Differentially Private Synthetic Data Using KD-Trees

Eleonora Kreačić[1] **Navid Nouri**[1] **Vamsi K. Potluru**[1] **Tucker Balch**[1] **Manuela Veloso**[1]

[1]J.P. Morgan AI Research

Abstract

Creation of a synthetic dataset that faithfully represents the data distribution and simultaneously preserves privacy is a major research challenge. Many space partitioning based approaches have emerged in recent years for answering statistical queries in a differentially private manner. However, for synthetic data generation problem, recent research has been mainly focused on deep generative models. In contrast, we exploit space partitioning techniques together with noise perturbation and thus achieve intuitive and transparent algorithms. We propose both data independent and data dependent algorithms for ϵ-differentially private synthetic data generation whose kernel density resembles that of the real dataset. Additionally, we provide theoretical results on the utility-privacy tradeoffs and show how our data dependent approach overcomes the curse of dimensionality and leads to a scalable algorithm. We show empirical utility improvements over the prior work, and discuss performance of our algorithm on a downstream classification task on a real dataset.

1 INTRODUCTION

Publishing data of a highly sensitive nature in domains of finance or health, carries a risk of compromising privacy of individuals and therefore a breach of privacy regulations (e.g. HIPPA, FCRA, GDPR). This limitation can be potentially circumvented by the use of synthetic data. However, synthetic data per se is not inherently private [Jordon et al., 2022]. In this paper, we study the problem of publishing a synthetic dataset that faithfully represents the original data whilst at the same time does not comproimise privacy of individuals in the original.

Preserving privacy of individuals while publishing a dataset for public use is a known challenge [Balog et al., 2018]. A de facto standard for privacy is differential privacy (DP), which is widely used in the literature and in practice. Many existing works aim to preserve quality of differentially private answers for a certain class of queries [Mohammed et al., 2011, Xiao et al., 2014, Hardt et al., 2012]; see Zhu et al. [2017] for a survey. However, a more general problem studies the release of a differentially private synthetic dataset that can be used for downstream tasks without additional privacy leaks. Recent work mainly applied generative adversarial networks (GAN) [Goodfellow et al., 2014] and utilized divergence metrics such as Jensen-Shannon divergence and Wasserstein distance as a metric of quality to compare synthetic and original datasets [Park et al., 2018, Torkzadehmahani et al., 2019, Jordon et al., 2019, Xie et al., 2018, Frigerio et al., 2019, Papernot et al., 2016, 2018]. Another class of utility metrics is based on kernels e.g., distance in reproducing kernel Hilbert space (RKHS) or similarly maximum mean discrepancy (MMD), see Section 1.3.1 for a detailed discussion. The advantage of kernel based metrics is that they compare two probability measures in terms of all possible moments [Harder et al., 2021] and thus capture

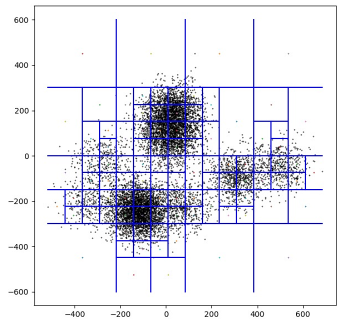


Figure 1: Data dependent partitioning in $\mathbb{R}^2$. Our data dependent algorithm achieves more refined partitioning in densely populated areas and thus better utility of DP synthetic data.

a wide class of statistical properties of the dataset.

We propose algorithms that for an input dataset output its differentially private synthetic counterpart. The key techniques used in this work are space partitioning [Ram and Gray, 2013] and noisy perturbation. Similar ideas have been used in the literature for designing differentially private statistical query algorithms and histogram release applications, e.g., [Qardaji et al., 2013, Xiao et al., 2014, Zhang et al., 2016]. However, naive implementation of space partitioning techniques for DP dataset release problem suffers from the curse of dimensionality. We show how our data-dependent approach solves this issue and also leads to scalable algorithms. As our method is based on the simple idea of space partitioning, the advantage in comparison to the state of the art generative models is interpretability of our algorithms.

1.1 RELATED WORK

Prior work which consider similar settings to ours is relatively sparse. The closest are Balog et al. [2018] and Harder et al. [2021], although the latter is more focused on labeled data as well as image data. Both of these papers aim to release a dataset that is close to the original one in terms of kernel RKHS distance while preserving privacy. Balog et al. [2018] presents two algorithms. Their first algorithm either needs a small fraction of the dataset to be publicly available, or reweights a sampled set of points from the support of the underlying distribution. The requirement on the public input data represents a limitation which we overcome. Their second algorithm is based on random features and an iterative gradient based optimization procedure to apply reduced set method which has the twin issues of being slow in high dimensions and also suffers from the lack of interpretability. Harder et al. [2021] improved upon this work and achieved better results, and in particular for lower dimensions, say 5, but not when the dimensions are much higher. Their approach relies on deep generative model to minimize MMD, and thus again suffers the lack of interpretability. None of these works provides theoretical results on the utility-privacy trade-off. Our algorithms are transparent which enables us to provide theoretical guarantees on utility. Our data dependent algorithm achieves good utility in high dimensions.

1.2 OUR CONTRIBUTION

Inspired by locality sensitive hash functions and nearest neighbor search algorithms introduced by Indyk and Motwani [1998], our approach is based on space partitioning schemes and in particular KD-trees Friedman et al. [1977]. Our goal is to output a synthetic dataset that imposes similar kernel density on the space as the input data, and does not compromise privacy of the input. Intuitively, if we partition the space into small sections (bins) and preserve the ratio of points after noise addition (for the purpose of privacy) we will approximately preserve the kernel density. Based on this idea, we introduce two algorithms that yield ϵ-differentially private synthetic data outputs. Our data independent approach, Algorithm 1, implements a naive version of the idea and naturally suffers from the curse of dimensionality. Our data dependent algorithm, Algorithm 2, overcomes this and achieves better utility. Our main contributions are:

- We provide an upper bound on utility loss of our data independent algorithm for a general setting of unknown distribution of input data (Theorem 3 and Theorem 5). We improve the bound for a special case of input data from a mixture of Gaussians in $\mathbb{R}^d$ (Theorem 6).
- Our data dependent algorithm achieves smaller number of empty bins and more refined partitioning in densely populated areas which yields better utility. We overcome the curse of dimensionality using an implicit sampling of empty bins (Theorem 7 and Theorem 8).
- Unlike previous approaches [Balog et al., 2018, Harder et al., 2021], we do not rely on black box methods, and thus achieve interpretability which is important in many practical settings. In addition, we do not require a fraction of input dataset to be public.
- In Section 4 we show how: (i) our algorithms outperform algorithms by Balog et al. [2018], (ii) our data dependent algorithm overcomes curse of dimensionality, (iii) empirical utility loss compares to our theoretical bound from Theorem 6, (iv) performance of our algorithms on downstream binary classification task compares with Harder et al. [2021].

1.3 BACKGROUND

We give an introduction to MMD and differential privacy.

1.3.1 Kernel Density Estimates and RKHS distance

For a (unweighted) dataset $P \subset \mathbb{R}^d$ and a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the kernel density (KD) with respect to P is defined at any point $x \in \mathbb{R}^d$ as $\mathrm{KD}_P^K(x) = \frac{1}{|P|}\sum_{p\in P} K(x,p)$. If P is equipped with weights such that $\sum_{p\in P} w_p = 1$, then kernel density is given by $\mathrm{KD}_P^K(x) = \sum_{p\in P} w_p K(x,p)$. For ease of notation, we will often write $\mathrm{KD}_P(\cdot)$ instead of $\mathrm{KD}_P^K(\cdot)$. For the two datasets P and Q, ℓ_∞ distance of two KDs is defined as $\|\mathrm{KD}_P - \mathrm{KD}_Q\|_\infty = \sup_{x\in\mathbb{R}^d} |\mathrm{KD}_P(x) - \mathrm{KD}_Q(x)|$. If K is positive definite, then $K(p,x)$ can be represented as an inner product in RKHS $\mathcal{H}_K$. That is, there is $\phi_K : \mathbb{R}^d \to \mathcal{H}_K$ such that $\phi_K(x) = K(x,\cdot)$. For a positive definite kernel K, if ϕ_K is injective then MMD is given by

$$\mathrm{MMD}(P,Q) = \sqrt{\kappa(P,P) + \kappa(Q,Q) - 2\kappa(P,Q)} \quad (1)$$

where $\kappa(P,Q) = \frac{1}{|P|}\frac{1}{|Q|}\sum_{p\in P}\sum_{q\in Q} K(p,q)$ represents a kernel metric between two datasets. It is possible to convert between bounds on ℓ_∞-distance of KDs and MMD. More precisely, $\mathrm{MMD}(P,Q) \leq \epsilon$ implies $\|\mathrm{KD}_P - \mathrm{KD}_Q\|_\infty \leq \epsilon$, and also $\|\mathrm{KD}_P - \mathrm{KD}_Q\|_\infty \leq \epsilon$ implies $\mathrm{MMD}(P,Q) \leq \sqrt{2\epsilon}$. Further details can be found in Phillips and Tai [2020] and Harder et al. [2021]. We present theoretical analysis on utility guarantees in terms of bounds on KD distance, whilst in experiments we rely on MMD due to the ease of computation.

1.3.2 Differential Privacy

Differential privacy (DP) [Dwork et al., 2006] has become a de facto standard to quantify privacy leakage. It provides theoretical guarantees that potential adversary with the knowledge of the output is not able to distinguish whether a particular individual was present in the input dataset.

Definition 1. *[Dwork et al., 2014] A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is ϵ-differentially private if for any two datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ that differ in only one entry, we have*

$$\forall \mathcal{C} \subseteq \mathcal{Y}, \quad \mathbb{P}(\mathcal{M}(\mathcal{D}) \in \mathcal{C}) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(\mathcal{D}') \in \mathcal{C}). \tag{2}$$

Standard way to achieve ϵ-DP is to employ Laplace mechanism, i.e., add Laplace noise to the output. More precisely, for a function f computed on sensitive data $\mathcal{D}$, we introduce $\mathcal{M}_{Lap}(\mathcal{D}, f(.), \epsilon) = f(\mathcal{D}) + \mathrm{Lap}(0, \Delta_f/\epsilon)$, where $\Delta_f = \max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_1$ is the ℓ_1 sensitivity of f with respect to change of a single entry in the dataset ($\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ are two neighboring datasets, i.e., that they differ in only one entry) and Lap denotes Laplace distribution parametrized by the mean and scale. **Post-processing property of ϵ-DP** guarantees that composition of any data independent function with the output of ϵ-DP mechanism is also ϵ-DP, i.e., it does not incur additional privacy leaks [Dwork et al., 2014]. This means that differentially private synthetic data can be safely used for downstream tasks. **Composition of ϵ-DP** guarantees that combination of ϵ_1-DP algorithm $\mathcal{M}_1$ and ϵ_2-DP algorithm $\mathcal{M}_2$ defined by $\mathcal{M}_{1,2} = (\mathcal{M}_1, \mathcal{M}_2)$ is $(\epsilon_1 + \epsilon_2)$-DP [Dwork et al., 2014].

2 PROBLEM FORMULATION

We are given a multidimensional numerical dataset $P = \{p_1, p_2, \ldots, p_n\}$ of n records in $\mathbb{R}^d$. Our task is to design a differentially private algorithm that outputs (possibly weighted) dataset $Q = \{(q_1, w_1), \ldots, (q_m, w_m)\}$ where $q_i \in \mathbb{R}^d$, $w_i \in \mathbb{R}^+$, such that for any $x \in \mathbb{R}^d$

$$\mathrm{KD}_P^K(x) \approx \mathrm{KD}_Q^K(x),$$

where K is some positive definite kernel. The closeness of KD_P and KD_Q in ℓ_∞-distance implies that relying on Q instead of P leads consistent estimation of population statistics of original dataset P (see Balog et al. [2018] for discussion) i.e., synthetic dataset Q faithfully represents the original P. In other words, both ℓ_∞-distance of KDs and MMD represent good utility measures when evaluating quality of synthetic datasets. In the rest of the paper, for simplicity of presentation we focus on Gaussian kernel $K(x,p) = e^{-\frac{\|x-p\|_2^2}{2\sigma^2}}$. It is however straightforward to adapt our analysis to a wider class of kernels.

3 OUR ALGORITHMS

We propose two algorithms for synthetic data generation in the next sections: (i) Data independent (ii) Data dependent.

3.1 DATA INDEPENDENT

In this section, we present data independent algorithm for synthetic dataset release with DP guarantees. Inspired by the widely used idea of space partitioning, we want to partition the space into a number of bins, e.g., J cubes of width w. Then we count the number of points inside each bin and present these counts on a J-dimensional vector. Any single data point can affect this vector at most by a constant in terms of ℓ_1 distance. In other words, it has a bounded ℓ_1 sensitivity with respect to any two neighbouring datasets (see Section 1.3.2). Thus, we can employ Laplace mechanism in order to achieve ϵ-DP. Bins with the noisy count below input threshold t will be removed i.e., filtered out. The algorithm outputs the dataset consisting of centers of the bins that survived filtering step and the corresponding noisy point counts. See Algorithm 1.

Algorithm 1 Data independent binning

```
1:  procedure DataIndependent($P, \epsilon, t, w$)
2:                  ▷ $P$ is the original dataset, $\epsilon$ is privacy budget
3:                           ▷ $t$ is filtering threshold, $w$ is bin width
4:      $R \leftarrow$ the edge length of the axes aligned hypercube
        that encompasses $P$
5:      Apply binning using bins of width $w$
6:      $J \leftarrow (R/w)^d$                        ▷ Number of bins
7:      $\mathbf{v} \in \mathbb{R}^J \leftarrow$ vector of point counts bin by bin
8:      $c_1, c_2, \ldots, c_J \leftarrow$ centers of bins
9:      $\widetilde{\mathbf{v}} \leftarrow \mathbf{v} + Lap(\frac{2}{\epsilon} I_{J\times J})$          ▷ Noisy point counts
10:     For any $i \in [J]$, if $\widetilde{\mathbf{v}}_i < t$, then $\widetilde{\mathbf{v}}_i \leftarrow 0$
11:                  ▷ **Filtering step:** Removing bins below $t$
12:     Output $Q := \{(c_i, \widetilde{\mathbf{v}}_i)$ for $i \in [J]$ if $\widetilde{\mathbf{v}}_i > 0\}$
```

First we prove that the output of Algorithm 1 is differentially private. Then, we analyze the performance of Algorithm 1 when $t = 0$ in terms of the worst-case utility-privacy trade-off, i.e. the case of a general input dataset where we do not impose any assumptions on its distribution. Finally, we present the utility-privacy trade-off of Algorithm 1 for the special case of input data coming from a mixture of Gaussians with a positive filtering threshold. Theorems 3 and 5 give clues on how to set up width w and threshold t in order to achieve better utility.

3.1.1 Differential privacy

Theorem 2 (DP). *Output of Algorithm 1 is ϵ-DP.*

Proof. Note that bins centers are picked independently of the input data.[1] For J-dimensional point counts (line 7 of Algorithm 1) v and $\hat{v}$ corresponding to two datasets P and $\hat{P}$ that differ in exactly one element, we have $||v - \hat{v}||_1 \leq 2$. Thus, the ℓ_1-sensitivity is at most 2, and we can get ϵ-DP by adding $\mathrm{Lap}(2/\epsilon)$ noise to each entry of the J dimensional embedding (line 9). Post processing feature gives that removing bins with noisy counts less than threshold t (line 10) does not yield additional privacy leaks. □

3.1.2 Worst-case utility-privacy trade-off ($t = 0$ case)

We now analyze the worst case utility of Algorithm 1, i.e., the general case when we do not impose any assumptions on the distribution of the input dataset P.

Theorem 3 (Worst-case trade-off of Algorithm 1)**.** *Suppose that dataset P lies on an axes aligned hypercube of edge length R in $\mathbb{R}^d$. Let $\delta > 0$ be such that $\left(\frac{R}{w}\right)^d < \frac{\epsilon n}{4 \log \frac{1}{\delta}}$. Then Algorithm 1 outputs ϵ-DP dataset Q such that*

$$\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_Q(x) - \mathrm{KD}_P(x)| \leq \frac{2}{\frac{\epsilon n}{4J \log \frac{1}{\delta}} - 1} + \frac{w}{2}\sqrt{\frac{d}{e}},$$

with probability at least $1 - \delta$, where $J = \left(\frac{R}{w}\right)^d$.

Proof sketch. For a detailed proof see Section B of Supplementary Material. Theorem 2 guarantees that the output of Algorithm 1 is ϵ-DP. Let $P' := \{(c_1, v_1), \ldots, (c_J, v_J)\}$, where v, c are defined as in line 7 and 8 of Algorithm 1, respectively. We have the following sources of error.

- Rounding to the bin centers. We prove that $\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_{P'}(x) - \mathrm{KD}_P(x)| \leq \frac{w\sqrt{d}}{2\sqrt{e}}$.
- Adding noise and removing negatively weighted bins. We prove $\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_Q(x) - \mathrm{KD}_{P'}(x)| \leq \frac{8J \log \frac{1}{\delta}}{\epsilon n - 4J \log \frac{1}{\delta}}$ as a consequence of upper bound on J.

Triangle inequality over the sources of error completes the proof. □

[1]Here we impose a mild assumption that any two neighboring datasets will be in the same axes aligned hypercube.

For $t = 0$, Algorithm 1 suffers some obvious shortcomings. If data is well spread, there will be many bins with small number of points. After Laplace noise addition, corresponding point counts would often be negative and consequently the bins would be removed. On the other hand, with probability 0.5 empty bins would exhibit positive noisy point counts and would thus falsely be represented in the output. Both aspects hurt utility. A natural way to overcome these shortcomings is to increase the cut-off threshold. This approach is particularly well suited if we know that (most) non-empty bins are densely populated.

3.1.3 Beyond worst-case utility-privacy trade-off ($t > 0$ case)

For appropriate non-zero threshold, we present utility-privacy trade-off in the case of a general input, i.e. with no assumptions on input distribution.

Definition 4. *For $t > 0$, bins with noiseless count less (greater than or equal) than t will be called t-light (t-heavy).*

Theorem 5 (Beyond worst-case trade-off of Algorithm 1)**.** *Suppose that input dataset P lies on an axes aligned hypercube of edge length R in $\mathbb{R}^d$. Assume that $\delta > 0$ is such that $\left(\frac{R}{w}\right)^d \leq \frac{1}{\delta}$. For $t = \frac{8}{\epsilon}\log(1/\delta)$, let M and m be the total number of $t/2$-heavy bins and the total number of points in $3t/2$-light bins, respectively. Then, Algorithm 1 outputs ϵ-DP dataset Q such that*

$$\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_Q(x) - \mathrm{KD}_P(x)| \leq \frac{\epsilon m + 8M \log \frac{1}{\delta}}{\epsilon n - \epsilon m - 4M \log \frac{1}{\delta}} + \frac{m}{n} + \frac{w\sqrt{d}}{2\sqrt{e}},$$

with probability at least $1 - \delta$.

Proof sketch. For detailes see Section C of Supplementary Material. Theorem 2 guarantees that the output of Algorithm 1 is ϵ-DP. Let $P' := \{(c_1, v_1), \ldots, (c_J, v_J)\}$, where v, c are defined as in lines 7 and 8 of Algorithm 1. We have the following sources of error.

- Rounding to the bins centers. We prove that $\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_P(x) - \mathrm{KD}_{P'}(x)| \leq \frac{w\sqrt{d}}{2\sqrt{e}}$.
- With probability $1 - \delta/2$, all bins that are removed in filtering step are $3t/2$-light. Thus, there are at most m points that are filtered out which contributes to $\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_{P'}(x) - \mathrm{KD}_Q(x)|$ by at most $\frac{m}{n}$.
- With probability $1 - \delta/2$, all bins that survive filtering step are $t/2$-heavy. Before noise addition step, number of points in these bins is at least $n - m$ and at most n. As there are at most M such bins, the total noisy count in these bins is at least $n - m - \frac{4M}{\epsilon}\log(\frac{1}{\delta})$

and at most $n + \frac{4M}{\epsilon}\log(\frac{1}{\delta})$. This yields additional $\frac{\epsilon m + 8M\log(1/\delta)}{\epsilon n - \epsilon m - 4M\log(1/\delta)}$ term for the upper bound.

Union bound and triangle inequality over the sources of error complete the proof. □

The threshold t in Theorem 5 depends on privacy level ϵ, and so both M and m depend on ϵ. With no assumptions on distribution of P, it is not possible to provide meaningful bounds on M and m. We next study a special case.

3.1.4 Mixture of Gaussians input data

We analyze performance of Algorithm 1 for the special case of input data from multivariate Gaussian distribution. More precisely, we consider dataset P of n records in $\mathbb{R}^d$ with Gaussian distribution $\mathcal{N}(\mathbf{c}, \sigma^2 I)$, $\mathbf{c} \in \mathbb{R}^d$ i.e., from density $f(X = x) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{||x-\mathbf{c}||_2^2}{2\sigma^2}}$. This straightforwardly generalizes to a mixture of multivariate Gaussians.

Theorem 6 (Gaussian trade-off using Algorithm 1). *Suppose that input dataset P lies on an axes aligned hypercube of edge length R in $\mathbb{R}^d$. If $n \geq \left(\frac{w}{\sigma\sqrt{2\pi}}\right)^d$, for $\delta > 0$ such that $\frac{n}{(\log n)^{d/2}} \geq 16 \cdot \log\frac{1}{\delta} \cdot (\frac{12\sigma}{w})^2$ and threshold $t = \frac{8}{\epsilon}\log(1/\delta)$, Algorithm 1 outputs ϵ-DP dataset Q s.t.*

$$\sup_{x \in \mathbb{R}^d} |\mathrm{KD}_Q(x) - \mathrm{KD}_P(x)| \leq \frac{8(\log\frac{1}{\delta})^{1/3} \cdot e^{-\frac{d}{3}(\log\frac{w}{\sigma\sqrt{2\pi}} - 2)}}{(\epsilon n)^{1/3}} + \frac{16\log\frac{1}{\delta} \cdot (\frac{12\sigma}{w})^d (\log n)^{d/2}}{\epsilon n} + \frac{w\sqrt{d}}{2\sqrt{e}},$$

with probability at least $1 - \delta$.

Proof sketch. For a detailed proof see Section D of Supplementary Material. This is a special case of Theorem 5. The Gaussian distribution assumption enables us to provide upper bounds on the number of $t/2$-heavy bins m, and total number of points in $3t/2$-light bins M, for $t = \frac{8}{\epsilon}\log(1/\delta)$. Loosely speaking, in this case majority of points lives within densly populated areas, and so there are neither too many points in the light bins nor the number of heavy bins is too large. □

In Section 4 we compare empirical MMD to the bound $(O(\epsilon n)^{-1/3})$ from Theorem 6. Results suggest that lower bound is smaller which is in line with results from Duchi et al. [2013] where the gap is of order $O(n^{-1/2})$. Their setting is different as they study local differential privacy in the context of distribution estimation (not synthetic data release). We also highlight recent work of Kamath et al. [2018] on DP learning parameters of multivariate Gaussian distribution. Their approach learns parameters and is thus not directly comparable as we output a dataset.

3.2 DATA DEPENDENT ALGORITHM

In the general case, data independent approaches suffer from the curse of dimensionality. The reason is that as opposed to traditional applications of hash functions, we need to keep track of empty bins in order to treat them similarly to non empty ones, as a bin that is empty with respect to P is not necessarily empty with respect to a neighbouring dataset. In high dimensions, this makes the data independent binning impractical, as there are typically many empty bins. Moreover, in densely populated areas finer grid would incur smaller error due to the rounding to the centers, and thus yield a higher utility.

We propose differentially private algorithm based on *adaptive binning* i.e., recursive partitioning of the space. Before we proceed, we introduce notion of a *decision tree*. We assume arbitrary but fixed enumeration of d dimensions denoted by i where $i \in [d]$. The root of the tree is characterized by the initial dataset P, the center c and the the edge R of the smallest axis aligned cube that contains whole P, and axes 0 to split along. If decision at the root is to proceed with recursion (we discuss decision making below), we proceed as follows. Initial recursion splits the dataset along axes 0 and divides the corresponding edge of the cube in two equal $R/2$ parts which results in the creation of two *children* nodes. Each of them is characterized by a fraction of the dataset that ended up in the corresponding part, center and the radius of the new cube that contains that fraction of the dataset, and new axes to cut along. The new axes to cut along is always previous axes $+1$ i.e., the next axes in ordering $[d]$. The recursion proceeds on the newly created nodes, subject to a positive decision on whether to recurse further. Nodes on which recursion does not proceed do not have any children and represent *leaves*.

Note that due to the data dependent aspect, we need to ensure that adaptive binning is differentially private. Thus, each decision on whether to recurse or not has to be based on noisy point counts. By composition property of DP, if there are l levels of data dependent decisions in the decision tree, one needs to guarantee ϵ'/l-DP for each recursion, so that once binning algorithm terminates we achieve ϵ'-DP. Note that the output of adaptive binning is partitioning of the space, not the synthetic dataset, and thus once the binning is done we are in the setting from the beginning of Algorithm 1. That is, in order to release differentially private synthetic dataset, we have to obtain noisy versions of the point counts bin by bin and output bins that pass certain threshold. By ensuring a ϵ''-DP for this part of the procedure, one will achieve $(\epsilon' + \epsilon'')$-DP guarantee for the whole algorithm by the composition property of DP. Before we formally introduce adaptive binning, we discuss some desirable settings.

Avoiding large bins In high dimensions, we will frequently observe large bins where recursion stops due to small noisy point counts. This would however yield large error due to rounding to the center, since the bin has large edge lengths. Thus it would be beneficial to have the algorithm run bin splitting *independently* of data for a few rounds, e.g., until it reaches a maximum edge length for all bins below some threshold s_1. After that, the algorithm would run in data dependent regime. Note that the depth of data independent part in this setting is $h = d\log_2(R/s_1)$, and up until that level there are no privacy leaks.

Avoiding decision trees with large depth Privacy cost of adaptive binning is determined by the number of data dependent levels in decision tree. Thus, even if noisy point counts are large, it might be beneficial to stop the recursion once the number of data dependent levels passes certain threshold. Equivalently, we stop the recursion over a bin when its maximal edge length is below certain threshold s_2, regardless of the value of the noisy point count. In this setting, the maximum tree depth is $h' = d\log_2(R/s_2)$, and having in mind above discussion, number of data dependent levels is at most $h' - h = \log_2(s_1/s_2)$.

Algorithm 2 formally describes our adaptive binning. More precisely, it identifies the root of the tree as discussed above, and passes it to Algorithm 3 (discussed below). Lines 3 - 4 identify the boundaries of the dataset along each of d dimensions, line 5 identifies the center of the cube, and line 6 the edge of the cube that contains the whole dataset. Axes for next split is set to 0, and the root node is passed to Algorithm 3 . Finally, Algorithm 3 implements differentially private recursive binning. It takes as an input a node, total privacy budget (until the recursion stops), threshold for the noisy point counts, and the maximum and minimum allowed edge lengths of final bins. The output of the algorithm is the set of bins. According to previous discussion, the algorithm will recurse if either noisy point count is larger than the threshold, or the bin's largest edge is too large (line 5), with the exception of that if the largest edge is too small (line 7), recursion stops regardless of the value of the noisy point count. If recursion proceeds, the current node is split in the two and recursion proceeds on each of them.

Algorithm 2 Adaptive binning

```
1: procedure ADAPTIVE-BINNING($P, \epsilon', \tau$)
2:                    ▷ dataset $P$, DP budget $\epsilon'$,cut off level $\tau$
3:     For any $i \in [d]$, low$_i \leftarrow \min_{p\in P}(p_i)$
4:     For any $i \in [d]$, high$_i \leftarrow \max_{p\in P}(p_i)$
5:     For any $i \in [d]$, c$_i \leftarrow \frac{\text{low}_i+\text{high}_i}{2}$
6:     $w \leftarrow \max_i(\text{high}_i - \text{low}_i)$
7:     curr-axis $\leftarrow 0$        ▷ axis the bin will be cut along
8:     node $\leftarrow$ NODE($P, c, w$, curr-axis)
9:     RECURSIVE-BINNING(node, $\epsilon', \tau$)
```

Algorithm 3 Recursive binning

```
1: procedure RECURSIVE-BINNING(NODE, $\epsilon', \tau, s_1, s_2$)
2:         ▷ Node to recurse on, DP budget $\epsilon'$, threshold $\tau$
3:     ▷ max and min edge length for final bins $s_1$ and $s_2$
4:     $P, c, w$, curr-axis $\leftarrow$ node
5:     if $|P| \leq \tau + \text{LAP}(2(h'-h)/\epsilon')$ and the bin's largest
       edge length $\leq s_1$ then return
6:     ▷ $h' - h = d\log_2(s_1/s_2)$ is the max depth of data
       dependent part of the tree
7:     if the bin's largest edge length $< s_2$ then return
8:     $P_\ell, P_r \leftarrow \emptyset, \emptyset$
9:     for $p \in P$ do
10:        if $p_{curr-axis} \leq c_{\text{curr-axis}}$ then $P_\ell \leftarrow P_\ell \cup \{p\}$
11:        else $P_r \leftarrow P_r \cup \{p\}$
12:    $c_\ell, c_r \leftarrow c, c$
13:    $c_{\ell,curr-axis} \leftarrow c_{curr-axis} - w/4$
14:    $c_{r,curr-axis} \leftarrow c_{curr-axis} + w/4$ ▷ coordinates of
       newly formed bins centers
15:    if curr-axis $= d-1$ then $w \leftarrow w/2$
16:    node.left $\leftarrow$ NODE($P_\ell, c_\ell, w$, (curr-axis $+ 1)\%d$)
17:    RECURSIVE-BINNING(node.left, $\epsilon', \tau$)
18:    node.right $\leftarrow$ NODE($P_r, c_r, w$, (curr-axis $+ 1)\%d$)
19:    RECURSIVE-BINNING(node.right, $\epsilon', \tau$)
```

Theorem 7. *For any dataset P, $\epsilon' > 0$, $\tau > 0$, $s_1, s_2 > 0$, Algorithm 2 returns a tree such that the set of bins determined by its leaves is ϵ'-differentially private.*

Proof. Data dependent part of the decision tree for this algorithm has maximum depth $h' - h = d\log_2(s_1/s_2)$. Each neighboring dataset affects only one root to leaf path, and thus it is enough to consider maximal privacy loss incurred along a single path to the leaf. Also, since the ℓ_1 sensitivity of the point counts is 1, it suffices to add $\text{LAP}(2(h'-h)/\epsilon')$ noise to each decision condition in order to guarantee ϵ'-differential privacy for the entire path. Two neighboring datasets differ in the point counts in at most two paths. □

Theorem 8 (DP of Data Dependent algorithm)**.** *For a dataset P and $\epsilon' > 0$, let $c_1, \ldots, c_J \in \mathbb{R}^d$ denote centers of bins corresponding to the leaves of tree output by Algorithm 2, and $\mathbf{v} \in \mathbb{R}^J$ be the vector of corresponding point counts in each bin. If $c_1, \ldots, c_k$ and $\mathbf{v}$ are passed to line 9 of Algorithm 1 with $\epsilon'' > 0$, then the final output of Algorithm 1 is $\epsilon' + \epsilon''$-DP dataset Q.*

Proof. Consequence of composition property of DP. □

3.2.1 Implicit sampling of empty bins

As already discussed, number of empty bins grows exponentially in dimension and this represents a challenge as we need to treat empty bins in a same manner as non empty ones for the purpose of achieving DP. Thus, efficient implementation of our algorithms would avoid storing all bins and iterating through them for the purpose of noise addition and filtering. We utilize idea exploited in Cormode et al. [2011], to implicitly implement the noise addition and filtering on empty bins. This benefits both data independent Algorithm 1 (see Section E.1 of Supplementary Material) and data dependent Algorithm 2 (discussed below).

Lemma 9. *Explicit implementation of noise addition and filtering on empty bins as per Algorithm 4 is equivalent to the implicit implementation provided in Algorithm 5 .*

See Section F of Supplementary Material for proof. In Algorithm 5, $\textsc{ConditionalLap}(2/\epsilon, t)$ denotes a random variable with Laplace distribution conditioned on being greater than or equal to t.

Algorithm 4 Explicit implementation

```
1: for any empty bin do
2:     $\eta \leftarrow \textsc{Lap}(2/\epsilon)$
3:     if $\textsc{PointCount}(\text{bin}) + \eta = 0 + \eta \geq t$ then
4:         add this bin center with weight $\eta$ to the output
```

Algorithm 5 Implicit implementation

```
1: $K \leftarrow$ the number of empty bins
2: $p \leftarrow \Pr[\textsc{Lap}(2/\epsilon) \geq t]$
3: $m \leftarrow \textsc{Binom}(K, p)$
4: sample $m$ empty bins out of $K$ empty bins without
   replacement
5: for each sampled empty bin do
6:     $\eta \leftarrow$ a sample of $\textsc{ConditionalLap}(2/\epsilon, t)$
7:     add this bin center with weight $\eta$ to the output
```

3.2.2 Implicit sampling in the data dependent Algorithm 2

The union of the set of empty and non empty bins given as output of Algorithm 2 coincides with the set of leaves in the tree of recursion decisions. The question is, whether from the information on the number of data independent levels h, and paths from the root to the leaves corresponding to non empty bins, one can recover the remaining set of leaves i.e., those corresponding to the empty bins. In this section, we prove that this is possible for a large enough cut off threshold τ (see line 5 in Algorithm 3) which guarantees that with high probability the algorithm does not partition empty bins further in the data dependent part of the algorithm. See Figure 2 for an illustration of a decision tree, and Section F of Supplementary Material for proof of Lemma 10.

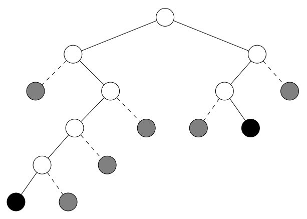


Figure 2: Tree with $h = 2$ and $h' = 5$. Black nodes are non-empty bins, gray nodes are empty bins we need to sample.

Lemma 10. *Let h' and h denote total depth and the depth of data independent part of the tree, respectively. If the threshold τ in line 5 of Algorithm 3 is set to be greater than*

$$\frac{2(h'-h)}{\epsilon'} \log\left(\frac{1}{\delta}\cdot\left(2^h + n(h'-h)\right)\right),$$

then with probability $1-\delta$ the adaptive binning will not divide any empty bin.

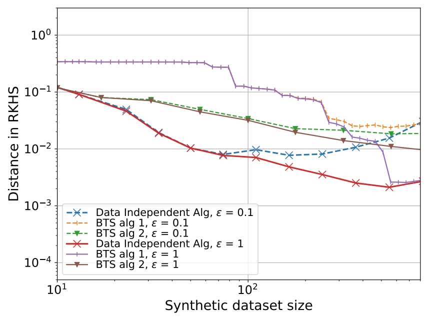
Theorem 11. *If we pick threshold τ in line 5 of Algorithm 3 as per Lemma 10, then storing information on the number of data independent levels h and paths to leaf nodes that represent non empty bins, enables implicit sampling of empty bins as per Algorithm 5.*

Proof. Lemma 10 guarantees that no empty bin is recursed on. Thus, each parent node has at most one child corresponding to an empty bin. In particular, for each two non-empty bins (black nodes in Figure 2) we can identify their common ancestor and the number of empty bins (gray nodes in Figure 2) between them. Thus, set of empty bins can be recovered from the encoding of non empty bins. See Section F of Supplementary Material for details. □

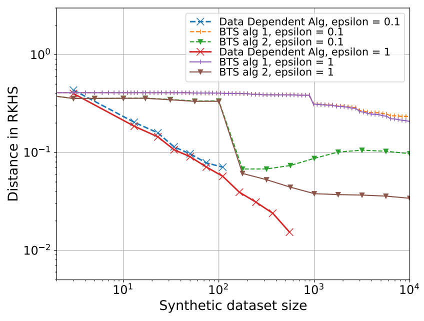
4 EXPERIMENTS

We provide experimental results on both of our proposed algorithms in various settings.

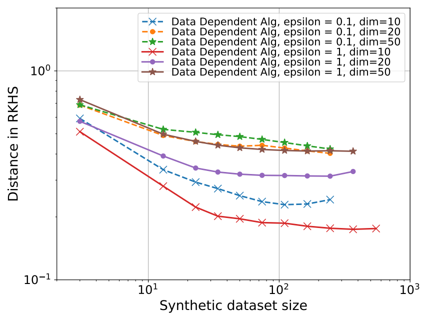
Privacy-utility trade-off First, we show improvements using our data independent Algorithm 1 over both algorithms of Balog et al. [2018] on a mixture of Gaussians dataset in dimension 2. The comparison is presented in Figure 3a, where a considerable improvement is achieved, in all regimes of privacy budget ϵ and dataset size. For a 5-dimensional mixture of Gaussians dataset, although our data independent algorithm is unable to outperform the algorithms of Balog et al. [2018], our data dependent algorithm overcomes the curse of dimensionality and it achieves lower error rates using smaller number of synthetic data points. This improvement for all regimes of privacy levels



(a)



(b)



(c)

Figure 3: (Left) Algorithm 1 vs Balog et al. [2018] in dimension 2. (Middle) Our Data Dependent Algorithm vs Balog et al. [2018] in dimension 5. (Right) Data Dependent Algorithm in high dimensions.

is presented in Figure 3b. For both settings, we use datasets utilized in Balog et al. [2018] (see Section G.1 of Supplementary Material for details). Figure 3c confirms that our data dependent algorithm achieves low errors for high dimensions and various privacy regimes, and thus overcomes curse of dimensionality.

Tightness of bound from Theorem 6 Theorem 6 provides the upper bound on utility loss of Algorithm 1 for samples from multivariate Gaussian. When bins are such that the error $\frac{w\sqrt{d}}{2\sqrt{e}}$ arising from rounding to centers is negligible, for $\epsilon \gg \frac{1}{n}$ the bound scales as $O\left((\epsilon n)^{-1/3}\right)$. Figure 4 shows empirical MMD for various privacy levels vs. $O\left((\epsilon n)^{-1/3}\right)$ benchmark and suggests that there might be a gap and our bound can be improved. See Section 5 for further discussion.

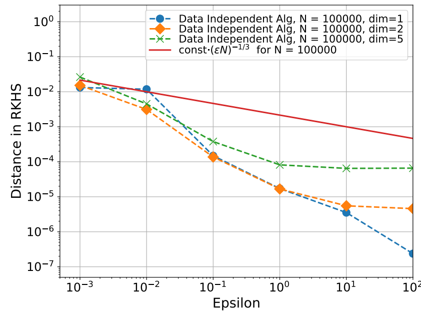


Figure 4: Utility loss of Algorithm 1 for various privacy budgets, and input of 100, 000 samples from standard Gaussian for various dimensions. It stabilizes regardless of privacy budget due the error of rounding to centers.

Classification task on real data We evaluate performance of our data dependent algorithm on downstream binary classification task on a real tabular dataset. We rely on credit card fraud detection dataset used in Harder et al. [2021] and compare our data dependent algorithm with their DP-MERF. We follow their experimental setup to train 12 classifiers on synthetic data and evaluate their performance on original data. See Section G.4 of Supplementary Material for more detailed setup and results. Our algorithm does not outperform DP-MERF in terms of ROC values which is not surprising as we do not rely on deep generative models. However, our performance degrades slower as privacy increases (Figure 5).

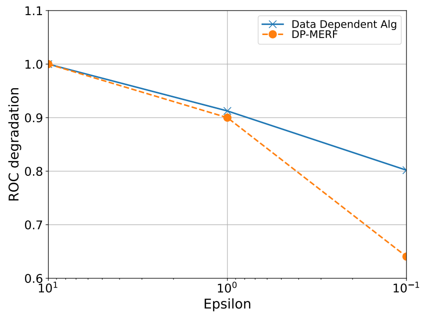


Figure 5: Comparison of our Data Dependent Algorithm and DP-MERF on downstream classification task. ROC degradation is represented as a ratio of ROC corresponding to a specified ϵ budget and ROC for $\epsilon = 10$.

Optimal size of bins We explore what combination of widths and weights of bins minimizes MMD between synthetic data and the sample (regime with no privacy). For 1-dimensional standard Gaussian input, we consider synthetic data given by a mixture of uniforms. More precisely, synthetic data consists of centers of $2k + 1$ bins of a given width, symmetrically arranged around the mean. Figure 6 shows KL and MMD between standard Gaussian and mixture of uniforms [Rustamov, 2021], as well as sample MMD for a Gaussian sample of size 100, 000. For appropriately set width, sample MMD is getting small which indicates that binning approach has potential to yield good utilities. For more details and derivation of optimal widths and weights see Section H of Supplementary Material.

For additional experiments, see Sections G.2 and G.3 of Supplementary Material.

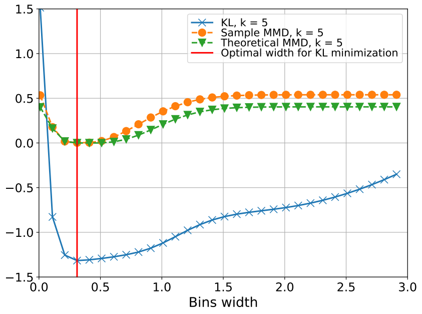


Figure 6: 1-dimensional standard Gaussian vs 11-mixture of uniforms and various widths.

5 CONCLUSIONS AND FUTURE WORK

We proposed an interpretable and efficient algorithms for differentially private synthetic data generation, where data takes multidimensional numerical form. We provide theoretical bounds in various settings and empirically validate our algorithm performance in terms of both simulated and real-world datasets.

Our algorithms can be extended to tabular data characterized by mixed numerical and categorical columns, as each category within a column corresponds to a bin along that dimension. However, this setting does not benefit from our theoretical guarantees on the utility (e.g. Theorem 3) which we measure in terms of kernel density metrics defined on $\mathbb{R}^d$. Moreover, our data dependent approach does not handle the curse of dimensionality problem as the total number of bins grows exponentially with the number of categorical columns. This is due to the fact that in mixed numerical and categorical settings there is no straightforward way to merge or split categories depending on point counts, as it is the case with bins in $\mathbb{R}^d$. It is an interesting future direction to combine our methods for numerical columns with a different method along categorical columns in order to achieve scalable solution in such setting.

Another interesting direction is to resolve the question of the theoretical utility-privacy guarantees for our data dependent algorithm. This is challenging in the data dependent setting as it is not easy to provide meaningful bounds on the total number of bins. Furthermore, we conjecture that one can improve upon our bounds using more sophisticated LSH functions, e.g., projection based LSHs such as Datar et al. [2004], Andoni and Indyk [2006]. This direction needs novel techniques in order to handle empty bins. For the special case of Gaussian multivariate distribution, we leave it as future work to compare our algorithms to existing results on DP, e.g., Kamath et al. [2018] who provide DP learning of distribution parameters. Lower bounds for our space-partitioning approach also remain open, in particular whether our bound from Theorem 6 can be improved to be of the form $O(n^{-1/2})$ as suggested by results of Duchi et al. [2013].

References

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 459–468. IEEE Computer Society, 2006. doi: 10.1109/FOCS.2006.49. URL `https://doi.org/10.1109/FOCS.2006.49`.

Matej Balog, Ilya O. Tolstikhin, and Bernhard Schölkopf. Differentially private database release via kernel mean embeddings. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 423–431. PMLR, 2018. URL `http://proceedings.mlr.press/v80/balog18a.html`.

Graham Cormode, Magda Procopiuc, Divesh Srivastava, and Thanh T. L. Tran. Differentially private publication of sparse data, 2011.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Jack Snoeyink and Jean-Daniel Boissonnat, editors, *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*, pages 253–262. ACM, 2004. doi: 10.1145/997817.997857. URL `https://doi.org/10.1145/997817.997857`.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and minimax bounds: Sharp rates for probability estimation, 2013. URL https://arxiv.org/abs/1305.6000.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Jerome Friedman, Jon Bentley, and Raphael Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3:209–226, 09 1977. doi: 10.1145/355744.355745.

Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In Gurpreet Dhillon, Fredrik Karlsson, Karin Hedström, and André Zúquete, editors, *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*, volume 562 of *IFIP Advances in Information and Communication Technology*, pages 151–164. Springer, 2019. doi: 10.1007/978-3-030-22312-0_11. URL https://doi.org/10.1007/978-3-030-22312-0_11.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Frederik Harder, Kamil Adamczewski, and Mijung Park. DP-MERF: differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1819–1827. PMLR, 2021. URL http://proceedings.mlr.press/v130/harder21a.html.

Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613. ACM, 1998. doi: 10.1145/276698.276876. URL https://doi.org/10.1145/276698.276876.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: generating synthetic data with differential privacy guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1zk9iRqF7.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. Synthetic data – what, why and how?, 2022.

Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan R. Ullman. Privately learning high-dimensional distributions. *CoRR*, abs/1805.00216, 2018. URL http://arxiv.org/abs/1805.00216.

Noman Mohammed, Rui Chen, Benjamin C. M. Fung, and Philip S. Yu. Differentially private data release for data mining. In Chid Apté, Joydeep Ghosh, and Padhraic Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 493–501. ACM, 2011. doi: 10.1145/2020408.2020487. URL https://doi.org/10.1145/2020408.2020487.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2016. URL https://arxiv.org/abs/1610.05755.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate, 2018. URL https://arxiv.org/abs/1802.08908.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11(10):1071–1083, 2018. doi: 10.14778/3231751.3231757. URL http://www.vldb.org/pvldb/vol11/p1071-park.pdf.

Jeff M. Phillips and Wai Ming Tai. Near-optimal coresets of kernel density estimates. *Discret. Comput. Geom.*, 63(4):867–887, 2020. doi: 10.1007/s00454-019-00134-6. URL https://doi.org/10.1007/s00454-019-00134-6.

Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. Differentially private grids for geospatial data. In Christian S. Jensen, Christopher M. Jermaine, and Xiaofang Zhou, editors, *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 757–768. IEEE Computer Society, 2013. doi: 10.1109/ICDE.2013.6544872. URL https://doi.org/10.1109/ICDE.2013.6544872.

Parikshit Ram and Alexander Gray. Which space partitioning tree to use for search? *Advances in Neural Information Processing Systems*, 26, 2013.

Raif M. Rustamov. Closed-form expressions for maximum mean discrepancy with applications to wasserstein auto-encoders. *Stat*, 10(1):e329, 2021. doi: https://doi.org/10.1002/sta4.329. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sta4.329. e329 sta4.329.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-CGAN: differentially private synthetic data and label generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 98–104. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPRW.2019.00018. URL http://openaccess.thecvf.com/content_CVPRW_2019/html/CV-COPS/Torkzadehmahani_DP-CGAN_Differentially_Private_Synthetic_Data_and_Label_Generation_CVPRW_2019_paper.html.

Yonghui Xiao, Li Xiong, Liyue Fan, Slawomir Goryczka, and Haoran Li. Dpcube: Differentially private histogram release through multidimensional partitioning. *Trans. Data Priv.*, 7(3):195–222, 2014. URL http://www.tdp.cat/issues11/abs.a136a13.php.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018. URL http://arxiv.org/abs/1802.06739.

Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In Fatma Özcan, Georgia Koutrika, and Sam Madden, editors, *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 155–170. ACM, 2016. doi: 10.1145/2882903.2882928. URL https://doi.org/10.1145/2882903.2882928.

Tianqing Zhu, Gang Li, Wanlei Zhou, and Philip S. Yu. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, 2017. doi: 10.1109/TKDE.2017.2697856.