DPSyn: Differentially Private Synthetic Data Publication

Ninghui Li, Zhikun Zhang, Tianhao Wang
Team DPSyn, Purdue University
{ninghui, zhan3072, tianhaowang}@purdue.edu

Changes for Match 3

In match 3, as the dataset becomes different, so we make changes to our marginal selection mechanism. We choose different marginals, with new public information, as described in Section 2 (the underlying privacy proof in Section 3 is unchanged). We also modify the data synthesize procedure; but as it is post-processing of the results, we thus omit details of it. Finally, we fix the size of output dataset to be 662000.

On small input. The code is mainly for input datasets that are close to the Colorado state. When small datasets (e.g., only a few records) are given, the code can see that it cannot generate useful information for some marginal, and thus do not generate the marginal. When there is no marginals to be generated, the whole procedure will exit.

1 Overview

We present DPSyn, an algorithm for synthesizing microdata while satisfying differential privacy, and its instantiation to the dataset used in the competition, namely Public Use Microdata Sample (PUMS) of the 1940 USA Census Data. DPSyn has the following steps:

- **1. Select Configuration.** Based on the privacy parameter ϵ, DPSyn selects a set of marginals to be queried on the dataset.
- **2. Obtain Noisy Marginals.** Add noises to the selected marginals. This is the only step that we actually access the input dataset.
- **3. Generate Synthesized Datasets.** Synthesize the dataset using the noisy marginals. This step does not access the input dataset.

In this document, we focus on the proof that DPSyn satisfies differential privacy. In a high level, we show how the meta-data are used to select the marginals in the first step. We then use composition theorems to mathematically prove that by adding noise to the marginals, differential privacy is guaranteed. The procedure to generate the synthetic dataset is a post-processing step and thus omitted in the current document (the detailed of this step can be found in our concept paper [2] from the earlier phase of the challenge on HeroX).

2 Using Meta-data to Suggest Marginals

Designing configurations to choose from and the criteria for choosing which configuration to use is done manually, with knowledge of meta-data of the input dataset, including the domain size for all attributes, and certain semantic relationships among these attributes.

We obtain the meta-data from the website that holds the data. We use the data dictionary to come up with marginal configurations. The dictionary includes, for each attribute, the name, description, domain size, and where applicable the set of values.

2.1 Recoding and Compressing Attributes

To more effectively obtain information from marginals, we want to reduce the domain size of attributes. We apply recoding and compressing to one-way marginals as well as a few groups of attributes that are obviously highly correlated together. After obtain noisy marginals, we keep marginal cells that have count above a threshold θ. For the cells that are below θ, we add them up, if the total is below θ, we assign 0 to all these cells. If their total is above θ, then we create a new value to represent all values that have low counts. After synthesizing the dataset, this new value is replaced by the values it represents using the original noisy marginal.

The threshold is chosen according to the following formula:

$$\theta = \max(4.5\sigma, 800) \tag{1}$$

Here σ is the standard deviation for Gaussian noises added to the marginal, and $\tilde{N}$ is an estimation of the total record count. The logic for using 4.5σ is that values below that are likely to be result of adding noises to 0 or very small counts, whereas values above that are much less likely to be just result of noises.

In the past, we use $\frac{\tilde{N}}{3000}$ (which is about 220 when $N \approx 660,000$) instead of 800 in the above formula. To make the code able to deal with input datasets of different sizes, we use a fixed number now. We use a larger threshold now because we now aggregate all values lower than the threshold into one new value. This enables us to use a higher threshold without missing a significant portion of the density. Also, given that we are using around 200 marginals, under the higher end of privacy budget ϵ is between 1 and 8 and $\delta = 1/(662,000^2)$, the standard deviation of the Gaussian noises added to the marginals is between around 100 to 14. If one attribute value has count lower than several hundreds, two-way marginals involving that value would likely be dominated by noises. We expect that the algorithm to perform similar when the value 800 is replaced with a similar one, e.g., 600 or 1000.

2.2 Handling the Attributes

We handle different attributes using different strategies based on their semantic meanings. In summary, we have used the following strategies:

- For attributes with large domain, we apply four different strategies to reduce the domain.

  1. **Read from dataset.** According to the forum post, we extract the distinct values of attributes CITY, COUNTY and METAREAD from the original dataset. This strategy do not consume privacy budget.

  2. **Use codebook.** The codebook is the public information on the IPUMS website. We use it to reduce the domain of attributes that have large domain size but only a small number of legitimate values. For example, the domain size of attribute RACED is 621 (because its maxval in specs file is 620), while its number of legitimate values is only 238 which is much smaller than the original domain. Since this strategy do not touch the original dataset, it will not consume privacy budget. This strategy also applies to attribute MTONGUED, EDUCD, BPL, *etc.*

  3. **Compress attributes.** Using the method discussed in Section 2.1, we compress values with estimations below the threshold into a dummy value. For example, in RACED, although there are 238 legitimate values, the number of values that appear in the Colorado dataset is about 11; thus, we compress it to further reduce the domain: We first use Gaussian mechanism (consume privacy budget) to obtain the histogram, and then take only values with estimations above the threshold given in Equation (1). This strategy also applies to attribute MIGPLAC5, MIGSEA5, MIGMET5, *etc.* Furthermore, we also use this strategy to combine several attributes together (the domain becomes the Cartesian product of the original domains) and then compress it to a new attribute with smaller domain, *e.g.*, we combine and then compress SPLIT + GQ + GQFUNDS to one new attribute SPL-GQ-GQF.

  4. **Bucketize attributes into bins.** For some attributes, we use its bucketized domain to generate marginals with other attributes. At the same time, we also generate one-way marginals for

these attributes using their original value. When generating the synthetic dataset, the coarse-grained values are used; but after the synthesizing procedure, we use the distribution of the one-way marginal to replace the bucketized value. Generating both one-way marginals and two-way marginals consume privacy budget. This strategy applies to attributes VALUEH, RENT, ENUMDIST, FAMSIZE, WKSWORK1, HRSWORK1 and AGE. Specifically, we use two different granularity for AEG to generate marginals with different attributes, which will depict in the next section. We use the bucketizing scheme of WKSWORK2 and HRSWORK2 for WKSWORK1 and HRSWORK1, since WKSWORK2 and HRSWORK2 are the bucketized version of WKSWORK1 and HRSWORK1 in nature.

- Generate 1-way marginal to capture the distribution of single attribute, *e.g.*, AGEMONTH, DURUNEMP, CITIZEN, *etc.* This strategy consumes privacy budget.
- Generate 2-way marginals to capture the correlation between two attributes, *e.g.*, AGE and EDUCD, AGE and EMPSTATD, CITY and METAREAD, *etc.* This strategy consumes privacy budget.
- Directly fill one attribute based on its mapping relationship with another attribute using the public information. This strategy do not consume privacy budget. Specifically, we use the following three mappings:
  1. Fill one attribute's general version using its detailed version. The mapping from detailed version to general version can be obtained from the codebook. For example, one can first synthesize the EDUCD attribute, and then fill EDUC based on their mapping relationship without consuming privacy budget.
  2. Synthesize BPL and fill BPLD using the formula BPL code × 100. This mapping assumes that for each code of BPL, the first value in BPLD dominate the others. The same mapping applies to MBPL and MBPLD, FBPL and FBPLD.
  3. Synthesize CITYPOP and fill SIZEPL using the fact that SIZEPL is the bucketized version of CITYPOP, *i.e.*, for each value of CITYPOP, we can determine its range in SIZEPL from the codebook.
- Use noisy marginals to figure out the mapping between two attributes; then, fix one attribute and use this mapping to fill another attribute. This strategy consumes privacy budget. We have four cases:
  1. Generate two-way marginal for COUNTY and SEA; then, for each value of COUNTY, we map it to the value of SEA with largest noisy count.
  2. Generate two-way marginal for COUNTY and SUPDIST. Then, for each COUNTY, keep the values of SUPDIST with noisy count larger than 3σ, where σ is the standard deviation of Gaussian noise. Finally, fill in SUPDIST using the two-way marginal.
  3. Generate two-way marginal for CITYPOP and URBPOP. After the synthesizing procedure, fill URBPOP based on the following rule: when CITYPOP equals 0, fill URBPOP based on URBPOP's distribution in two-way marginal; when CITYPOP ≥ 25, use CITYPOP; otherwise fill 0.
  4. Generate one-way marginal for CITIZEN. When BPL ≤ 99, set CITIZEN $= 0$; otherwise, fill CITIZEN based on the one-way marginal.
- For INCWAGE attribute, we use the bucketized value to generate two-way marginals with work-related attributes and three-way marginal with CITY and SEX. We apply different bucketizing scheme for different privacy budget. Then, we use the one-way marginal of values $\{[0, 5000], 999998\}$ to replace the bucketized value. Generating all the marginals consumes privacy budget.

2.3 Synthesizing dataset

We divide all attributes into different groups based on the marginals, ensuring there are no inter-group marginals among groups. Within each group, we can generate 1-way, 2-way or 3-way marginals among all attributes. Then, one synthesize four separate datasets and join them. Specifically, the attributes are divided into the following four groups:

- **Group A.** This group comprises race-related attributes and misc information about individual person, such as RACED, HISPAND, GQTYPED, *etc.* We assume these attributes have no correlation with other attributes, thus have no inter-group marginals with attributes in other groups.
- **Group AGE1.** This group consists of person's individual information such as AGE, BPL, FBPL, MBPL, *etc.* In this group, AGE is bucketized into coarser-grained bins to generate marginals with other attributes.
- **Group AGE2.** This group contains most of the important information about one person, including work-related attributes, geo-spatial-related attributes and migration-related attributes. Attribute AGE is also contained in this group and is bucketized into finer-grained bins.
- **Group SL.** This group includes the sample-line indicator attribute SLREC and all the sample-line attributes such as SSENROLL, MTONGUED, CHBORN, *etc.* We only generate marginals of the sample-line attributes when SLREC equals 2. When SLREC equals 1, all sample-line attributes use their default values.

After synthesizing four separate datasets, we join them using the following strategies.

- Directly join group A and group AGE1, since there are no inter-group attributes between them.
- For each coarser-grained value in group AGE1, join it with all the records containing the corresponding finer-grained values in group AGE2.
- When join group SL with group AGE1, ensuring that if FBPL is 0 or MBPL is 0, SLREC equals 1.

3 Privacy Analysis

From the description of DPSyn, the only step where the raw data is touched is the step to generate the marginals. The Gaussian/Laplacian noise is then added to the marginals to make sure DP is guaranteed. After that, the consistent step manipulates the noisy marginals; then the results are used to generate the synthetic dataset. To prove the overall process satisfies DP, it suffices to prove that the Gaussian/Laplacian noise added satisfies DP. We start by reviewing the formal definition of differential privacy.

3.1 Differential Privacy

Definition 1 ((ϵ, δ)-DP)**.** *A randomized mechanism* $\mathbf{A}$ *satisfies* (ϵ, δ)*-differential privacy if for any pair of neighboring datasets* D *and* D'*, and any* $x \in \mathit{Range}(\mathbf{A})$*,*

$$\Pr\left[\mathbf{A}(D) = x\right] \leq e^{\epsilon} \cdot \Pr\left[\mathbf{A}(D') = x\right] + \delta.$$

In this paper we consider two datasets D and D' to be *neighbors* if and only if either $D = D' + r$ or $D' = D + r$, where $D + r$ denotes the dataset resulted from adding the record r to the dataset D. This protects the privacy of any single record, because even if one leaves r out of the dataset, in which case the privacy of r can be considered to be protected, one may still publish the same outputs with a similar probability. Differential privacy enjoys the property of sequential composition:

Theorem 1 (Composition of DP)**.** *If a mechanism* $\mathbf{A}$ *consists of a sequence of adaptive mechanisms* $\mathbf{A}_1, \ldots, \mathbf{A}_t$ *such that for any* $i \in [t]$*,* $\mathbf{A}_i$ *guarantees* $(\varepsilon_i, \delta_i)$*-DP, then* $\mathbf{A}$ *guarantees* $(\sum_{i=1}^{t} \varepsilon_i, \sum_{i=1}^{t} \delta_i)$*-DP.*

Sequential composition can be used to compose a small number of DP mechanisms. When there are many mechanisms to compose, sequential composition gives poor privacy guarantee. Recently, several advanced techniques are proposed to improve the privacy guarantee after composition. In the following, we use Rényi DP to prove DPSyn satisfies (ε, δ)-DP.

3.2 Composition via Rényi DP

Rényi Differential Privacy (RDP) [4] generalizes differential privacy and is is useful as a more natural analysis framework when dealing with Gaussian noise. Defined below, the RDP of a mechanism is stated in terms of the Rényi divergence.

Definition 2 (Rényi Divergence). *The Rényi divergence of order λ between two distributions P and Q is defined as:*

$$D_\lambda(P\|Q) \triangleq \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim Q}\left[(P(x)/Q(x))^{\lambda}\right] = \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim P}\left[(P(x)/Q(x))^{\lambda-1}\right].$$

Definition 3 (Rényi Differential Privacy (RDP)). *A randomized mechanism $\mathbf{A}$ is said to guarantee (λ,ε)-RDP with $\lambda \geq 1$ if for any neighboring datasets D and D',*

$$D_\lambda(\mathbf{A}(D)\|\mathbf{A}(D')) = \frac{1}{\lambda-1}\log \mathbb{E}_{x\sim \mathbf{A}(D)}\left[\left(\frac{\Pr\left[\mathbf{A}(D)=x\right]}{\Pr\left[\mathbf{A}(D')=x\right]}\right)^{\lambda-1}\right] \leq \varepsilon.$$

We first present several key facts that allow easy composition of RDP guarantees and their conversion to (ε,δ)-differential privacy bounds. The corresponding proofs are deferred to the Appendix.

Theorem 2 (From Gaussian to RDP). *The Gaussian Mechanism with variance σ satisfies $(\lambda, \lambda\mu^2/2\sigma^2)$-RDP, given the sensitivity being μ.*

Theorem 3 (Composition). *If a mechanism $\mathbf{A}$ consists of a sequence of adaptive mechanisms $\mathbf{A}_1,\ldots,\mathbf{A}_t$ such that for any $i\in[t]$, $\mathbf{A}_i$ guarantees (λ,ε_i)-RDP, then $\mathbf{A}$ guarantees $(\lambda,\sum_{i=1}^{t}\varepsilon_i)$-RDP.*

Theorem 4 (From RDP to DP). *If a mechanism $\mathbf{A}$ guarantees (λ,ε)-RDP, then $\mathbf{A}$ guarantees $(\varepsilon+\frac{\log 1/\delta}{\lambda-1},\delta)$-differential privacy for any $\delta\in(0,1)$.*

The three theorems can be chained to prove that given a Gaussian mechanism, we obtain the RDP, then compose several mechanisms, and finally convert RDP to DP. To find such a Gaussian mechanism, we reverse the order of the theorems where given the privacy budget ϵ, δ and the number of marginals t, we first define ϵ_0 so that $\epsilon=\epsilon_0-\frac{\ln 1/\delta}{\lambda-1}$. (ϵ,δ)-DP is then converted to (λ,ϵ_0)-RDP. Finally each of the t Gaussian noises should satisfy $(\lambda,\epsilon_0/t)$-RDP, which give the standard deviation for each Gaussian noise:

$$\sigma=\sqrt{\frac{t\lambda}{2\epsilon_0}}=\sqrt{\frac{t\lambda}{2\left(\epsilon-\frac{\ln 1/\delta}{\lambda-1}\right)}}=\sqrt{\frac{t\lambda(\lambda-1)}{2\epsilon(\lambda-1)-\ln 1/\delta}}$$

By varying λ, one can obtain the optimal σ.

Theorem 5. *When t Gaussian Mechanism are applied, each with variance $\sigma=\min_\lambda\sqrt{\frac{t\lambda}{2\epsilon_0}}$, the overall mechanism satisfies (ϵ,δ)-DP, where $\epsilon_0=\epsilon-\frac{\ln 1/\delta}{\lambda-1}$.*

Proof. We prove a more general case that for any λ, $\sigma=\sqrt{\frac{t\lambda}{2\epsilon_0}}$ leads to (ϵ,δ)-DP, which makes $\sigma=\min_\lambda\sqrt{\frac{t\lambda}{2\epsilon_0}}$ a special case that satisfies (ϵ,δ)-DP.

First of all, we prove that in our case, the sensitivity is 1. Specifically, we add Gaussian noise to the marginals. Based on the definition neighboring datasets, adding or removing one record, only one number in the marginal will change by one, thus the ℓ_2 distance of the marginals from two neighboring datasets is 1. Now we proceed with composition.

By Theorem 2, the Gaussian mechanism with $\sigma=\sqrt{\frac{t\lambda}{2\epsilon_0}}$ satisfies $(\lambda,\lambda\mu^2/2\sigma^2)$-RDP. With $\mu=1$, $\lambda\mu^2/2\sigma^2=\lambda/2\frac{t\lambda}{2\epsilon_0}=\epsilon_0/t$; thus the Gaussian mechanism with σ achieves $(\lambda,\epsilon_0/t)$-RDP for any λ (note that here ϵ_0 is a variable of λ). After composing the same Gaussian mechanism t times, by Rényi composition theorem (Theroem 3), we have (λ,ϵ_0)-RDP. Now we can convert RDP to DP by Theorem 4 and obtain $(\varepsilon_0+\frac{\log 1/\delta}{\lambda-1},\delta)$-DP. Finally, by the assumption that $\epsilon_0=\epsilon-\frac{\ln 1/\delta}{\lambda-1}$, (ϵ,δ)-DP is achieved. □

Finally, we can prove that our method achieves (ϵ, δ)-DP.

Theorem 6. DPSyn *is* (ϵ, δ)*-DP.*

Proof. Depending on the configuration, DPSyn will obtain two kinds of marginals

- $k_1 \geq 0$ marginals, where to each cell an independent random noise drawn from the Laplace distribution with scale β.
- k_2 marginals, where to each cell an independent random noise drawn from the Gaussian distribution with standard deviation $\sigma = \min_\lambda \sqrt{\frac{k_2\lambda}{2\epsilon_0}}$, where $\epsilon_0 = \epsilon - \frac{k_1}{\beta} - \frac{\ln 1/\delta}{\lambda - 1}$.

By sequential composition (Theorem 1), the k_1 marginals satisfy $\left(\frac{k_1}{\beta}, 0\right)$-DP. For the k_2 marginals, by Theorem 5, the application of the k_2 Gaussian Mechanisms satisfies $\left(\epsilon - \frac{k_1}{\beta}, \delta\right)$-DP. Taking the two sets of marginals together, by sequential composition, the overall mechanism is proven to satisfy (ε, δ)-DP. □

References

[1] A. Langlois, D. Stehlé, and R. Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–256. Springer, 2014.

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

[3] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013.

[4] I. Mironov. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

A Supplementary Proofs

Statement of Theorem 2. The Gaussian Mechanism with variance σ satisfies $(\lambda, \lambda\mu^2/2\sigma^2)$-RDP, given the sensitivity being μ.

Proof. By direct computation we verify that

$$\begin{aligned}
&D_\lambda(N(0,\sigma^2)\|N(\mu,\sigma^2)) \\
&\quad = \frac{1}{\lambda-1}\log\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}\exp(-\lambda x^2/(2\sigma^2))\cdot\exp(-(1-\lambda)(x-\mu)^2/(2\sigma^2))\,\mathrm{d}x \\
&\quad = \frac{1}{\lambda-1}\log\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp[(-x^2+2(1-\lambda)\mu x-(1-\lambda)\mu^2)/(2\sigma^2)]\,\mathrm{d}x \\
&\quad = \frac{1}{\lambda-1}\log\left\{\frac{\sigma\sqrt{2\pi}}{\sigma\sqrt{2\pi}}\exp\left[(\lambda^2-\lambda)\mu^2/(2\sigma^2)\right]\right\} \\
&\quad = \lambda\mu^2/(2\sigma^2).
\end{aligned}$$

□

Statement of Theorem 3. If a mechanism $\mathbf{A}$ consists of a sequence of adaptive mechanisms $\mathbf{A}_1, \ldots, \mathbf{A}_t$ such that for any $i \in [t]$, $\mathbf{A}_i$ guarantees (λ, ε_i)-RDP, then $\mathbf{A}$ guarantees $(\lambda, \sum_{i=1}^t \varepsilon_i)$-RDP.

Proof. By induction, it suffices to prove that given $\mathbf{A}_1$ and $\mathbf{A}_2$ are (λ, ε_1)-RDP and (λ, ε_2)-RDP, respectively, $\mathbf{A}$ consisted of $\mathbf{A}_1$ and $\mathbf{A}_2$ guarantees $(\lambda, \varepsilon_1 + \varepsilon_2)$-RDP.

Suppose the range of $\mathbf{A}_1$ and $\mathbf{A}_1$ are R_1 and R_2, respectively. We write X, Y, and Z for the distributions $\mathbf{A}_1(D)$, $\mathbf{A}_2(X, D)$, and the joint distribution $(X, Y) = Z$. X', Y', and Z' are similarly defined if the input is D'. Then

$$\begin{aligned}
&\exp\left[(\lambda-1)D_\lambda(\mathbf{A}(D)\|\mathbf{A}(D'))\right] \\
&= \int_{R_1\times R_2} Z(x,y)^\lambda Z'(x,y)^{1-\lambda}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{R_1}\int_{R_2} (X(x)Y(x,y))^\lambda (X'(x)Y'(x,y))^{1-\lambda}\,\mathrm{d}y\,\mathrm{d}x \\
&= \int_{R_1} X(x)^\lambda X'(x)^{1-\lambda}\left\{\int_{R_2} Y(x,y)^\lambda Y'(x,y)^{1-\lambda}\,\mathrm{d}y\right\}\mathrm{d}x \\
&\le \int_{R_1} X(x)^\lambda X'(x)^{1-\lambda}\,\mathrm{d}x \cdot \exp((\lambda-1)\varepsilon_2) \\
&\le \exp((\lambda-1)\varepsilon_1)\exp((\lambda-1)\varepsilon_2) \\
&= \exp((\lambda-1)(\varepsilon_1+\varepsilon_2)),
\end{aligned}$$

from which the claim follows. □

Before proving Theorem 4, we first state the following proposition that appears in Langlois et al. [1], which generalizing Lyubashevsky et al. [3].

Proposition 1 (Probability preservation [1]). *Let $\lambda > 1$, P and Q be two distributions defined over R with identical support, $A \subset R$ be an arbitrary event. Then*

$$P(A) \le \left(\exp[D_\lambda(P\|Q)] \cdot Q(A)\right)^{(\lambda-1)/\lambda}.$$

Proof. The result follows by application of Hölder's inequality, which states that for real-valued functions f and g, and real $p, q > 1$, such that $1/p + 1/q = 1$,

$$\|fg\|_1 \le \|f\|_p \|g\|_q.$$

By setting $p \triangleq \lambda$, $q \triangleq \lambda/(\lambda-1)$, $f(x) \triangleq P(x)/Q(x)^{1/q}$, $g(x) \triangleq Q(x)^{1/q}$, and applying Hölder's, we have

$$\begin{aligned}
\int_A P(x)\,\mathrm{d}x &\le \left(\int_A P(x)^\lambda Q(x)^{1-\lambda}\,\mathrm{d}x\right)^{\frac{1}{\lambda}}\left(\int_A Q(x)\,\mathrm{d}x\right)^{\frac{\lambda-1}{\lambda}} \\
&\le \exp[D_\lambda(P\|Q)]^{(\lambda-1)/\lambda} Q(A)^{(\lambda-1)/\lambda},
\end{aligned}$$

completing the proof. □

Statement of Theorem 4. If a mechanism $\mathbf{A}$ guarantees (λ, ε)-RDP, then $\mathbf{A}$ guarantees $(\varepsilon + \frac{\log 1/\delta}{\lambda-1}, \delta)$-differential privacy for any $\delta \in (0, 1)$.

Proof. Take any two adjacent inputs D and D', and $S \subseteq \mathit{Range}(\mathbf{A})$. To show that $\mathbf{A}$ is (ε', δ)-differentially private, where $\varepsilon' = \varepsilon + \frac{1}{\lambda-1}\log 1/\delta$, we need to demonstrate that $\Pr[\mathbf{A}(D) \in S] \le e^{\varepsilon'}\Pr[\mathbf{A}(D') \in S] + \delta$. In fact, we prove a stronger statement that $\Pr[\mathbf{A}(D) \in S] \le \max(e^{\varepsilon'}\Pr[\mathbf{A}(D') \in S], \delta)$.

Recall that by Proposition 1

$$\Pr[\mathbf{A}(D) \in S] \le \{e^\varepsilon \Pr[\mathbf{A}(D') \in S]\}^{1-1/\lambda}.$$

Denote $\Pr[\mathbf{A}(D') \in S]$ by Q and consider two cases.

Case I. $e^\varepsilon Q > \delta^{\lambda/(\lambda-1)}$. Continuing the above,

$$\begin{aligned}\mathsf{Pr}\left[\mathbf{A}(D) \in S\right] &\leq \{e^\varepsilon Q\}^{1-1/\lambda} = e^\varepsilon Q \cdot \{e^\varepsilon Q\}^{-1/\lambda}\\ &\leq e^\varepsilon Q \cdot \delta^{-1/(\lambda-1)}\\ &= \exp\left(\varepsilon + \frac{\log 1/\delta}{\lambda - 1}\right) \cdot Q.\end{aligned}$$

Case II. $e^\varepsilon Q \leq \delta^{\lambda/(\lambda-1)}$. This case is immediate since

$$\mathsf{Pr}\left[\mathbf{A}(D) \in S\right] \leq \{e^\varepsilon Q\}^{1-1/\lambda} \leq \delta,$$

which completes the proof. □