THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MATHEMATICS

ESTIMATING NOISY HISTOGRAMS WITH QUALITY GUARANTEES

GAVIN KERRIGAN
SPRING 2019

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Mathematics
with honors in Mathematics

Reviewed and approved* by the following:

Daniel Kifer
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Sergei Tabachnikov
Professor of Mathematics
Honors Adviser

*Signatures are on file in the Schreyer Honors College.

Abstract

The ability to disseminate useful statistics generated from sensitive data while still preserving the privacy of its contributors is an open challenge with highly desirable consequences. In this thesis, we investigate the problem of histogram estimation within the framework of differential privacy. Given a histogram with noisy counts, we seek to generate an estimate of the original, non-noisy histogram such that our estimate is expected to improve upon the identity estimator in terms of sum-squared-error. As a stepping stone towards this goal, this thesis investigates several possible choices of such an estimator as a baseline for comparison.

Table of Contents

Acknowledgements

I would like to thank Professor Daniel Kifer for his guidance and patience throughout the creation of this thesis.

Chapter 1

Introduction to Differential Privacy

In today's data-driven world, the ability to release useful information about a database while still hiding the content of said database is crucial. For example, a medical research scientist may want to publish the results of their study in which they statistically analyze medical records from HIV-positive patients. If an agent were able to use the published information to discover the presence of a certain individual in the database, such a release would constitute a violation of privacy. Naive methods, such as releasing only summary statistics, are vulnerable to attacks and do not provide privacy guarantees.

With this intuition in mind, Dalenius proposed a desired requirement of statistical databases. In essence, what was sought was a guarantee that no information is revealed about an individual record in a database when releasing information about the database as a whole [1]. However, Dwork proves in [2] that such a guarantee is impossible if one takes into account the necessity of releasing statistics which are useful. That is, outside of trivial cases (such as simply releasing no information), Dalenius' proposed requirement is unfeasible.

As such, there was a desire to quantify the degree to which the privacy of individual records of a database could be preserved. First developed by Dwork [2], differential privacy is a framework which allows one to measure the risk one takes on by contributing to a database. Several relaxations of this definition exist, such as concentrated differential privacy [3] and Rényi differential privacy [4]. These relaxations allow one to derive better analyses of the privacy guarantees afforded [4].

As histograms are a basic tool in data visualization and summary, there is much interest in releasing high-quality histograms which are differentially private. In this thesis, we investigate methods by which we may estimate an unknown histogram given both a noisy version of said histogram and the distribution followed by the added noise. In particular, we seek to develop an estimator which is permutation invariant and outperforms the identity estimator in terms of expected sum-squared-error.

1.1 Background

We begin by recalling a few key definitions and results. In the following sections, let $\mathcal{D}$ represent the space of databases in our domain.

1.1.1 Differential Privacy

Definition 1 (ϵ-differential privacy [2])**.** *A randomized function $f : \mathcal{D} \to \mathcal{R}$ satisfies ϵ-differential privacy if for all databases $D_1, D_2 \in \mathcal{D}$ differing by at most one element and for all $S \subset \mathcal{R}$,*

$$P(f(D_1) \in S) \leq e^{\epsilon} P(f(D_2) \in S)$$

The following definition of (ϵ, δ)-differential privacy is a relaxation of ϵ-differential privacy. Informally, this definition captures the condition of satisfying standard ϵ-differential privacy with probability $1 - \delta$.

Definition 2 ((ϵ, δ)-differential privacy [5])**.** *A randomized function $f : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for all databases $D_1, D_2 \in \mathcal{D}$ differing by at most one element and for all $S \subset \mathcal{R}$,*

$$P\left(f(D_1) \in S\right) \leq e^{\epsilon} P(f(D_2) \in S) + \delta$$

The key differences between ϵ-differential privacy and (ϵ, δ)-differential privacy are discussed further in [4].

1.1.2 The Laplace and Gaussian Mechanisms

These definitions are not only intuitively sound, but also is easily satisfiable. When the information released is real-valued (i.e. the outputs of our function are elements of $\mathbb{R}^n$), a common technique which allows for privacy guarantees is the addition of random noise. In particular, the Laplace mechanism can be used to satisfy ϵ-differential privacy, and the Gaussian mechanism can be used to satisfy (ϵ, δ)-differential privacy. In defining these mechanisms, we will make use of the concept of the ℓ_p sensitivity of a function.

Definition 3 (ℓ_p sensitivity [4])**.** *Let $\mathcal{N} \subset \mathcal{D} \times \mathcal{D}$ be the set of all pairs of databases in $\mathcal{D}$ differing by at most one element. The ℓ_p sensitivity of a function $f : \mathcal{D} \to \mathcal{R}$ is defined as*

$$\Delta_p(f) = \max_{(D_1, D_2) \in \mathcal{N}} ||f(D_1) - f(D_2)||_p$$

Given a fixed privacy budget of ϵ, a simple method of guaranteeing that our release is ϵ-differentially private is the Laplace($1/\epsilon$) mechanism. Given a query function $f : \mathcal{D} \to \mathbb{R}^n$, ϵ-differential privacy may be attained by the addition of independently drawn zero-mean Laplace distributed noise with scale $\Delta_1(f)/\epsilon$ to each component of f. Explicitly, to each component of f we add independent noise with PDF

$$\text{Laplace}(1/\epsilon)(x) = \frac{\epsilon}{2} \exp(-\frac{\epsilon|x|}{\Delta_1(f)})$$

A similar technique is the Gaussian mechanism. It is shown in [4] that the addition of Gaussian noise can not satisfy pure ϵ-differential privacy for any value of ϵ. However, for any $\epsilon \in (0, 1)$ and any δ, the addition of zero-mean Gaussian noise with variance $\sigma > \frac{\Delta_2(f)}{\epsilon} \sqrt{2 \ln(1.25/\delta)}$ to our function will achieve (ϵ, δ)-differential privacy [6].

Chapter 2

Problem Description

In this thesis, we consider an application of differential privacy to histograms. An n-bin histogram is formally defined as a vector $H = (a_1, a_2, \ldots a_n) \in \mathbb{Z}^n_{\geq 0}$. The counts a_i are known only to the individual who owns the histogram's original data. The data owner then generates n independent Laplace$(1/\epsilon)$ random variables $\eta_1, \eta_2, \ldots, \eta_n$. Then, the data owner publicly releases a noisy version of the original histogram, denoted by $\widetilde{H}$, whose counts are $\widetilde{H} = (a_1 + \eta_1, a_2 + \eta_2, \ldots, a_n + \eta_n)$. Notably, this release satisfies ϵ-differential privacy. In summary, the only publicly available information is the noisy histogram $\widetilde{H}$ and the scale/distribution of the added noise.

An individual who wants to use the information contained in H would then want to produce an estimate of the original histogram from only the publicly known information. We represent such an estimate by $g(\widetilde{H}, \epsilon)$. A naive choice for such a function is the identity estimator, for which the user simply accepts $\widetilde{H}$ as an estimate for H.

Our goal is to develop a function $g : \mathcal{D} \to \mathbb{R}^n$ which is expected to improve over the identity function in terms of sum-squared-error (SSE) for all histograms. More precisely, a estimator g is said to be better than an estimator f if

$$\mathbb{E}\left(||g(\widetilde{H}) - H||_2^2\right) \leq \mathbb{E}\left(||f(\widetilde{H}) - H||_2^2\right) \qquad \forall H \in \mathcal{D}$$

2.1 Developing a Baseline

To design such an estimator, we begin by considering the case of a two-dimensional histogram. As a baseline, we would like to compare the expected SSE values of several different estimators. Below, we illustrate a few existing options.

2.1.1 Boundary Inflated Estimator

Since the histogram H is known to contain non-negative counts, one possible choice for an estimator given $\widetilde{H}$ is the *Boundary Inflated* estimator, which returns $\widetilde{H}$ where any negative counts are set to zero. This is denoted by $\widetilde{H}_+ = \max(0, \widetilde{H})$. For example, the histogram $[5, 5, -1, 12, -8]$ would become $[5, 5, 0, 12, 0]$. However, a potential drawback of this estimator is that it is biased in the sense that the total number of counts in the histogram is not preserved.

2.1.2 Resized Estimator

If we desire that an estimator preserve the total number of counts in the given noisy histogram, one possible estimator is the *Resized* estimator. Given a noisy histogram $\widetilde{H}$, the resized estimator will return $\max(0, \widetilde{H})$ where the non-negative counts have been uniformly docked such that the total count of the resulting histogram is equal to that of $\widetilde{H}$. Following the above example with $\widetilde{H} = [5, 5, -1, 12, -8]$, the resized estimate of H would be $[2, 2, 0, 9, 0]$.

2.1.3 NoiseFirst

Xu et al. develop the NoiseFirst algorithm for releasing differentially private histograms in [7]. In this section, we describe the key concepts of NoiseFirst for the sake of completeness. Given a

noisy histogram $\widetilde{H}$ with n independtly Laplace($1/\epsilon$) perturbed counts, NoiseFirst begins by computing the optimal k-bin histogram structure for $k \in \{1, 2, \ldots, n\}$ via the dynamic programming procedure described in [8]. This optimal k-bin histogram is denoted $\hat{H}_k^*$. In particular, the counts of $\hat{H}_k^*$ are comprised of the mean counts of the subsumed bins of $\widetilde{H}$. Then, the optimal number of bins k^* is chosen via the objective

$$k^* = \arg\min_k \mathbb{E}\left(||\hat{H}_k^* - H||_2^2 - \frac{2n - 4k}{\epsilon^2}\right)$$

The NoiseFirst algorithm then returns $\hat{H}_{k^*}^*$ as an estimate for the original histogram H.

Chapter 3

Searching for Counterexamples

3.1 Analyzing the Identity Estimator

In this section we analyze the SSE of the identity estimator.

Lemma 1. *Let $H \in \mathbb{R}^n$ be a histogram and let $\eta = (\eta_1, \eta_2, \ldots, \eta_n)$ be a random vector with each η_i i.i.d., following the Laplace($1/\epsilon$) distribution. Define $\widetilde{H} = H + \eta$. We then have that the identity estimator has expected SSE*

$$\mathbb{E}(||\widetilde{H} - H||_2^2) = \frac{2n}{\epsilon^2}$$

Proof. It is well known that the Laplace distribution with scale parameter b has variance $2b^2$. Using the linearity of the expectation,

$$\begin{aligned}
\mathbb{E}(||\widetilde{H} - H||_2^2) &= \mathbb{E}(||(H + \eta) - H||_2^2) \\
&= \mathbb{E}(||\eta||_2^2) \\
&= \sum_i^n \mathbb{E}(\eta_i^2) \\
&= 2n \left(\frac{1}{\epsilon}\right)^2
\end{aligned}$$

□

3.2 Searching for a Counterexample: NoiseFirst vs Identity

In this section, we search for a counterexample which demonstrates that the NoiseFirst algorithm is not expected to improve over the identity estimator for all histograms. In particular, we want to find a histogram H and a value of ϵ such that the estimator produced by the NoiseFirst algorithm is expected to perform worse than the identity estimator in terms of SSE. To do so, we implement the NoiseFirst algorithm for the two-dimensional case and perform various numerical experiments.

3.2.1 Two-Dimensional NoiseFirst

Algorithm 1 describes the pseudocode for the NoiseFirst algorithm in the two-dimensional case. Lines $1-2$ describe the construction of the optimal one-bin and two-bin histograms. Lines $4-7$ describe the steps for choosing the optimal bin number, as described in the ComputeOptimalK algorithm (Algorithm 2) of [7].

Algorithm 1 Two-Dimensional NoiseFirst

Input: A noisy 2D histogram $\widetilde{H} = (\widetilde{h}_1, \widetilde{h}_2)$, a privacy parameter ϵ

$\hat{H}_1^* \leftarrow \left(\frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2), \frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2)\right)$
$\hat{H}_2^* \leftarrow \widetilde{H}$

if $\left|\widetilde{h}_1 - \widetilde{h}_2\right| \leq \frac{2\sqrt{2}}{\epsilon}$ **then** ▷ $2\frac{\sqrt{2}}{\epsilon} = n\sigma$, where $\sigma = \text{StdDev}(\text{Laplace}(1/\epsilon))$
    return $\hat{H}_1^*$
else
    return $\hat{H}_2^*$

Algorithm 1 Derivation

We now describe the derivation of Algorithm 1. The construction of the optimal one-bin histogram and two-bin histogram is straightforward, as there is only one choice for each. Namely, for $\widetilde{H} = \left(\widetilde{h}_1, \widetilde{h}_2\right)$,

$$\hat{H}_1^* = \left(\frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2), \frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2)\right)$$
$$\hat{H}_2^* = \widetilde{H}$$

We note that the analyses in [7] give us

$$\mathbb{E}(SSE(\hat{H}_k^*, \widetilde{H})) = SSE(H_k, H) + \frac{2(n-k)}{\epsilon^2} \tag{3.1}$$

$$\mathbb{E}(SSE(\hat{H}_k^*, H)) = SSE(H_k, H) + \frac{2k}{\epsilon^2} \tag{3.2}$$

We now need to determine the optimal bin count k^*. By optimal, we mean that the choice of k^* minimizes the expected SSE between $\hat{H}_{k^*}^*$ and H. That is,

$$\begin{aligned}
k^* &= \arg\min_k \mathbb{E}(SSE(\hat{H}_k^*, H)) \\
&\overset{\text{(Eqn 3.1)}}{=} \arg\min_k \mathbb{E}\left(SSE(H_k, H) + \frac{2k}{\epsilon^2}\right) \\
&= \arg\min_k \mathbb{E}\left(SSE(H_k, H) + \frac{2(n-k)}{\epsilon^2} - \frac{2n-4k}{\epsilon^2}\right) \\
&\overset{\text{(Eqn 3.2)}}{=} \arg\min_k \mathbb{E}\left(SSE(\hat{H}_k^*, \widetilde{H}) - \frac{2n-4k}{\epsilon^2}\right)
\end{aligned}$$

However, it is not possible to compute $\mathbb{E}\left(SSE(\hat{H}_k^*, \widetilde{H})\right)$ without knowledge of H. The authors of [7] choose to estimate $\mathbb{E}\left(SSE(\hat{H}_k^*, \widetilde{H})\right)$ by $SSE(\hat{H}_k^*, \widetilde{H})$.

Hence, for the two-dimensional case the algorithm then returns $\hat{H}^*_{k^*}$ where k^* is chosen as

$$\begin{aligned} k^* &= \arg\min_{k\in\{1,2\}} \left\{ \left(SSE(\hat{H}^*_k, \widetilde{H}) - \frac{4-4k}{\epsilon^2} \right) \right\} \\ &= \arg\min \left\{ SSE(\hat{H}^*_1, \widetilde{H}), 4/\epsilon^2 \right\} \end{aligned}$$

We can expand $SSE(\hat{H}^*_1, \widetilde{H})$ as

$$SSE(\hat{H}^*_1, \widetilde{H}) = \left(\frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2) - \widetilde{h}_1 \right)^2 + \left(\frac{1}{2}(\widetilde{h}_1 + \widetilde{h}_2) - \widetilde{h}_2 \right)^2 \tag{3.3}$$

$$= \frac{1}{4}\left(\widetilde{h}_2 - \widetilde{h}_1 \right)^2 + \frac{1}{4}\left(\widetilde{h}_1 - \widetilde{h}_2 \right)^2 \tag{3.4}$$

$$= \frac{1}{2}\left(\widetilde{h}_1 - \widetilde{h}_2 \right)^2 \tag{3.5}$$

Hence, we select $k^* = 1$ if

$$SSE(\hat{H}^*_1, \widetilde{H}) = \frac{1}{2}\left(\widetilde{h}_1 - \widetilde{h}_2 \right)^2 \leq \frac{4}{\epsilon^2}$$

$$\iff \left| \widetilde{h}_1 - \widetilde{h}_2 \right| \leq \frac{2\sqrt{2}}{\epsilon} \tag{3.6}$$

Otherwise, we select $k^* = 2$. Note that the right-hand side of 3.6 can also be written as $n\sigma$, where σ is the standard deviation of the Laplace$(1/\epsilon)$ distribution.

3.2.2 Sampling Approach

Given a histogram H and a privacy parameter ϵ, we can estimate the expected SSE by generating a large number of noisy versions of H and computing the average SSE of the NoiseFirst algorithm over all such noisy histograms. This procedure is detailed in Algorithm 2.

Algorithm 2 $\hat{\mathbb{E}}(||\text{NoiseFirst}(\widetilde{H}, \epsilon) - H||_2^2)$

Input: A 2D histogram H, a privacy parameter ϵ, an integer number of trials n
Output: An estimate of the expected SSE of the NoiseFirst algorithm on H and ϵ

totalSSE $= 0$
for i in $\{1, 2, 3, \ldots, n\}$ **do**
    $\eta_1, \eta_2 \leftarrow$ independent random values following Laplace$(1/\epsilon)$ distribution
    $\widetilde{H} \leftarrow H + (\eta_1, \eta_2)$
    totalSSE $\leftarrow$ totalSSE $+||\text{NoiseFirst}(\widetilde{H}) - H||_2^2$
averageSSE $\leftarrow$ totalSSE$/n$
return averageSSE

From here, to find a counterexample we simply need to compute $\hat{\mathbb{E}}(||\text{NoiseFirst}(\widetilde{H}) - H||_2^2, \epsilon)$ for varying histograms and privacy parameters until we find a pair (H, ϵ) which has a higher estimated SSE than the identity. Algorithm 3 describes how to do so if we specify a set of valid histogram counts $C \subset \mathbb{Z}$ and a set of privacy parameters $E \subset \mathbb{Z}$.

Algorithm 3 Searching for a Counterexample

Input: A finite set of histogram counts $C \subset \mathbb{Z}$, a finite set of privacy parameters $E \subset \mathbb{R}_{\geq 0}$

for $\epsilon \in E$ **do**
  for $H \in C \times C$ **do**
    NoiseFirstErrEstimate $\leftarrow \hat{\mathbb{E}}(||\text{NoiseFirst}(\widetilde{H}, \epsilon) - H||_2^2)$
    if NoiseFirstErrEstimate $> 4/\epsilon^2$ **then**
      return H, ϵ
return failure

With this experimental setup, we all investigated two-dimensional histograms with counts in $[0, 100] \subset \mathbb{Z}$ for all privacy parameter values in $\{0.1, 0.2, \ldots, 1.0\}$ with $50,000$ trials per (H, ϵ) pair. Our numerical experiment did not find any (H, ϵ) pair such that the estimated NoiseFirst SSE was higher than the identity SSE of $4/\epsilon^2$.

3.2.3 Numerical Integration Approach

A second approach to estimating the expected SSE of the NoiseFirst algorithm is to directly compute the expected value via numerical integration. This technique is in some sense superior to that described above, as it is less susceptible to sampling bias and gives numerical error bounds.

If x is a random variable with probability distribution function $f(x)$ with support S, the expected value is defined as

$$\mathbb{E}(x) = \int_S x f(x) dx$$

If we want to compute the expected value of a function of a random variable x, the Law of the Unconscious Statistician tells us

$$\mathbb{E}(g(x)) = \int_S g(x) f(x) dx$$

For a fixed histogram H and a fixed privacy parameter ϵ, we can use this to compute $\mathbb{E}(||\text{NoiseFirst}(\widetilde{H}, \epsilon) - H||_2^2)$. In the two-dimensional case, we have that $\widetilde{H} = H + (x, y)$ is dependent on the independent Laplace$(1/\epsilon)$ distributed variables x and y. As a result of the independence assumption, the joint probability distribution is $f_{X,Y}(x, y) = f_X(x) f_Y(y)$, where $f_X(x)$ and $f_Y(y)$ represent the PDF of the Laplace$(1/\epsilon)$ function. Using these facts, we can compute the expected SSE for the NoiseFirst algorithm as

$$\mathbb{E}(||\text{NoiseFirst}(\widetilde{H}, \epsilon) - H||_2^2) = \int_{\mathbb{R}^2} ||\text{NoiseFirst}(\widetilde{H}, \epsilon) - H||_2^2 f_X(x) f_Y(y) dx dy$$

From here, the numerical integration package in SciPy was used in order to evaluate this integral for all two-dimensional histograms with counts in $[0, 1000] \subset \mathbb{Z}$ for all privacy parameter

values in $\{0.1, 0.2, \ldots, 3.0\}$. This search did not result in any pair (H, ϵ) such that the NoiseFirst algorithm was expected to perform worse than the identity estimator.

In all, these experiments give us evidence in favor of the claim that the NoiseFirst algorithm is expected to improve at least as well as the identity.

3.2.4 Towards a Proof

In this section we discuss what would need to be proved to show that the expected error of the NoiseFirst algorithm is no larger than that of the identity estimator. When the NoiseFirst algorithm chooses $k^* = 2$, the algorithm is the same as the identity estimator. Hence, we need to compare the expected error of the NoiseFirst algorithm in the case when $k^* = 1$ to that of the identity.

Let $H = (d_1, d_2)$, and let $\widetilde{H} = (d_1 + v, d_2 + w)$. Consider the special case $\epsilon = 2\sqrt{2}$, so that the NoiseFirst algorithm returns $k^* = 1$ when

$$|d_2 + w - (d_1 + v)| \leq 1 \qquad \text{(by Equation 3.6)}$$

We can then make use of the integral definition of the expected value. That is,

$$\mathbb{E}\left(SSE(\hat{H}_1^*, H) \middle| k^* = 1\right) \leq \mathbb{E}\left(SSE(\widetilde{H}, H) \middle| k^* = 1\right)$$

is equivalent to

$$\int_{-\infty}^{\infty} \int_{v+d_1-d_2-1}^{v+d_1-d_2+1} \left[(\frac{1}{2}(d_1 + v + d_2 + w) - d_1)^2 + (\frac{1}{2}(d_1 + v + d_2 + w) - d_2)^2 \right] f_w f_v dw dv \tag{3.7}$$

$$\leq \int_{-\infty}^{\infty} \int_{v+d_1-d_2-1}^{v+d_1-d_2+1} (v^2 + w^2) f_w f_v dw dv$$

We leave proving this inequality as an open problem.

3.3 Searching for a Counterexample: Boundary Inflated vs Resized

In this section, we search for a counterexample showing that the resized estimator is expected to perform worse than the boundary inflated estimator. Recall that the boundary inflated estimator returns $\widetilde{H}_+ = \max(0, \widetilde{H})$ and the resize estimator returns $\widetilde{H}_+$ where the counts have been uniformly compensated to preserve the original total count of $\widetilde{H}$.

We begin our search with a sample-based approach analogous to Section 3.2.1. This numerical experiment found that for $H = [0, 16]$ with $\epsilon = 0.1$ and 10^7 trials,

$$\mathbb{E}(SSE(Resize(\widetilde{H}), H)) \approx 251.2000 > \mathbb{E}(SSE(\max(0, \widetilde{H}), H)) \approx 247.676$$

Using the approach of Section 3.2.2, we used the numerical intergration package of SciPy to directly evaluate the expected error of both estimators on the given H and ϵ. Doing so, we obtain

$$\mathbb{E}(SSE(\max(0, \widetilde{H}), H)) \approx 247.5069053167384$$

$$\mathbb{E}(SSE(Resize(\widetilde{H}), H)) \approx 251.00037027604492$$

In both cases the error bound is less than 10^{-5}, which confirms our sampling results. As such, we have found a potential candidate showing on which the resize estimator is worse than the boundary inflate estimator.

3.4 Searching for a Counterexample: Resized vs Identity

In this section, we search for a counterexample showing that the resized estimator is expected to perform worse than the identity estimator. Using both the sample-based and integration-based methods of Section 3.2.1 and Section 3.2.2, we investigated all two-dimensional histograms with counts less than 1000 for all $\epsilon \in \{0.1, 0.2, \ldots, 3.0\}$. However, we did not find any pair in which the resized estimator was expected to perform worse than the identity estimator in terms of expected SSE.

3.5 Next Steps

While the numerical experiments of Section 3.3 and Section 3.4 provide us with evidence as to the comparisons we may draw between such estimators, this evidence is not conclusive. As such, we desire to prove two things:

- For $H = [0, 16]$ and $\epsilon = 0.1$, the resized estimator has a higher expected SSE than the boundary inflated estimator.
- For all histograms H and all $\epsilon \in \mathbb{R}_{\geq 0}$, the identity estimator has a higher expected SSE than the resized estimator.

Bibliography

[1] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–444, 1977.

[2] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, July 2006.

[3] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *CoRR*, abs/1605.02065, 2016.

[4] Ilya Mironov. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017.

[5] Cynthia Dwork, , , , and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer Verlag, May 2006.

[6] Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. *CoRR*, abs/1808.09501, 2018.

[7] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *2012 IEEE 28th International Conference on Data Engineering*, pages 32–43, April 2012.

[8] H. V. Jagadish, Viswanath Poosala, Nick Koudas, Ken Sevcik, S. Muthukrishnan, and Torsten Suel. Optimal histograms with quality guarantees. In *In VLDB*, pages 275–286, 1998.

Gavin Kerrigan Academic Vita

CONTACT INFORMATION GavinMKerrigan@gmail.com

EDUCATION

The Pennsylvania State University, State College, PA
Schreyer Honors College, expected May 2019
B.Sc. in Mathematics, minor in Physics

ACADEMIC EXPERIENCE

The Pennsylvania State University
Research Assistant, Department of Computer Science *February 2018 to Present*

- Researched and developed privacy-preserving algorithms

Independent Study *May 2017 to August 2018*

- Studied Atiyah-MacDonald's *Introduction to Commutative Algebra* thoroughly under the supervision of Penn State math faculty
- Solved challenging exercises in preparation for graduate-level coursework

Pulsar Search Collaboratory, State College, PA
Data Team Analyst and Researcher *August 2015 to May 2017*

- Analyzed statistical data through modeling and graphics in order to discover and classify astronomical objects
- Resulted in a poster presentation to faculty of the astronomy department

WORK EXPERIENCE

Penn State Learning, State College, PA
Peer Coordinator *January 2018 to Present*

- Ensured smooth operation of the tutoring center through conflict resolution and team communication

Mathematics Tutor *January 2017 to Present*

- Tutored undergraduates in mathematics courses ranging from College Algebra to Differential Equations
- Led collaborative exam review sessions on Differential Equations with more than 200 students in attendance with high levels of satisfaction

HONORS AND AWARDS

The Pennsylvania State University

- Steven and Sherry McCrystal Mathematics Award, 2019
- Mary Lister McCammon Scholarship (awarded for academic excellence in mathematics), 2017, 2018
- Best Presentation in Functional Analysis, 2017
- Phi Beta Kappa Honors Society, 2017

SKILLS

Languages: English (native), Spanish (proficient)
Programming: Python, C++, Java, Matlab
Software: LaTeX, Git, Minitab