

Tracking DP budget While Handling Basic SQL Queries

Joshua Allen^{*} Janardhan Kulkarni[†] Abhradeep Thakurta[‡] Sergey Yekhanin[§]

Abstract

The following write up presents a simple mechanism (closely based on existing literature) to track DP budget while handling basic SQL queries, namely, `COUNT`, `SUM`, `MEAN`, and `VAR`, while providing record level privacy.

1 Problem Statement

Given a data set $D = \{d_1, \dots, d_N\}$ from some domain \mathcal{D} , design differentially private algorithms to allow the following types of SQL queries:

1. `SELECT COUNT(*) FROM D WHERE $\langle predicate \rangle$`
2. `SELECT SUM(column) FROM D WHERE $\langle predicate \rangle$`
3. `SELECT MEAN(column) FROM D WHERE $\langle predicate \rangle$`
4. `SELECT VAR(column) FROM D WHERE $\langle predicate \rangle$`

Furthermore, we want to effectively keep track of the privacy budget if multiple of these queries are made to the data set D .

2 Privacy guarantees

We assume that individual records in the database correspond to different users. Our goal is to protect user level privacy, i.e., ensure that an adversary that interacts with the database through our mechanism may learn very little (in a strict formal sense) about any specific record, no matter how much auxiliary information he can access. We measure users' privacy loss in terms of (ϵ, δ) -differential privacy [1, Definiton 2.4].

The values of ϵ and δ need to be fixed in advance. They should be thought of as our privacy budget. One possible setting for δ is

$$\delta = \frac{1}{N\sqrt{N}}, \tag{1}$$

^{*}Microsoft

[†]Microsoft

[‡]Work done while at University of Santa Cruz, consulting for Microsoft.

[§]Microsoft

where N is the total number of rows in the database. In existing deployments in industry, e.g., [3, 2], the value of ϵ is usually set between 1 and 4. Smaller values correspond to higher privacy guarantees but yield lower accuracy or smaller number of queries that can be asked against the database. Depending on the application it may be deemed acceptable to refresh the privacy budget after some period of time, e.g., one month. The high level view of our protocol is detailed in Section 5.2.

3 Notation

In what follows

- \log denotes the natural logarithm;
- $\mathcal{N}(0, \sigma^2)$ represents a sample from a normal distribution with mean zero, and variance σ^2 .
- We use t to denote our privacy budget. We discuss this parameter in Section 5.
- When we talk about a specific `column` in a database, let M denote the maximum absolute value that may appear in it.

4 Handling a single query

Below we present algorithms for handling individual queries. The basic algorithms are quite simple, and proceed by computing the exact answer based on raw data, and then adding Gaussian noise. The magnitude of that noise depends on the range of values in the corresponding database `column`, query budget t , and a key parameter σ that is discussed in detail in Section 5.

1. **COUNT** query: Let X be the correct integer value representing the response to the query. Our algorithm outputs $X + \mathcal{N}(0, \sigma_1^2)$, where

$$\sigma_1 = \sqrt{t} \cdot \sigma. \quad (2)$$

We note that issuing multiple **COUNT** queries that correspond disjoint predicates (e.g., as done when computing a histogram) does not lead to additional privacy losses, and can be accounted for as a single query.

2. **SUM** query: For the `column` on which the sum is calculated on, let M be the maximum possible absolute value that may appear in it. Let X be the real number which is the correct response to the query. Our algorithm outputs $X + \mathcal{N}(0, \sigma_2^2)$, where

$$\sigma_2 = \sqrt{t} \cdot |M| \cdot \sigma. \quad (3)$$

3. **MEAN** query: For the `column` on which the sum is calculated on, let M be the maximum possible absolute value that may appear in it. In order to compute the **MEAN** one needs to issue two basic queries, i.e., **SUM** and **COUNT** and then divide the output of **MEAN** by the output of **COUNT**.

Given that outputs of **COUNT** and **SUM** are noisy, and the fact that we are dividing by the output of **COUNT**; we only get a reliable estimate for true value of the **MEAN**, when the **COUNT**

is sufficiently large. In particular, let \hat{n} denote the output of the **COUNT** query, and n be the exact value. Similarly, let $\hat{\Sigma}$ denote the output of the **SUM** query, and Σ be the exact value. We guarantee that with probability $(1 - \alpha)$, the following logical implication is valid:

$$\text{If } \left(\hat{n} > 2\sqrt{2\ln(4/\alpha)} \cdot \sigma_1 \right); \text{ then} \quad (4)$$

$$\left| \frac{\hat{\Sigma}}{\hat{n}} - \frac{\Sigma}{n} \right| < \frac{\sqrt{2\ln(4/\alpha)} \cdot \sigma_2}{\hat{n}} + \frac{2\sqrt{2\ln(4/\alpha)} \cdot |\hat{\Sigma}| \cdot \sigma_1 + 4\ln(4/\alpha) \cdot \sigma_1 \cdot \sigma_2}{\hat{n}^2}. \quad (5)$$

The detailed analysis of the our accuracy guarantees for **MEAN** is given in Section 6. One important thing to note is that our algorithm for the **MEAN** does not produce an unbiased estimator. Also, the error bound (5) is just an upper bound; and the error is typically smaller.

4. **VAR** query: For a set of n real numbers $Z = \{z_1, \dots, z_n\}$,

$$\text{Var}(Z) = \frac{1}{n} \sum_{i=1}^n (z_i - \mu)^2 = \frac{1}{n} \sum_{i=1}^n z_i^2 - \left(\frac{1}{n} \sum_{i=1}^n z_i \right)^2.$$

Our algorithm for the **VAR** query is based on our algorithm for the **MEAN**. Given the predicate, we issue the **COUNT** query, the **SUM** query, and the **SUM** query for squares of the values $\{z_i\}$. Let n, Σ, Σ_2 be the true answers to the above queries, and $\hat{n}, \hat{\Sigma}, \hat{\Sigma}_2$ be the noisy answers. To compute $\hat{\Sigma}_2$, we set $\hat{\Sigma}_2 = \Sigma_2 + \mathcal{N}(0, \sigma_3^2)$, where

$$\sigma_3 = \sqrt{t} \cdot M^2 \cdot \sigma. \quad (6)$$

Our estimate for **VAR** is given by

$$\text{VAR}'(Z) = \frac{\hat{\Sigma}_2}{\hat{n}} - \left(\frac{\hat{\Sigma}}{\hat{n}} \right)^2. \quad (7)$$

Our accuracy guarantee is similar to the one that we have for the **MEAN**. Specifically, we ensure that with probability $1 - 3\alpha/2$ the following logical implication is valid:

$$\text{If } \left(\hat{n} > 2\sqrt{2\ln(4/\alpha)} \cdot \sigma_1 \right); \text{ then} \quad (8)$$

$$|\text{VAR}(Z) - \text{VAR}'(Z)| < f(\hat{n}, \sigma_1, \hat{\Sigma}_2, \sigma_3) + f(\hat{n}, \sigma_1, \hat{\Sigma}, \sigma_2) \cdot \left(f(\hat{n}, \sigma_1, \hat{\Sigma}, \sigma_2) + \frac{2 \cdot \hat{\Sigma}}{\hat{n}} \right), \quad (9)$$

where $f(\hat{n}, \sigma_1, \hat{\Sigma}, \sigma_2)$ denotes the expression in the right hands side of (5). The detailed analysis of the our accuracy guarantees for **VAR** is given in Section 7. Again we remark that our algorithm for the **VAR** does not produce an unbiased estimator. Also, the error bound (5) is just an upper bound; and the error is typically smaller.

5 Handling multiple queries

Our approach is as follows. On the onset, we choose the query budget, which is an integer q that is initially set to some value t . When we respond to a **COUNT** or **SUM** query, or when we compute a histogram; we reduce our query budget q by 1. When we answer a **MEAN** query, we reduce q by 2. When we answer a **VAR** query, we reduce q by 3. We spread our privacy budget uniformly, expecting that we will exhaust the query budget. When the query budget is exhausted the database has to be taken offline, as no more queries can be answered.

5.1 Technical details

Here are the technical details of our privacy accounting with references to the literature.

- When we discuss differential privacy neighboring relation between databases is with respect to a single user leaving or joining the database.
- Note that all basic privacy mechanisms described in Section 4 (i.e., the **COUNT**, **SUM**, and **SUM** of squares of values) are instances of the standard Gaussian mechanism [1, Appendix A]. By simple scaling (respectively by 1, M^{-1} and M^{-2} all these mechanisms can be made to have sensitivity 1. Therefore by [4, Corollary 3], each of the mechanisms satisfies $(\alpha, \frac{\alpha}{2 \cdot t \cdot \sigma^2})$ -Renyi Differential Privacy (RDP), for all $\alpha > 1$.
- By [4, Proposition 1], by the time we exhaust our query budget of t , our privacy loss satisfies $(\alpha, \frac{\alpha}{2 \cdot \sigma^2})$ -RDP.
- Finally, by [4, Proposition 3], our privacy loss satisfies

$$\left(\frac{\alpha}{2 \cdot \sigma^2} + \frac{\log 1/\delta}{\alpha - 1}, \delta \right) - \text{DP}. \quad (10)$$

- We set $\alpha = \sigma \cdot \sqrt{2 \cdot \log 1/\delta} + 1$. Clearly, $\alpha > 1$. This yields

$$\left(\frac{\sigma \cdot \sqrt{2 \cdot \log 1/\delta} + 1}{2 \cdot \sigma^2} + \frac{\log 1/\delta}{\sigma \cdot \sqrt{2 \cdot \log 1/\delta}}, \delta \right) - \text{DP} \quad (11)$$

- The expression above yields (ϵ, δ) -DP when

$$\sigma = \frac{\sqrt{\log 1/\delta} + \sqrt{\log 1/\delta + \epsilon}}{\sqrt{2} \cdot \epsilon}. \quad (12)$$

5.2 The protocol

In what follows N denotes the number of records (rows) in the database.

- Set δ according to (1).
- Set ϵ between 1 and 4.
- Set σ according to (12).

- Initialize the query budget q with an initial value t .
- Execute queries against the database using the single query mechanisms from Section 4. Decrement q when computing **COUNT** or **SUM**, or when computing a histogram. Reduce q by two, when computing **MEAN** (as with our algorithm a single **MEAN** query amounts to two basic queries). Similarly, reduce q by three, when computing **VAR**, as a single **VAR** query amounts to three basic queries.
- Terminate the algorithm and take the database offline when the query budget runs out.

5.3 Example

Suppose we have a database with $n = 100,000$ records. We set δ as in (1) and we set $\epsilon = 3$. Now by (12) $\sigma \approx 2.04$. We set our query budget to $t = 2,000$. With such a query budget we could, for instance, answer 1,000 **COUNT** queries, and 500 **SUM** queries, and 250 **MEAN** queries. Imagine that we are dealing with numeric data that is in the range $[0, M]$. Our accuracy guarantees would be as follows:

- For **COUNT** queries, the standard deviation of our noise is: 92;
- For **SUM** queries, the standard deviation of our noise is: $92 \cdot M$;
- For **MEAN** and **VAR** queries estimates of our accuracy guarantees are given by formulae (5) and (9).

5.4 Remark

Imagine that we are in a situation where we do not know (even approximately) the number of queries that will be asked against the database. In such scenarios we may choose to answer queries that arrive earlier with higher accuracy, consuming more of the privacy budget. With this approach we may be able to handle an unbounded number of queries, although queries that arrive late will have very noisy answers. We do not work out the corresponding math in the current version of this manuscript.

6 Accuracy of the MEAN estimation

In this section we present a detailed analysis of the our accuracy guarantees for the **MEAN** query. As in Section 4, let \hat{n} denote the output of the **COUNT** query, and n be the exact value. Similarly, let $\hat{\Sigma}$ denote the output of the **SUM** query, and Σ be the exact value. We have $\hat{n} = n + \xi_1$, where ξ_1 is drawn according to the normal distribution $\mathcal{N}(0, \sigma_1^2)$. Similarly, $\hat{\Sigma} = \Sigma + \xi_2$, where ξ_2 is drawn according to the normal distribution $\mathcal{N}(0, \sigma_2^2)$. By the standard tail bound for the normal distribution, for ξ that is drawn from $\mathcal{N}(0, \sigma'^2)$,

$$\Pr[|\xi| > x \cdot \sigma'] \leq 2 \cdot e^{-x^2/2}. \quad (13)$$

Thus with probability $1 - \alpha$, we have both

$$|\xi_1| \leq \sqrt{2 \ln(4/\alpha)} \cdot \sigma_1 \quad \text{and} \quad |\xi_2| \leq \sqrt{2 \ln(4/\alpha)} \cdot \sigma_2. \quad (14)$$

We now need to show that in case inequalities (14) and (4) are satisfied; then the inequality (5) is well defined and valid. Our assumptions clearly imply,

$$\frac{|\xi_1|}{|\hat{n}|} \leq \frac{1}{2}. \quad (15)$$

Our goal is to bound

$$\begin{aligned} \left| \frac{\hat{\Sigma} - \xi_2}{\hat{n} - \xi_1} - \frac{\hat{\Sigma}}{\hat{n}} \right| &= \left| \frac{\hat{\Sigma} - \xi_2}{\hat{n}} \cdot \left(1 + \frac{\xi_1/\hat{n}}{1 - \xi_1/\hat{n}} \right) - \frac{\hat{\Sigma}}{\hat{n}} \right| \\ &= \left| \frac{\hat{\Sigma} \cdot \xi_1/\hat{n}}{\hat{n} \cdot (1 - \xi_1/\hat{n})} - \frac{\xi_2}{\hat{n}} - \frac{\xi_2 \cdot \xi_1/\hat{n}}{\hat{n} \cdot (1 - \xi_1/\hat{n})} \right| \\ &\leq \frac{\sqrt{2 \ln(4/\alpha)} \cdot \sigma_2}{\hat{n}} + \frac{2\sqrt{2 \ln(4/\alpha)} \cdot |\hat{\Sigma}| \cdot \sigma_1 + 4 \ln(4/\alpha) \cdot \sigma_1 \cdot \sigma_2}{\hat{n}^2}, \end{aligned}$$

where the latter inequality relies on (14) and (15).

7 Accuracy of the VAR estimation

In this section we present a detailed analysis of the our accuracy guarantees for the **VAR** query. Recall that n, Σ, Σ_2 denote the true values of **COUNT**, **SUM**, and **SUM** of squares. Similarly, $\hat{n}, \hat{\Sigma}, \hat{\Sigma}_2$ denote the noisy answers to the respective queries. We have

$$\hat{n} = n + \xi_1 \quad \hat{\Sigma} = \Sigma + \xi_2 \quad \hat{\Sigma}_2 = \Sigma_2 + \xi_3,$$

where ξ_1 is drawn $\mathcal{N}(0, \sigma_1^2)$, ξ_2 is drawn $\mathcal{N}(0, \sigma_2^2)$, ξ_3 is drawn $\mathcal{N}(0, \sigma_3^2)$. By (13), with probability $1 - 3\alpha/2$, we have

$$|\xi_1| \leq \sqrt{2 \ln(4/\alpha)} \cdot \sigma_1 \quad \text{and} \quad |\xi_2| \leq \sqrt{2 \ln(4/\alpha)} \cdot \sigma_2 \quad \text{and} \quad |\xi_3| \leq \sqrt{2 \ln(4/\alpha)} \cdot \sigma_3. \quad (16)$$

We now need to show that in case inequalities (16) and (8) are satisfied; then the inequality (9) is well defined and valid. Our assumptions clearly imply,

$$\frac{|\xi_1|}{|\hat{n}|} \leq \frac{1}{2}. \quad (17)$$

Our goal is to bound

$$\begin{aligned} |\text{VAR}(Z) - \text{VAR}'(Z)| &= \left| \frac{\hat{\Sigma}_2 - \xi_3}{\hat{n} - \xi_1} - \left(\frac{\hat{\Sigma} - \xi_2}{\hat{n} - \xi_1} \right)^2 - \frac{\hat{\Sigma}_2}{\hat{n}} + \left(\frac{\hat{\Sigma}}{\hat{n}} \right)^2 \right| \\ &\leq f(\hat{n}, \sigma_1, \hat{\Sigma}_2, \sigma_3) + \left| \left(\frac{\hat{\Sigma} - \xi_2}{\hat{n} - \xi_1} \right)^2 - \left(\frac{\hat{\Sigma}}{\hat{n}} \right)^2 \right| \\ &\leq f(\hat{n}, \sigma_1, \hat{\Sigma}_2, \sigma_3) + f(\hat{n}, \sigma_1, \hat{\Sigma}, \sigma_2) \cdot \left| \left(\frac{\hat{\Sigma} - \xi_2}{\hat{n} - \xi_1} \right) + \left(\frac{\hat{\Sigma}}{\hat{n}} \right) \right|. \end{aligned}$$

To obtain the bound (9), it suffices to note that

$$\begin{aligned}
\left| \left(\frac{\hat{\Sigma} - \xi_2}{\hat{n} - \xi_1} \right) + \left(\frac{\hat{\Sigma}}{\hat{n}} \right) \right| &= \left| \frac{\hat{\Sigma} - \xi_2}{\hat{n}} \cdot \left(1 + \frac{\xi_1/\hat{n}}{1 - \xi_1/\hat{n}} \right) + \frac{\hat{\Sigma}}{\hat{n}} \right| \\
&= \left| \frac{\hat{\Sigma} \cdot \xi_1/\hat{n}}{\hat{n} \cdot (1 - \xi_1/\hat{n})} - \frac{\xi_2}{\hat{n}} - \frac{\xi_2 \cdot \xi_1/\hat{n}}{\hat{n} \cdot (1 - \xi_1/\hat{n})} + \frac{2 \cdot \hat{\Sigma}}{\hat{n}} \right| \\
&\leq \frac{\sqrt{2 \ln(4/\alpha)} \cdot \sigma_2}{\hat{n}} + \frac{2\sqrt{2 \ln(4/\alpha)} \cdot |\hat{\Sigma}| \cdot \sigma_1 + 4 \ln(4/\alpha) \cdot \sigma_1 \cdot \sigma_2}{\hat{n}^2} + \frac{2 \cdot \hat{\Sigma}}{\hat{n}}.
\end{aligned}$$

References

- [1] C. Dwork, A. Roth, “The algorithmic foundations of differential privacy,” 2014.
- [2] B. Ding, J. Kulkarni, S. Yekhanin, “Collecting telemetry data privately,” NIPS 2017.
- [3] <https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appliedifferentialprivacysystem.pdf>
- [4] Ilya Mironov, “Renyi differential privacy,” In 30th IEEE Computer Security Foundations Symposium (CSF), pages 263–275, 2017.