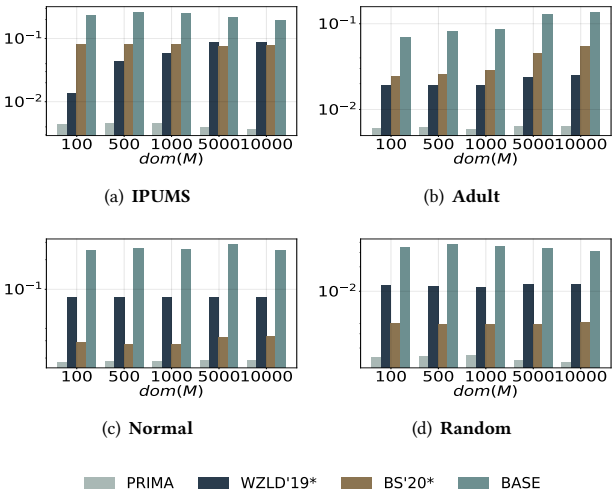

Figure 7: Varying $dom(M)$ on all datasets.

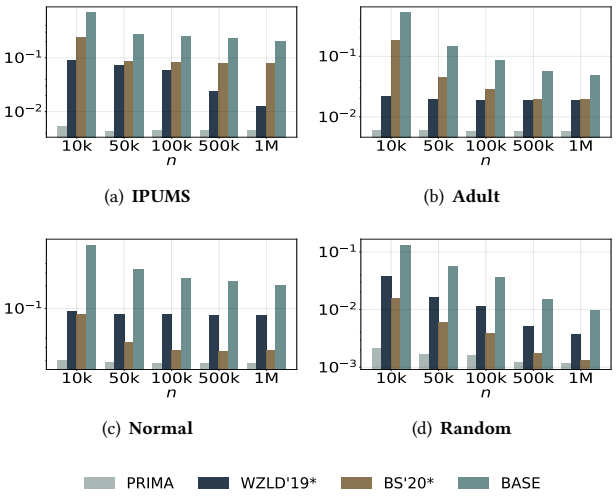

Figure 8: Varying n on all datasets.

- *IPUMS [35]:* A US census dataset from the IPUMS repository, which contains around 3 million records.
- *Adult [9]:* A dataset from the UCI machine learning repository. After removing missing values, the dataset contains around 50 thousands records.
- *Normal:* A dataset which is synthesized from the multivariate normal distribution with mean 0, standard deviation 1, contains 50 ordinal attributes. The covariance between every two attributes is 0.5.
- *Random:* A dataset which is synthesized from the uniform distribution, contains 50 ordinal attributes.

To experiment with different sizes of data table, we generate multiple test datasets from the four datasets with the number of records ranging from 10k to 1M. In addition, for evaluating different numbers of attributes and domain sizes, we generate multiple versions of these four datasets with the domain sizes of attributes ranging from 10 to 50.

Competitors. To evaluate the overall effectiveness of PRIMA, we compare PRIMA against the three baseline approaches:

- *BASE:* A straightforward approach which computes a base cuboid from the data table under DP without bounding users' contribution and constructs the other cuboids from the noisy base cuboid.
- *WZLD'20* [38]:* The method rewrites data table via Wilson et als' method and constructs data cubes using the Noise Control Framework[6].
- *BS'19* [4]:* The method employs a strategy that refines the work of Bun et al. to rewrite data tables, where the data curator limits the largest m values to the $m+1$th values rather than deletes them directly, and constructs data cubes using the Noise Control Framework[6].

Utility Metric. We use the Mean Absolute Error to measure the accuracy of constructed cuboids: $MAE = AVG\left(\left|\frac{\tilde{c}-c}{n}\right|\right)$, where n is the number of cells in a certain cuboids, $\tilde{c}$ and c are the noisy and true values stored in the cells, respectively.

Methodology. For each dataset and each approach, we construct data cube between any λ attributes and measure the MAE of each cell in the m-D cuboids. We repeated all experiments 10 times to obtain stable results. In all experiments, unless explicitly stated, we use the following default values for other relevant parameters: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4, \lambda = 4$.

Environment. We implement all the approaches in Python. All experiments are conducted on an Intel Core i5 2.50GHz PC with 8GB RAM.

5.2 Performance of PRIMA

In this part, we evaluate the performance of different methods under various privacy budget ϵ, the domains of attributes $dom(A)$, the domain of measure $dom(M)$ and the total number of records n.

Overall performance. Fig. 5- 8 shows the results of PRIMA and three baseline approaches when varying different parameters. We can find that in these approaches, PRIMA outperforms all other approaches. Although WZLD'19* and BS'20* perform comparably in most scenarios over IPUMS and Adult, when it comes to synthetic datasets, the performance of BS'20* has significantly improved.

Varying privacy budget ϵ. Fig. 5 shows the results varying ϵ from 1.0 to 5.0. It is obvious that PRIMA has a clear advantage over other approaches. We also find that, compared with other approaches, the MAEs of PRIMA are robust to the growth of ϵ. The reason can be explained as follows. Since PRIMA reduces the DP noise and the systematic bias via SBS, these two types of errors have only a minor impact on PRIMA. As a result, the effectiveness of PRIMA is not sensitive to the variations in the privacy budget ϵ.

Varying the domains of attributes $dom(A)$. Fig. 6 presents the results varying $dom(A)$ from 10 to 50 on all datasets. We observe that PRIMA consistently outperforms all other approaches. We can find that as $dom(A)$ increases, the MAEs of PRIMA are stable while the MAEs of the baseline approaches increase. The reason can be explained as follows. A larger $dom(A)$ indicates there are more cells in each cuboid. For the baseline approaches, as the number of cells in each cuboid increases, more noise aggregation

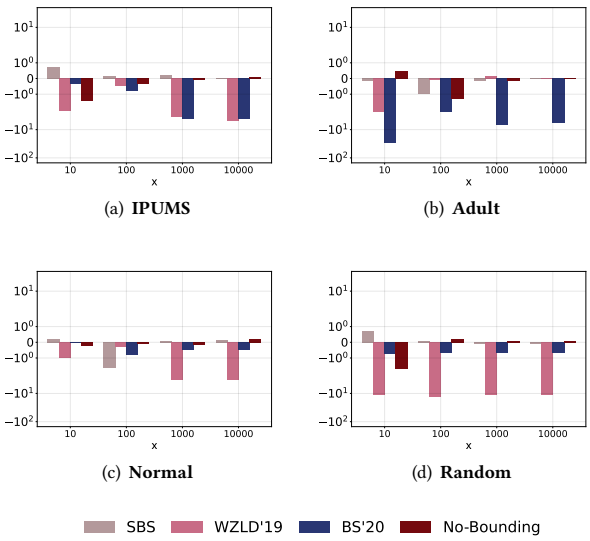

Figure 9: MBs of x queries on all datasets.

will take place, resulting in higher MAEs. While PRIMA alleviates noise aggregation by constructing prefix-sum cubes. As a result, the effectiveness of PRIMA is not sensitive to the increase of $dom(A)$.

Varying the domain of attributes $dom(M)$. Fig. 7 studies the impact of $dom(M)$. It is obvious that PRIMA performs the best among all approaches. Moreover, we observe that as $dom(M)$ increases, the MAEs of PRIMA are stable while the MAEs of the baseline approaches increase. The reason can be explained as follows. A large $dom(M)$ leads to a higher sensitivity of the sum query. In BASE, a high sensitivity makes the magnitude of DP noise grow. In WZLD'19*, massive systematic bias is introduced since only maximum contribution is bounded. While PRIMA can guarantee the utility by bounding the records from double sides.

Varying the number of records n. Fig. 8 shows the results varying n from 10K to 1M on all datasets. We can observe that the MAEs of all approaches decrease with the increase in the number of records, and PRIMA always performs the best.

5.3 Performance of Key Building Blocks

We first evaluate the advantage of SBS by comparing SBS with WZLD'20, BS'20 and No-Bounding, which is a straightforward method where the data curator answers sum queries under DP without rewriting the data table. For each dataset and each method, we randomly choose x sum queries and measure the Mean Bias $MB = \frac{1}{x}\sum_{i=1}^{x} \tilde{c} - c$. In all experiments, we use the following default values: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4, \lambda = 4$.

The MBs of four methods are shown in Fig. 9. For the four methods, the MBs of SBS and No-Bounding are close to zero as the number of queries x increases. While the MBs of SVT and R2T are negative in all cases. It indicates that SVTC and R2TC will result in a greater systematic bias.

We evaluate the advantages of SCOPE by comparing SCOPE with NCF, which refers to the Noise Control Framework, and CDB, which refers to the method for constructing a data cube in BASE. In this experiment, to construct the m-D cuboids via each method, we first obtain the required noisy sum vectors by answering linear sum queries in the pre-query matrix based on the original data table. We evaluate the effectiveness of these methods by measuring the MAE of m-D cuboids. In all experiments, we use the following default values: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4$.

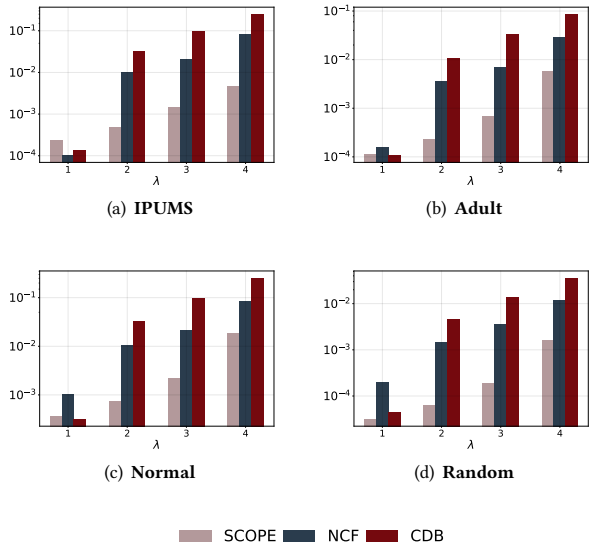

Figure 10: MAEs of λ-D cuboids on all datasets.

According to the results shown in Fig. 10, we find SCOPE outperforms the other methods in all cases except for $m = 1$. The reason is that: since SCOPE alleviates the problem of noise aggregation by constructing prefix-sum cubes, the ancestor prefix-sum cuboids can be constructed without excessive DP noise aggregation. As m increases, the DP noise in the constructed base cuboid generally aggregates in the ancestor cuboids that are constructed via NCF and CDB, while SCOPE can maintain lower MAEs. We also observe that when $m = 1$, the MAEs of CDB are smaller than those of SCOPE and NCF. This is because, in SCOPE, the base prefix-sum cuboid contains prediction error; in NCF, there is more than one cuboid in the initial set $\mathcal{L}_{pre}$ required to be computed, leading to a higher sensitivity when answering linear sum queries.

6 CONCLUSION

In this paper, we propose PRIMA for processing sum queries over data cubes under DP. To answer sum queries accurately while alleviating systematic bias, we put forward SBS, which rewrites the data table in a symmetric manner. Furthermore, to improve the utility of noisy prefix-sum cube and alleviate noisy aggregation, we design SCOPE, where the distribution of noisy sum vector is predicted via hypothesis testing and the prefix-sum cube is constructed based on the predicted distribution. Sufficient experiments are conducted to prove the effectiveness of PRIMA.

REFERENCES

[1] Kareem Amin, Alex Kulesza, Andres Muñoz Medina, and Sergei Vassilvitskii. 2019. Bounding User Contributions: A Bias-Variance Trade-off in Differential Privacy. In *ICML 2019*, Vol. 97. PMLR, 263–271.

[2] T. W. Anderson and D. A. Darling. 1952. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics* 23, 2 (1952), 193–212.

[3] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. 2017. Prochlo: Strong Privacy for Analytics in the Crowd. In *Proceedings of Operating Systems Principles*. ACM, 441–459.

[4] Mark Bun and Thomas Steinke. 2019. Average-Case Averages: Private Algorithms for Smooth Sensitivity and Mean Estimation. In *NeurIPS 2019*. 181–191.

[5] W.W. Daniel. 1990. *Applied Nonparametric Statistics*. PWS-KENT Pub.

[6] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. 2011. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of SIGMOD*. ACM, 217–228.

[7] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2022. R2T: Instance-optimal Truncation for Differentially Private Query Evaluation with Foreign Keys. In *Proceedings of SIGMOD*. ACM, 759–772.

[8] Wei Dong and Ke Yi. 2023. Universal Private Estimators. In *PODS 2023*. ACM, 195–206.

[9] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[10] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of Theory of Computing*. ACM, 371–380.

[11] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*.

[12] Alexander Edmonds, Aleksandar Nikolov, and Jonathan R. Ullman. 2020. The power of factorization mechanisms in local and central differential privacy. In *Proccedings of Theory of Computing*, Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.). ACM, 425–438.

[13] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Computer and Communications Security*. ACM, 1054–1067.

[14] Chang Ge, Xi He, Ihab F. Ilyas, and Ashwin Machanavajjhala. 2019. APEx: Accuracy-Aware Differentially Private Data Exploration. In *Proceedings of SIGMOD*. ACM, 177–194.

[15] Samuel Haney, Ashwin Machanavajjhala, John M. Abowd, Matthew Graham, Mark Kutzbach, and Lars Vilhuber. 2017. Utility Cost of Formal Privacy for Releasing National Employer-Employee Statistics. In *Proceedings of on Management of Data*. ACM, 1339–1354.

[16] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A Simple and Practical Algorithm for Differentially Private Data Release. In *Proceedings of NIPS*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 2348–2356.

[17] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proc. VLDB Endow.* 3, 1 (2010), 1021–1032.

[18] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proceedings of the VLDB Endowment* 3, 1 (2010), 1021–1032.

[19] Ching-Tien Ho, Rakesh Agrawal, Nimrod Megiddo, and Ramakrishnan Srikant. 1997. Range Queries in OLAP Data Cubes. In *Proceedings of SIGMOD*. ACM, 73–88.

[20] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. In *NeurIPS 2021*. 25993–26004.

[21] Noah M. Johnson, Joseph P. Near, and Dawn Song. 2018. Towards Practical Differential Privacy for SQL Queries. *Proc. VLDB Endow.* 11, 5 (2018), 526–539.

[22] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2019. PrivateSQL: A Differentially Private SQL Query Engine. *Proc. VLDB Endow.* 12, 11 (2019), 1371–1384.

[23] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. 2010. Optimizing linear counting queries under differential privacy. In *Proceedings of PODS*, Jan Paredaens and Dirk Van Gucht (Eds.). ACM, 123–134.

[24] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. 2015. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.* 24, 6 (2015), 757–781.

[25] Yuhan Liu, Ananda Theertha Suresh, Wennan Zhu, Peter Kairouz, and Marco Gruteser. 2023. Algorithms for bounding contribution for histogram estimation under user-level privacy.

[26] Ashwin Machanavajjhala, Daniel Kifer, John M. Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy: Theory meets Practice on the Map. In *Proceedings of International Conference on Data Engineering*. IEEE Computer Society, 277–286.

[27] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2018. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment* 11, 10 (2018), 1206–1219.

[28] Frank McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of SIGMOD*. ACM, 19–30.

[29] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of Theory of Computing*. ACM, 75–84.

[30] Karl Pearson. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.

[31] Davide Proserpio, Sharon Goldberg, and Frank McSherry. 2014. Calibrating Data to Sensitivity in Private Data Analysis. *Proc. VLDB Endow.* 7, 8 (2014), 637–648.

[32] Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding Hierarchical Methods for Differentially Private Histograms. *Proc. VLDB Endow.* 6, 14 (2013), 1954–1965.

[33] Yuan Qiu, Wei Dong, Ke Yi, Bin Wu, and Feifei Li. 2022. Releasing Private Data for Numerical Queries. In *Proceedings of KDD*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 1410–1419.

[34] Vibhor Rastogi and Suman Nath. 2010. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of SIGMOD*, Ahmed K. Elmagarmid and Divyakant Agrawal (Eds.). ACM, 735–746.

[35] Sobek, Matthew, Ruggles, and Steven. 1999. The IPUMS Project. *Historical Methods* 32, 3 (1999), 102–102.

[36] Apple Differential Privacy Team. 2017. Local privacy and statistical minimax rates.

[37] Yufei Wang and Xiang Cheng. 2022. PRISM: Prefix-Sum based Range Queries Processing Method under Local Differential Privacy. In *Proceedings of ICDE*. IEEE, 433–445.

[38] Royce J. Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2020. Differentially Private SQL with Bounded User Contribution. *Proc. Priv. Enhancing Technol.* 2020, 2 (2020), 230–250.

[39] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2010. Differential privacy via wavelet transforms. In *Proceedings of ICDE*. IEEE, 225–236.

[40] Ganzhao Yuan, Yin Yang, Zhenjie Zhang, and Zhifeng Hao. 2016. Convex Optimization for Linear Query Processing under Approximate Differential Privacy. In *Proceedings of KDD*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 2005–2014.

[41] Ganzhao Yuan, Zhenjie Zhang, Marianne Winslett, Xiaokui Xiao, Yin Yang, and Zhifeng Hao. 2015. Optimizing Batch Linear Queries under Exact and Approximate Differential Privacy. *ACM Trans. Database Syst.* 40, 2 (2015), 11:1–11:47.

[42] Dan Zhang, Ryan McKenna, Ios Kotsogiannis, George Bissias, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. 2020. eKTELO: A Framework for Defining Differentially Private Computations. *ACM Trans. Database Syst.* 45, 1 (2020), 2:1–2:44.

A SUPPLEMENTARY ANALYSIS

A.1 Proof of Theorem 4.1

Proof. The systematic bias E_b between the original records $\{t_i|1 \le i \le n\}$ and the clipped records $\{t_i'|1 \le i \le n\}$ is

$$E_b = \frac{\sum_{i=1}^{n} |M^{t_i}| - \sum_{i=1}^{n} |M^{t_i'}|}{n}. \tag{11}$$

Suppose there are α records that satisfy $|M^{t_i}| < \Phi - \tau (1 \le i \le \alpha)$, β records that satisfy $|M^{t_j}| > \Phi + \tau (1 \le j \le \beta)$, γ records that satisfy $\Phi - \tau \le |M^{t_k}| \le \Phi + \tau (1 \le k \le \gamma)$, Equation (11) can be converted to

$$\begin{aligned} E_b &= \frac{\sum_{i=1}^{\alpha} |M^{t_i}| - (\Phi - \tau)}{n} + \frac{\sum_{k=1}^{\gamma} |M^{t_k}| - |M^{t_k}|}{n} \\ &\quad + \frac{\sum_{j=1}^{\beta} |M^{t_j}| - (\Phi + \tau)}{n} \\ &= \frac{\sum_{i=1}^{n} |M^{t_i}|}{n} - \frac{n-\gamma}{n}\Phi + \frac{\alpha - \beta}{n}\tau - \frac{\sum_{i=1}^{\gamma} |M^{t_i}|}{n}. \end{aligned} \tag{12}$$

As shown in Equation (12), the systematic bias is affected by both Φ and τ. For determining the optimal Φ, we ignore the effect of τ by considering the situation where τ approaches 0. Then, we have

$$\lim_{\tau \to +0} E_b = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n} - \Phi. \tag{13}$$

To minimize the systematic bias, we can claim the optimal clipping axis is

$$\Phi = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n}. \tag{14}$$

A.2 Proof of Theorem 4.2

Proof. We use $Err(\tau)$ to represent the objective function in Equation (7). We can convert $Err(\tau)$ to

$$\begin{aligned} Err(\tau) &= \sum_{i=1}^{\alpha} \frac{[(\Phi - \tau) - |M^{t_i}|]^2}{n} + \sum_{j=1}^{\beta} \frac{[(\Phi + \tau) - |M^{t_j}|]^2}{n} \\ &\quad + 2[\frac{\Phi + \tau}{\epsilon}]^2 \\ &= \frac{\alpha + \beta}{n}\tau^2 + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n}\tau \\ &\quad + \frac{\alpha\Phi^2 + \beta\Phi^2 - 2\Phi(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &\quad + 2[\frac{\Phi + \tau}{\epsilon}]^2. \end{aligned} \tag{15}$$

The problem is a convex optimization problem. By taking the partial derivative of Equation (15) with respect to τ, the derivative of $Err(\tau)$ is

$$\begin{aligned} Err(\tau)' &= 2\frac{\alpha + \beta}{n}\tau + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &\quad + 4[\frac{\Phi + \tau}{\epsilon}]. \end{aligned} \tag{16}$$

Let $Err(\tau)' = 0$, we have

$$\begin{aligned} &Err(\tau)' = 0 \Rightarrow \\ &2\frac{\alpha + \beta}{n}\tau + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &+ 4[\frac{\Phi + \tau}{\epsilon}] = 0 \Rightarrow \\ &\tau = \frac{\epsilon^2 B - [\epsilon^2(\beta - \alpha) + 2n]\Phi}{\epsilon^2(\alpha + \beta) + 2n} \end{aligned} \tag{17}$$