DPSyn: Differentially Private Synthetic Data Publication

Ninghui Li, Anqi Chen, Zitao Li, Tianhao Wang
Team DPSyn, Purdue University, Peking University
{ninghui, zitaoli, tianhaowang}@purdue.edu, caq@pku.edu.cn

Overview

Since we serve for synthesizing a dataset in general case, the high-level idea of our method is to first obtain all two-way differentially private marginals (so that we have a estimate of the number of records by averaging the marginals' sum), and then synthesize a dataset by updating the final dataset iteratively.

Method for the Final Submission

We first describe the encoding scheme (similar to the numerical binning strategy provided in the code) that we extract from examining the data.

Extracting Information from the Dataset.
Starting from the original dataset, we first apply the binning for the numerical attributes. The default binning strategy is provided in the schema-genration github repository(https://github.com/hd23408/nist-schemagen) and users may fine-tune it by changing the input config file. We then perform the following encoding of the attributes. The given "parameters.json" file describes the full domain of these attributes; our goal here is to get rid of some values that never appear so that the noise is reduced.
The encoding procedure (`dataloader/Dataloader.py`) works as a pre-processing step to the private dataset. We encode all the categorical attributes to corresponding index codes like '0', '1', '2', etc. And we remove the identifier column from the original dataset. The main procedure (described later) will work on the encoded version. After the main procedure finishes, we post-process (`dataloader/RecordPostprocessor.py`) the differentially private dataset we get by decoding its values back to their original values.
After the previous steps, we have a dataset with all the categorical attributes. For the set of 1-way marginals, we denote it as $\mathcal{M}_{T1}$ for private dataset. For all 2-way marginals, we denote them as $\mathcal{M}_{T2}$ for private dataset.

Synthesize Data When Given a Private Dataset.
We always use θ, which is the value for "max_records_per_individual", as the sensitivity.
When given a private dataset, we firstly obtain all the 2-way marginal from the input dataset. We then decide the noise type by the function get_noise in utils/advanced_composition.py, deciding whether to use laplace noise or gauss noise.
We average the processed noisy 2-way marginals' sum to get an estimate of the records' num to generate, otherwise, users can manually input it in command lines.
Synthesizing methods are presented below:
Using all Two-way Marginals: Here because $\mathcal{M}_{T2}$ is already differentially private, using DPSyn can be thought of as a post-processing step and do not have privacy concerns. We first ensure that all two-way marginals are consistent (summing up to be the same) and non-negative and get $\mathcal{M}_{T2'}$.
We then use DPSyn [4, 5] to generate n records using $\mathcal{M}_T$ (`method/dpsyn.py`). DPSyn is the method our team developed for synthetic data generation in last competition.
We only spend privacy budget for obtaining the two-way marginals $\mathcal{M}_{T2}$; the remaining operations are part of the post-processing.

Mathematical Proof for DP satisfaction

We decide whether to use laplace or gauss noise by comparing the estimated varience.
We add noise by using (1)Laplace Mechanism[2] or (2)Gaussian Mechanism [2] together with zCDP [1] to obtain the noisy 2-way marginals from the input dataset ($\mathcal{M}_{T2}$).
After obtaining the noisy 2-way marginals, we can use them to construct synthetic dataset without consuming privacy budget, since this is a post processing procedure.
Referring to[5] , you can see mathematical proof as below.

Gaussian Mechanism

There are several approaches for designing mechanisms that satisfy (ϵ, δ)-differential privacy. In this paper, we use the Gaussian mechanism. The approach computes a function f on the dataset D in a differentially privately way, by adding to $f(D)$ a random noise. The magnitude of the noise depends on Δ_f, the *global sensitivity* or the ℓ_2 sensitivity of f. Such a mechanism $\mathbf{A}$ is given below:

$$\begin{aligned} \mathbf{A}(D) &= f(D) + \mathsf{N}\left(0, \Delta_f^2 \sigma^2 \mathbf{I}\right) \\ \text{where} \qquad \Delta_f &= \max_{(D,D'):D \simeq D'} ||f(D) - f(D')||_2. \end{aligned}$$

In the above, $\mathsf{N}(0, \Delta_f^2 \sigma^2 \mathbf{I})$ denotes a multi-dimensional random variable sampled from the normal distribution with mean 0 and standard deviation $\Delta_f \sigma$, and $\sigma = \sqrt{2 \ln \frac{1.25}{\delta}}/\epsilon$.

Composition via Zero Concentrated DP

For a sequential of k mechanisms $\mathcal{A}_1, \ldots, \mathcal{A}_k$ satisfying (ϵ_i, δ_i)-DP for $i = 1, \ldots, k$ respectively, the basic composition result [2] shows that the privacy composes linearly, i.e., the sequential composition satisfies $(\sum_i^k \epsilon_i, \sum_i^k \delta_i)$-DP. When $\epsilon_i = \epsilon$ and $\delta_i = \delta$, the advanced composition bound from [3] states that the composition satisfies $(\epsilon\sqrt{2k \log(1/\delta')} + k\epsilon(e^\epsilon - 1),\ k\delta + \delta')$-DP.
To enable more complex algorithms and data analysis task via the composition of multiple differentially private building blocks, zero Concentrated Differential Privacy (zCDP for short) offers elegant composition properties. The general idea is to connect (ϵ, δ)-DP to Rényi divergence, and use the useful property of Rényi divergence to achieve tighter composition property. In another word, for fixed privacy budget ϵ and δ, zCDP can provide smaller standard deviation for each task compared to other composition techniques.
Formally, zCDP is defined as follows:

Definition 1 (Zero-Concentrated Differential Privacy (zCDP) [1])**.** *A randomized mechanism $\mathcal{A}$ is ρ-zero concentrated differentially private (i.e., ρ-zCDP) if for any two neighboring databases D and D' and all $\alpha \in (1, \infty)$,*

$$\mathcal{D}_\alpha(\mathcal{A}(D)||\mathcal{A}(D')) \triangleq \frac{1}{\alpha - 1} \log\left(\mathbb{E}\left[e^{(\alpha-1)L^{(o)}}\right]\right) \leq \rho\alpha$$

Where $\mathcal{D}_\alpha(\mathcal{A}(D)||\mathcal{A}(D'))$ is called α-Rényi divergence between the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$. L^o is the privacy loss random variable with probability density function $f(x) = \log \frac{\mathsf{Pr}[\mathcal{A}(D)=x]}{\mathsf{Pr}[\mathcal{A}(D')=x]}$.

zCDP has a simple linear composition property [1]. we giive the introduction of zCDP's composition property to as below.

Proposition 1. *Two randomized mechanisms $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy ρ_1-zCDP and ρ_2-zCDP respectively, their sequential composition $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ satisfies $(\rho_1 + \rho_2)$-zCDP.*

The following two propositions restates the results from [1], which are useful for composing Gaussian mechanisms in differential privacy.

Proposition 2. *If $\mathbf{A}$ provides ρ-zCDP, then $\mathbf{A}$ is $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$-differentially private for any $\delta > 0$.*

Proposition 3. *The Gaussian mechanism which answers $f(D)$ with noise $\mathcal{N}(0, \Delta_f^2\sigma^2\mathbf{I})$ satisfies $(\frac{1}{2\sigma^2})$-zCDP.*

Given the privacy constraint ϵ and δ, we can calculate the amount of noise for each task using Propositions 1 to 3. In particular, we first use Proposition 2 to compute the total ρ allowed. Then we use Proposition 1 to allocate ρ_i for each task i. Finally, we use Proposition 3 to calculate σ for each task.

Theorem 4. *Given privacy budget ϵ, δ and the number of tasks m, the standard deviation for each task is* $\sigma = \left(\sqrt{2m\Delta_f^2 \log\frac{1}{\delta}} + \sqrt{2m\Delta_f^2 \log\frac{1}{\delta} + 2m\epsilon\Delta_f^2}\right)/2\epsilon$.

Proof. Proposition 2 states that ρ-zCDP is equivalent to $(\rho + 2\sqrt{\rho\log(1/\delta)}, \delta)$-DP; thus we have $\epsilon = \rho + 2\sqrt{\rho\log(1/\delta)}$. Rearranging the above equation, we have

$$\sqrt{\rho}^2 + 2\sqrt{\log\frac{1}{\delta}} \cdot \sqrt{\rho} - \epsilon = 0 \tag{1}$$

By solving Equation 1, we get the relationship between ρ and ϵ, δ:

$$\sqrt{\rho} = \sqrt{\log\frac{1}{\delta} + \epsilon} - \sqrt{\log\frac{1}{\delta}}$$

Assume the total privacy budget for zCDP is ρ, it is obvious that the privacy budget for each task is $\rho_0 = \frac{\rho}{m}$ based on Proposition 1. From Proposition 3, we have

$$\begin{aligned}
\sigma_0 &= \sqrt{\frac{\Delta_f^2}{2\rho_0^2}} = \sqrt{\frac{m\Delta_f^2}{2\rho}} \\
&= \frac{\sqrt{k\Delta_f^2}}{\sqrt{2}\left(\sqrt{\log\frac{1}{\delta} + \epsilon} - \sqrt{\log\frac{1}{\delta}}\right)} \\
&= \frac{\sqrt{2m\Delta_f^2 \log\frac{1}{\delta}} + \sqrt{2m\Delta_f^2 \log\frac{1}{\delta} + 2m\epsilon\Delta_f^2}}{2\epsilon}
\end{aligned}$$

□

Compared with (ϵ, δ)-DP, zCDP provides a tighter bound on the cumulative privacy loss under composition, making it more suitable for algorithms consist of a large number of tasks.

References

[1] M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[2] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[3] C. Dwork, G. Rothblum, and S. Vadhan. Boosting and differential privacy. *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 51 – 60, 2010.

[4] N. Li, Z. Zhang, and T. Wang. DPSyn: Differentially private synthetic data publication, 2018.

[5] Z. Zhang, T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, and Y. Zhang. Privsyn: Differentially private data synthesis, 2020.