DPSyn: Differentially Private Synthetic Data Publication

Ninghui Li, Zitao Li, Tianhao Wang
Team DPSyn, Purdue University
{ninghui, zitaoli, tianhaowang}@purdue.edu

Overview

Our approach relies on the fact that the provided Illinois-Ohio synthetic dataset is considered public and can be used without privacy cost. We call this the *public dataset* (stored in `dataloader/public.csv` in our submission). Since the scoring scheme measures each PUMA-year group separately, the high-level idea of our method is to first obtain a differentially private PUMA-year marginal (so that we have a good estimate of the number of records in each PUMA-year), and then for each PUMA-year, synthesize a small dataset, using the public dataset together with differentially private information from the private dataset. The final dataset is the union of these small datasets.

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. We then describe three methods dealing with low, middle, and high ϵ values. Finally we give the precise criteria of when to use which method.

Exacting Information from the Public Dataset. Starting from the public dataset, we first apply the binning for the numerical attributes. The binning strategy is provided in the given code. 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 both the public dataset and the private 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.

- Encode SEX and MARST into one attribute with 12 values.
- Encode the five insurance related attributes into one attribute. These are HCOVANY, HCOVPRIV, HINSEMP, HINSCAID, HINSCARE. They are all binary valued. Out of the 32 combinations, only 13 actually appear due to semantic relationships.
- Encode EMPSTAT, LABFORCE, EMPSTATD into one attribute. Since EMPSTAT and LABFORCE are uniquely determined by EMPSTATD, this effectively means that the total number of values is that for EMPSTATD.
- Encode the two attributes WORKEDYR, WRKLSTWK into one with 28 values that appear in the public dataset.
- Encode the two attributes ABSENT and LOOKING. While the domains have 5 and 4 values, only 3 values actually appeared for each attribute in the public dataset. So we encode these two attributes into one with 9 values.
- Encode the two attributes AVAILBLE and WRKRECAL. While the domains have 6 and 4 values, only 4 and 3 appeared in the public dataset. We encode these two attributes into one with 12 values.

After the previous steps, we have a dataset with 25 categorical attributes. For the set of 23 1-way marginals (other than PUMA and year), we denote it as $\mathcal{M}_{P1}$ for public dataset and $\mathcal{M}_{T1}$ for private dataset. For the 253 (or $\binom{23}{2}$ for all attributes other than PUMA and year) 2-way marginals, we denote them as $\mathcal{M}_{P2}$ for public dataset and $\mathcal{M}_{T2}$ for private dataset.

Synthesize Data When Given a Private Dataset. Our method (`method/sample_parallel.py`) does not obtain the 3- or 4-way marginals with PUMA and year. Instead, we only use the general distributions across all PUMA and year. *We always use θ, which is the value for "max_records_per_individual", as the sensitivity.*

When given a private dataset, we use the Laplace Mechanism [2] to obtain the PUMA-year 2-way marginal from the input dataset. We then process the noisy marginal as follows: we first change any cell value less than 300 to 300, and then round each cell to the nearest multiple of K. For each pair of PUMA-year values, let n_i be the processed noisy count for it (n_i is a multiple of K), we synthesize n_i records. Typically we set K to be 10 or 100. A larger K allows us to reuse more results because there will be less unique n_i's, reducing the running time.

Three synthesizing methods (for low, middle, and high ϵ ranges) are presented below:

- **Public Marginals Only.** When ϵ is low, we only obtain the PUMA-year 2-way marginal from the input data, and use the public marginal to synthesize the data. We use $K = 10$. To synthesize a set of n_i records, we randomly sample n_i records from the public dataset, and then iteratively update the set to make its 2-way distributions close to the 2-way marginals from the public dataset ($\mathcal{M}_{P2}$).

- **Using One-way Marginals.** When ϵ is in the middle range (the range and the ϵ allocation are described more precisely later), in addition to the PUMA-year 2-way marginal, we also use the Laplace Mechanism to get 23 1-way marginals ($\mathcal{M}_{T1}$) for the other 23 attributes. After this step, the remaining operations are post-processing and do not have privacy concerns. For completeness, we describe them below.

  We first ensure that all 24 marginals are consistent (summing up to be the same) and non-negative and get $\mathcal{M}_{T1'}$. We then adjust marginals in $\mathcal{M}_{P2}$ so that they are consistent with these 24 marginals. Let us call this updated version $\mathcal{M}_T = \mathcal{M}_{T1'} \cup \mathcal{M}_{P2'}$. We then use DPSyn [4, 5] to generate 10000 records using $\mathcal{M}_T$ (`method/dpsyn.py`). DPSyn is the method our team developed for synthetic data generation in last competition. Here because $\mathcal{M}_{T1}$ is already differentially private, using DPSyn can be thought of as a post-processing step. We found that DPSyn can only generate datasets with marginals similar to the target ones when the dataset is large. Here we need datasets of sizes n_i's, which range from a few hundreds to around 3000. Directly using DPSyn to synthesize such datasets result in dataset of poor qualities. So to obtain a smaller one of n_i records for each PUMA-year, we further process it using the following: to synthesize a set of n_i records, we randomly sample n_i records from the DPSyn dataset, and then iteratively update these n_i records to make its distributions close to the $\mathcal{M}_{T1'}$ obtained from the input dataset.

  We use $K = 100$ in this setting. We only spend privacy budget for obtaining the one-way marginals $\mathcal{M}_{T1}$; the remaining operations are part of the post-processing.

- **Using Two-way Marginals.** When ϵ is sufficiently large, we use Laplace Mechanism to obtain PUMA-year marginal, and use Gaussian Mechanism [3] together with zCDP [1] to obtain 253 2-way marginals from the input dataset ($\mathcal{M}_{T2}$).

  Similarly as above, we first ensure that all 253 marginals are consistent (summing up to be the same) and non-negative. We then use DPSyn to synthesize 10000 records using consistent marginals, $\mathcal{M}_{T2'}$. To synthesize a set of n_i records, we randomly sample n_i records from the DPSyn dataset, and then iteratively update the set to make its 2-way distributions closer to $\mathcal{M}_{T2'}$.

  We use $K = 100$ in this setting. We only spend privacy budget for obtaining the two-way marginals $\mathcal{M}_{T2}$; the remaining operations are part of the post-processing.

Choosing Among the Three Methods. To decide which method to use, we first consider ϵ, δ, and θ ("max_records_per_individual") from the json file. In addition, we also define the following threshold values:

$\epsilon_0 = \frac{\theta}{1400}$ (for total noisy count), $\epsilon_1 = \frac{\theta}{50}$ (low-end budget for PUMA-year marginal), $\epsilon_2 = \frac{\theta}{35}$ (mid-range budget for PUMA-year marginal), and $\epsilon_3 = \frac{\theta}{25}$ (high-end budget for PUMA-year marginal).

- When $\epsilon < \epsilon_2$, we use the low-range method: obtaining the PUMA-year marginal with all privacy budget, and performing synthesis using only Public Marginals.
- Otherwise, we choose whether to use the other two methods based on the size of the given dataset. In particular, we use budget ϵ_0 to get a noisy count $\tilde{N}$ of the total number of records, and then compute:

$$\tau_1 = \frac{1500 \cdot \theta}{\tilde{N}}$$
$$\tau_2 = 6 \cdot \tau_1$$

  - If $\epsilon < \epsilon_0 + \epsilon_1 + \tau_1$ (even if using low-end for PUMA-year, remaining budget too little for one-way marginals), then use $\epsilon - \epsilon_0$ to get a PUMA-year marginal, and perform synthesis using Public Marginals.
  - Else if $\epsilon < \epsilon_0 + \epsilon_1 + \tau_2$ (cannot get both low-end PUMA-year and 2-way marginals, settle for one-way), then let $x = \epsilon - \epsilon_0 - \epsilon_1 - \tau_1$, and use $y = \min(\epsilon_3, \epsilon_1 + x/2)$ to get PUMA-year marginal, and use Laplace mechanism with $\epsilon - \epsilon_0 - y$ to get all one-way marginals. We perform synthesis based on one-way marginals.
  - Else, let $x = \epsilon - \epsilon_0 - \epsilon_1 - \tau_2$, and $y = \min(\epsilon_3, \epsilon_1 + x/2)$. We use Laplace Mechanism with budget y to get PUMA-year marginal, and Gaussian mechanism with budget $\epsilon - \epsilon_0 - y$ to get all 2-way marginals. We use zCDP for composition [1]. We perform synthesis based on two way marginals.

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, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

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

[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. In *USENIX Security 21*, 2021.

[1] For zCDP, we have $\sqrt{\rho} = \sqrt{\epsilon + \log(1/\delta)} - \sqrt{\log(1/\delta)}$ (based on [1]), and for Gaussian mechanism, we have $\sigma = \frac{\theta}{\sqrt{2\rho'}}$, i.e., $\rho' = \frac{\theta^2}{2\sigma^2}$ (based on Lemma 1.6 of the zCDP paper), where ρ' is the privacy budget allocated for one marginal.