Bootstrap Inference and Differential Privacy: Standard Errors for Free[*]

Thomas Brawner[†] James Honaker[‡]

July 15, 2018

1 Introduction

The bootstrap is a common and powerful statistical tool for numerically computing the standard error of estimators, that is, a calculation of the uncertainty of functions computed on sample data so as to make an inference back to the original population from which the sample was drawn. Understanding uncertainty, and inferential questions, in the context of private data is an increasingly important task within the literature of differential privacy [7, 20, 15].

We show how to construct an implementation of the bootstrap within differential privacy. Most importantly, we show that, for a broad class of functions under zero concentrated differential privacy, the bootstrap can be implemented at no cost. That is, for a given choice of privacy parameter and associated expected error of some query, the bootstrap can be implemented for the exact same privacy guarantee, resulting in the same expected error (or sometimes less) in the desired query, but additionally provide the standard error of that query.

In section 2 we provide a brief overview of differential privacy. Then to describe these results on bootstrap inference, in section 3 we describe some foundational results on the aggregation of repeated queries under contrasting privacy and composition definitions. This leads to a tangential result in section 4 on a low-noise Gaussian mechanism for pure differential privacy. Next we provide a brief foundation on the bootstrap algorithm in statistics in section 5, before showing our algorithmic construction of the bootstrap using the mechanisms of differential privacy in section 6. In section 7 we describe how to use the differentially private estimate of the standard error in the construction of confidence intervals and hypothesis tests, and then demonstrate this in section 8 with examples using published Census microdata in the style of privacy sensitive data.

2 Differential Privacy

Social scientists and other human subjects researchers often want to analyze data that contains information that must remain private, for ethical or legal reasons, or to prevent the loss of trust or even harm to participants [1]. The increasing ability of linked data collections, the ubiquity of sensors, and the ability of social media to measure individual behavior in nuance, ensures that such privacy concerns continue to dramatically increase [4].

Differential Privacy, deriving from roots in cryptography, is a formal, mathematical conception of privacy preservation [8]. Instead of attempting to produce a "de-identified" dataset, differential privacy allows the release of statistical summaries, queries or estimates from a dataset, and even allows future data analysts

[*] For helpful comments we thank John Abowd, Vishesh Karwa, Ilya Mironov, Kobbi Nissim, Adam Smith, Jonathan Ullman, Salil Vadhan, and participants of the Theory and Practice of Differential Privacy 2017 (TPDP2017) Workshop.

[†] Center for Research on Computation and Society, Harvard John A. Paulson School of Engineering and Applied Sciences. thomasbrawner@gmail.com

[‡] Center for Research on Computation and Society, Harvard John A. Paulson School of Engineering and Applied Sciences. james@hona.kr; http://hona.kr

to make their own statistical queries of the data. It guarantees that any released statistical result does not reveal information about any one single individual.

Informally, to satisfy the definition of differential privacy, the distribution of answers one would get with an algorithm from a dataset that does not include myself must be so close as to be indistinguishable from the distribution of answers where I have added my own information. Thus I have no reason not to add my own personal data to a dataset, as no released answers can leak my information. A differentially private algorithm injects a precisely calculated quantity of noise to any statistical query to mask the possible contribution of any one individual to the result. It then becomes mathematically provable that no possible combination of queries or model results can tease out information that is specific to any individual data subject. This is a nontrivial guarantee – if insufficient noise is introduced (without applying the theory of differential privacy), then there are known attacks that combine many "aggregate" statistics to infer sensitive attributes of specific individuals [17, 10, 6].

Using differential privacy enables one to provide wide access to statistical information from a privacy sensitive dataset without worries of individual-level information being leaked inadvertently or due to an adversarial attack. There is now both a rich theoretical literature on differential privacy and numerous efforts to bring differential privacy closer to practice. Large technology companies such as Google [12], Apple [16] and Uber [19] have started using differential privacy to protect the data of their customers. Most Census products from the 2020 Decennial Census will be differentially private releases [5]. The PSI (Ψ): Private data Sharing Interface [14] allows differentially private access to research data in repositories such as Dataverse [21, 3].

However, the large literature of DP statistics primarily provide point estimates, and few current DP algorithms exist to provide standard errors or confidence intervals. Those that do exist [7, 15, 20] provide very noisy estimates of uncertainty, and also draw from the "privacy budget" of the privacy loss parameter, thus making the release of the original quantity of interest itself noisier.

As a overview of key concepts that will be used in this paper, we show a simple but effective class of re-identification attacks, then provide the definition of a differentially private algorithm, an example of such an algorithm, and highlight some key properties that DP algorithms obtain. For a more detailed overview, Dwork and Roth [9] provide a technical review of the literature, and Nissim *et al.* [22] a non-technical introduction.

2.1 Re-identification Attacks

The mean, given by $\bar{x} = \sum_i^N \frac{x_i}{N}$, like many statistical queries, aggregates individual information into a larger quantity that appears to lose the individual into the population level information. Often, simple statistical disclosure limitation heuristics attempt to safeguard privacy by not releasing means that contain fewer than some threshold number k of individuals, with the idea that if k is large enough, the individual level information is lost in the sum. However, each individual contributes exactly x_i/N to the mean function. Consider a setting where x are student test scores, and $\bar{x}$ is released periodically for classes. If I happen to know that Alice moved into a class at a particular time (and no one else changed), and see the release $\bar{x}_t$ just before she joined and $\bar{x}_{t+1}$ just after, then Alice's data is the sole reason that the mean score changed, and I can back out Alice's private test score exactly.[1] This style of attack, where the difference between two seemingly benign aggregate statistics can leak individual information is . We could for example release regression coefficients on large subsets of the data, but if they are carefully subsetted by geography, time, or other predicates to add or remove one individual then the difference between regressions exactly reveals that individual's information. In general, an increasing host of sophisticated re-identification attacks exist that use auxiliary information or combinations of innocuous aggregate queries to precisely re-identify individual's information from datasets.

[1] In this case, $x_{\text{Alice}} = N_{t+1}\bar{x}_{t+1} - N_t\bar{x}_t$.

2.2 Indistinguishability and Differential Privacy

Differential privacy states that the distribution of answers released by an algorithm should appear almost the same regardless of the inclusion of any one possible observation. Formally, for any two *neighboring datasets*, X and X' that differ only in one observation, then a function M is differently private if:

$$Pr[T(M(X)) = 1] \leq e^{\epsilon} Pr[T(M(X')) = 1] + \delta, \qquad \forall\ T, X, X'. \tag{1}$$

where $T(.)$ is any decision rule based on the function output. This says that the distribution of the outputs of a function, and their inferences or consequences are *close*, regardless of any one observation in the dataset. Thus any observing the output of a differentially private release can not distinguish whether one observation was in the dataset, or it's values. Thus any individual should feel safe to be included in the data, since their information will not effect any answers.

The particular definition of closeness or indistinguishability in equation 1 uses two *privacy loss parameters*, ϵ and δ, which formalize what it means for two distributions to be close. Specifically, the ratio of the distribution of outcomes must be within a factor of e^{ϵ} where ϵ is typically $0.5 > \epsilon > 0.01$, and δ is some additional very small factor commonly $\delta < 1e^{-6}$ or less than $1/N$. When $\delta = 0$ then we refer to this as *pure differential privacy* and for $\delta > 0$ this is *approximate differential privacy.* Other definitions of the closeness of the distributions use the Kullback-Leibler divergence or the Rényi divergence of the distributions, which give alternate definitions named *concentrated* and *zero-concentrated differential privacy* respectively, and which we will see permit slightly different algorithms and composition theorems, and label their parameter ρ.

Key here is that the relevant parameter, be it ϵ, or (ϵ, δ) or ρ, measures the total worst-case informational leakage or privacy loss from the dataset. If there is a limit to the allowable privacy loss, this is referred to as the *privacy budget.*

2.2.1 Post Processing and Composition

Differential Privacy is a definition that an algorithmic mechanism can either be proven to satisfy or not satisfy. Statistical releases from algorithms that satisfy DP have two important key properties. First, they are immune to *post processing*, which means a release from a differentially private algorithm can undergo any transformation and it is still remains differentially private (with the same ϵ value). Second they *compose*, which means that if a larger algorithm is constructed from pieces which are each themselves differentially private, then the total algorithm is differentially private, and the grand privacy loss ϵ of the larger algorithm can be obtained as a function of the constitutent $\epsilon_1, \dots, \epsilon_k$ privacy loss parameters of the underlying peices. In the simplest form, this just additive, in that $\epsilon = \sum \epsilon_i$, but more advanced composition rules can lead to lower total privacy loss, and thus more queries, or less noisy queries, for the same privacy budget.

2.2.2 Sensitivity and the Laplace Mechanism

The Laplace is a convenient distribution to use for this noise, given the construction of the definition of differential privacy. The Laplace has distribution function:

$$f_{Laplace}(x|b, \mu) = \frac{1}{2b} \exp\Big(- \frac{|x - \mu|}{b} \Big) \tag{2}$$

with mean μ, and variance $2b^2$. The Laplace is a mirrored, and thus symmetric version of the exponential distribution. The exponential is common to survival and event history models, which use its *memoryless* distribution[2], which we are also about to exploit. To make a continuous variable differentially private, we add a draw from a mean zero Laplace, with parameter b as:

$$b = \frac{\Delta}{\epsilon} \tag{3}$$

[2] The hazard function of an exponential waiting time, as the ratio of two exponentials, is a constant.

Where ϵ is our privacy loss parameter as previously described and Δ is a quantity known as the *sensitivity* which calculates the *worst case change in a function that could occur from changing one observation in the data*. In the example of a mean of a variable that has range R, this is $\Delta = R/N$, the change in the mean that would occur from moving an observation from the minimum value to the maximum value. Sensitivity is key to many differentially private algorithms, as it is the largest effect one persons information could have on a statistical release, and thus noise needs to be calibrated to drown out this magnitude of effect.

From this, our differentially private mean, $M(X)$, which combines the "true" sample mean with Laplace noise, becomes:

$$M(X) = \bar{X} + Y; \qquad Y \sim f_{Laplace}(b = \Delta/\epsilon, \mu = 0) \tag{4}$$

To check this mechanism meets the definition of differential privacy, consider some probability of any outcome, z. The ratio of this probability between two adjacent datasets, is given by:

$$\frac{pr[M(X) = z]}{pr[M(X') = z]} = \frac{e^{\frac{-\epsilon|\bar{X}-z|}{\Delta}}}{e^{\frac{-\epsilon|\bar{X}'-z|}{\Delta}}} = e^{\frac{\epsilon|\bar{X}'-z| - \epsilon|\bar{X}-z|}{\Delta}} = e^{\frac{\epsilon|\bar{X}'-\bar{X}|}{\Delta}} \leq e^{\epsilon} \tag{5}$$

the last step following since we know $\Delta \geq |\bar{X}' - \bar{X}|$ by the definition of the sensitivity. It thus follows that $Pr[M(X) = z] \leq e^{\epsilon} Pr[M(X') = z]$, thus meeting the definition of ϵ-differential privacy (in this case, with parameter $\delta = 0$). For other continuously valued summary statistics, the same Laplace mechanism works for preserving privacy, however, the analytic form for the sensitivity, Δ, will change by statistic.

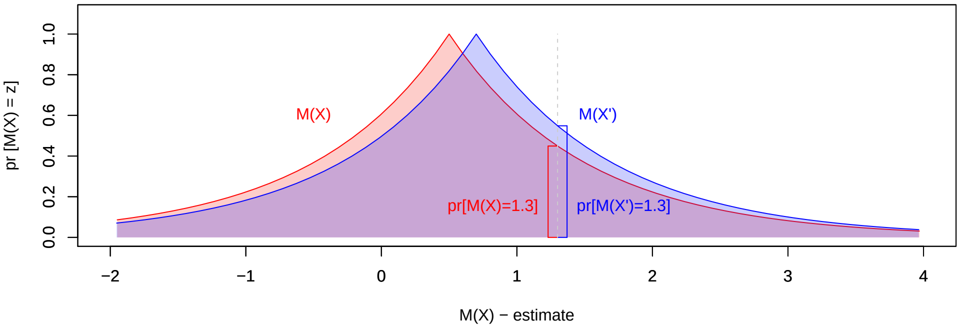


Figure 1: *Two Laplace distributions, for two adjacent datasets X and X'. The definition of ϵ-differential privacy requires the ratio of $M(X)/M(X')$ is not greater than e^{ϵ} for all points along the x-axis. Thus for any realized output z – for example here, $z = 1.3$ – we can not determine that X or X' were more likely to have produced z.*

3 Aggregation of Repeated DP Statistical Releases

Consider an analyst receives two differentially private estimates, m_1 and m_2, of the mean of the same variable from the same dataset. Each has been released using the Laplace mechanism, using privacy parameters ϵ_1 and ϵ_2 respectively.

Rather than relying on either answer, they want to post-process the releases to construct a better answer. It is intuitive to average them, since they are both estimates of the same quantity, however, if one ϵ is greater than the other, then the released statistic has less noise and less expected error, and thus should be given greater weight. The efficient answer, in the statistical meaning of unbiased and minimum variance, for a sequence of means $m_1, \cdots, m_k$, is given by:

$$m^* = \sum_{i=1}^{k} w_i m_i; \qquad w_i = \frac{\epsilon_i^{-2}}{\sum_{j=1}^{k} \epsilon_j^{-2}} \tag{6}$$

where we are summing together a weighted mean, and the weights are inversely proportional to the variances, which here are simply a function of ϵ. Moreover, while each release has variance from the Laplace of $2(\Delta/\epsilon_i)^2$ (where Δ is the sensitivity of the particular mean), this new mean estimate has variance:

$$\sigma^2(m^*) = \sum_{j=1}^{k} w_j^2 \sigma^2(m_i) = 2\Delta \sum_{j=1}^{k} \frac{w_j^2}{\epsilon_i^2} \tag{7}$$

Weighting by inverse of the variance is well known in measurement theory and particularly in approaches to the problem of measurement error. A proof in the context of differential privacy is given by [18].

3.1 Equipartition

Consider then an analyst wants to get an estimate of the mean of a variable while staying within a global privacy parameter of some definition of privacy loss; we will consider both the ϵ of pure differential privacy (DP) and ρ of zero concentrated differential privacy (zCDP) [2].

The simplest approach is to ask one query exhausting the entire privacy parameter, for example:

Method 1:

$$m_{DP} = \frac{1}{N} \sum_{i=1}^{N} x_i + \mathcal{L}(0, \Delta/\epsilon); \qquad \sigma^2(m_{DP}) = 2(\Delta/\epsilon)^2 \tag{8}$$

$$m_{zCDP} = \frac{1}{N} \sum_{i=1}^{N} x_i + \mathcal{N}(0, \Delta^2/2\rho); \qquad \sigma^2(m_{zCDP}) = \Delta^2/2\rho \tag{9}$$

Where $\mathcal{L}$ and $\mathcal{N}$ represent Laplace and Normal distributions respectively.

A second approach is to partition ϵ or ρ to make k queries of the mean of some variable resulting in releases $m_1, \cdots, m_k$. Under simple composition, for the i-th query we assign $\epsilon_i \leq \epsilon/k$. In the limit for large k, under advanced composition we can instead use $\epsilon_i \leq \epsilon/\sqrt{k \ln(1/\delta)}$, but have to use $\delta > 0$. Under zCDP we can use $\rho_i \leq \rho/k$. If the analyst then recombines these answers, we get an estimator for the mean as:

Method 2:

$$m^* = \frac{1}{k} \sum_{i=1}^{k} m_i; \tag{10}$$

where the variance of the estimate is determined by the composition method used:

Basic Composition $(\epsilon, 0)$-DP:

$$\begin{aligned} \sigma^2(m^*) &= 2\Delta^2 \sum_{j=1}^{k} (1/k)^2 (k/\epsilon)^2 \\ &= 2k(\Delta/\epsilon)^2 n = k \ \sigma^2(m_{DP}) \end{aligned} \tag{11}$$

Advanced Composition (ϵ, δ)-DP:

$$\begin{aligned} \sigma^2(m^*) &= 2\Delta^2 \sum_{j=1}^{k} (1/k)^2 (\sqrt{k \ln(1/\delta)}/\epsilon)^2 \\ &= 2\ln(1/\delta)(\Delta/\epsilon)^2 = \ln(1/\delta) \ \sigma^2(m_{DP}) \end{aligned} \tag{12}$$

zero Concentrated DP $(0, \rho)$-zCDP:

$$\sigma^2(m^*) = \frac{\Delta^2}{2} \sum_{j=1}^{k} (1/k)^2 \frac{k}{\rho} = \Delta^2/2\rho = \sigma^2(m_{zCDP}) \tag{13}$$

We see here the expansion in the variance from partitioning the privacy parameter, and recombining the releases, compared to the conventional one-shot release given in equation 8. Under either form of composition under differential privacy, we always get an answer with higher error than if the budget had been assigned to one single query (inflation factors in red). Under zero concentrated differential privacy we get exactly the same variance as the one-shot release.

Thus under composition with zero concentrated differential privacy, we get exactly the same expected error from releasing k queries equally partitioning the privacy parameter, as we would if we had simply made one encompassing query. This is a central point that will be utilized in the following work so we formalize it as:

Remark 1 *Consider two zero Concentrated Differentially Private estimators of an additive function, m and m^*, where m is one query using ρ and m^* is the average of k such queries each using ρ/k. Then $\sigma^2(m) = \sigma^2(m^*)$.*

4 A Pure-DP Lower-noise Gaussian Mechanism

In addition to the main remark, which will subsequently draw our attention toward zCDP, there are additional valuable points to explore about the partitioning of a query under pure differential privacy.

First, the distribution of the one-shot release, m, in equation 8 is the Laplace, however, the distribution of m^*, the averaged value of k Laplace releases tends to a Gaussian, asymptotically as k increases, due to the central limit theorem.

Second, with basic composition the variance goes up by a factor of k. At first this seems that partition coupled with basic composition is an inefficient means of release, however, recall the standard Gaussian mechanism in differential privacy has variance $2\ln(1.25/\delta)(\Delta f/\epsilon)^2$. Figure 2 shows the value of $\ln(1.25/\delta)$ is on the order of 10 to 20 for typical values of δ. So for $k < \ln(1.25/\delta)$ the average of k Laplace queries has lower error than the Gaussian mechanism, *as well as being pure differentially private*, rather than using approximate differential privacy as the conventional Gaussian mechanism requires.

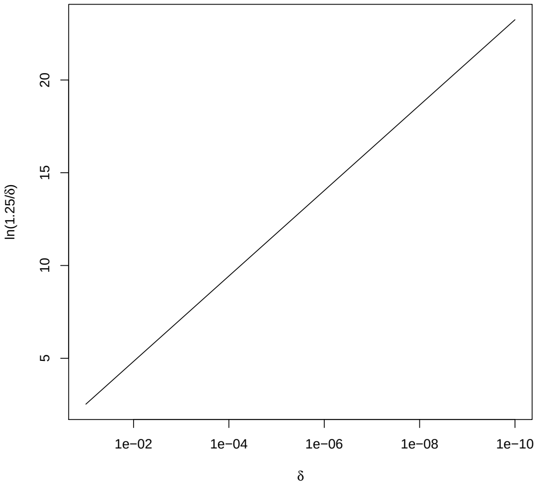


Figure 2: *Factor size of* $\ln 1.25/\delta$ *as a function over typical values of* δ.

The average of the Laplace approaches Normality quite speedily. Figure 3 shows two methods of judging the speed of this convergence. We take a hundred draws from the average of k Laplace draws, where k goes from 1 to 20. From this, we test whether the 100 draws using the average can be distinguished from a Normal distribution. The top figure shows the fraction of the time the Anderson-Darling and the Shapiro-Wilk tests fail to reject the (incorrect) null hypothesis that the data was drawn from a Normal distribution (here using $p = 0.05$). By the time we averaging 10 draws from the Laplace, both tests are failing to reject the hypothesis that the data actually came from a Normal distribution over 90 percent of the time (90.8 and 92.7 respectively). These p values are approximations, so for a stronger test we use the following numerical experiment. We create two datasets of size 100. One dataset is actually drawn from the Normal, and the other created by averaging k draws from the

Laplace. We then use the test statistic to try to judge which of the two datasets is more likely to be the true data drawn from a Normal. For small k the correct dataset is commonly picked, but by k of 10, the Normal dataset is being chosen less than 55 percent of the time, and more than 45 percent of the time the data from the average of Laplaces is being judged as more likely to be Normal than the actual Normal data. Thus we conclude the average of 10 Laplace draws is functionally numerically indistinguishable from draws from a Normal.

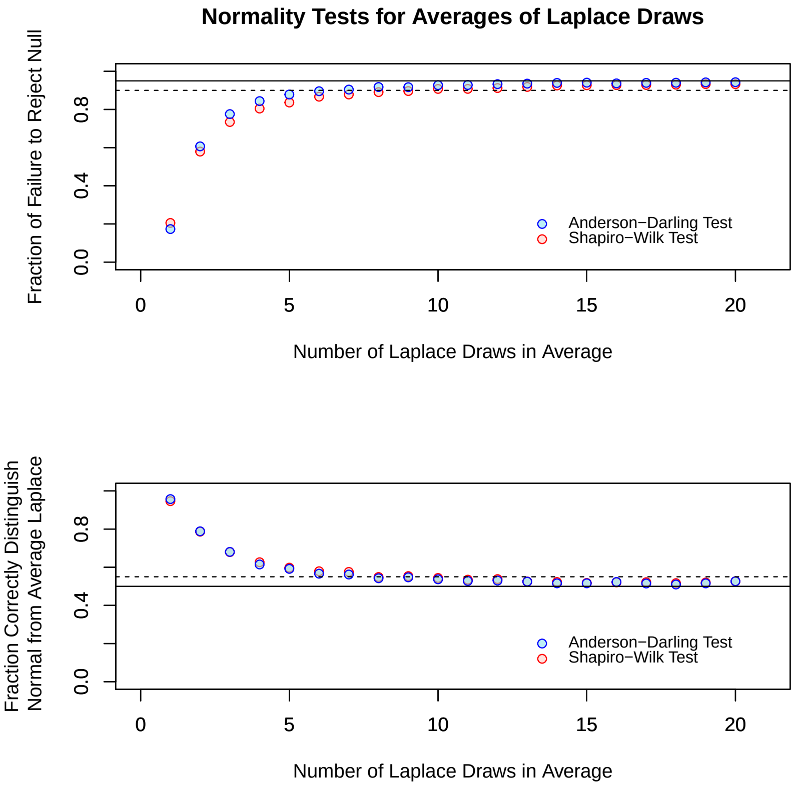


Figure 3: *Tests of deviation from normality for datasets of 100 observations of the average of k Laplace draws. The top graph shows the fraction of time tests fail to reject the null hypothesis that the data came from a Normal distribution, across an increasing range of k. The bottom graph shows the fraction of the time a dataset actually drawn from Gaussians is correctly selected in a choice between a Gaussian and average of k Laplace draws. By both measures, the k-average Laplace is appearing numerically indistinguishable from the Normal for k of 10 or greater.*

This provides a very useful mechanism for pure differential privacy that is *numerically indistinguishable from the Gaussian, but has both lower noise, and the ease of pure differential privacy.*

Remark 2 *Consider two estimators of an additive function, m and m^*, where m is $(\epsilon, \delta) - DP$ by the Gaussian mechanism, and m^* is the average of k independent $(\epsilon/k, 0) - DP$ releases from the Laplace mechanism. Then $\sigma^2(m^*) < \sigma^2(m)$ $\forall k < ln(1/\delta)$ while m^* is both approximately Normal for $k > 10$, and $(\epsilon, 0)$-differentially private.*

As one final note, in the case of advanced composition in equation 12, we return to approximate differential privacy, but the average of k Laplace draws appears to have variance scaled by $\ln 1/\delta$, rather than the slightly larger $\ln 1.25/\delta$ typically employed for the Gaussian mechanism.

5 Introduction to the Bootstrap

The bootstrap is part of a larger family of sampling techniques, including the jackknife and subsampling, used to numerically simulate the sampling distribution of a statistic θ. Generally, resampling is often employed as an alternative to analytical techniques for statistical inference when, for example, no analytical solution exists for the variance of the estimate, the theoretical assumptions underlying the sampling distribution do not hold, or the sample size n is too small for the asymptotic properties of the sampling distribution to hold. Thus, the bootstrap is an attractive technique for understanding the variance of θ when the analyst is unable or unwilling to analytically evaluate its sampling distribution.

For dataset $X = \{x_1, x_2, \ldots, x_n\}$, a bootstrap sample X^* is constructed by randomly drawing observations, each with probability $1/n$, from X with replacement, and $\hat{\theta}^*$ is the estimate learned from X^*, referred to as the bootstrap replication of $\hat{\theta}$. Figure 4 shows the probability that any observation is copied any particular number of times in a resampled dataset of 1000 observations. We see about a third of original observations are omitted entirely, a third are sampled exactly once, and the rest are sampled multiple times.

The sampling distribution for θ is simulated by evaluating a large number, say $J = 1000$, bootstrap replications of $\hat{\theta}$. The estimated standard error of $\hat{\theta}$ is then the sample standard deviation of the J bootstrap replications [11].

$$\widehat{\text{se}}_{\hat{\theta}} = \left\{ \frac{\sum_j (\hat{\theta}^*_j - \hat{\theta}^*)}{J-1} \right\}^{\frac{1}{2}} \tag{14}$$

Here, $\hat{\theta}^*$ is the mean across the bootstrap replications.

$$\hat{\theta}^* = \frac{1}{J} \sum_j \hat{\theta}^*_j \tag{15}$$

As implied in (15), the bootstrap provides an estimate of $\hat{\theta}$ itself in addition to its variance. This is commonly referred to as *bagging*, or *bootstrap aggregation* [13], and $\hat{\theta}^*$ as the bagged estimator. When $\hat{\theta}$ has a symmetric sampling distribution then $\mathbb{E}\hat{\theta}^* = \hat{\theta}$ while otherwise $\hat{\theta}^*$ is biased but has sufficiently lower variance to obtain lower mean squared error [13].

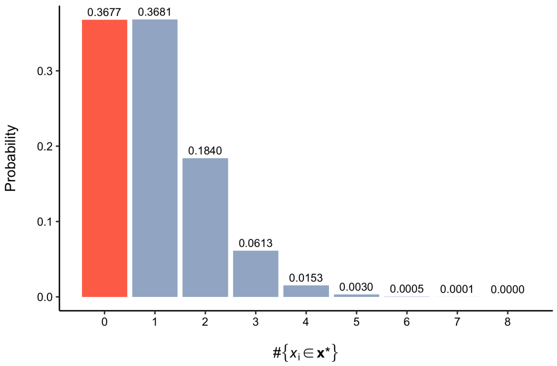


Figure 4: *Probability that observation x_i is sampled with given frequency into bootstrap sample* $\mathbf{x}^*$ *(Here the probabilities are shown specifically for a dataset of 1000 observations.)*

6 Composition for the Bootstrap

To show how privacy composes over the bootstrap, we first illustrate two naive approaches, and then we will build from these approaches to achieve a better result with lower noise.

6.1 Naive sensitivity factor

In the simplest case, imagine we want k bootstraps of a statistic given by function $f(.)$, which has sensitivity Δ, from budget ρ. Assume this function is additive[3]. As seen in Figure 4, very few observations ever occur in a bootstrapped dataset with more than 5 copies (and only 3 tenths of a percent of the data is copied five times). If we imagined a slightly bounded bootstrap, that limits any observation to appear at most j times (say 5), then the new sensitivity is at most $j\Delta$. That is, if an observation in the original dataset was changed, it might change up to j observations in the bootstrapped dataset, and lead to a worst case change is $j\Delta$ of the function result.

6.2 Naive Secrecy of the Sample

The ρ parameter in zCDP is amplified by subsampling the original data, as in secrecy of the sample. If observations have probability p of being included in the data, then the effective parameter, ρ', is:

$$\rho' = \frac{\rho}{p^2} \tag{16}$$

We can also see from figure 4 that some of the original observations are not included in the bootstrapped dataset. Each observation has a $1/n$ chance of being resampled for any particular row of the new dataset. There is then a $(1-1/n)^n$ chance of the i-th original observation not appearing anywhere in some particular bootstrapped dataset. In the limit:

$$\lim_{n\to\infty}(1-1/n)^n = e^{-1} \tag{17}$$

Therefore, e^{-1} fraction of the data does not appear in the bootstrapped dataset. We could use secrecy of the sample to improve on our previous privacy loss calculation. If we have a limited bootstrap with at most j copies of any observation, can calculate a function with sensitivity Δ in the original dataset, then the variance is:

$$\sigma^2(m) = \frac{(j\Delta)^2(1-e^{-1})^2}{\rho} \tag{18}$$

6.3 Private Bootstrap Construction

In summary, in the naive approach to bootstrapping with differential privacy, the potential repetition of observations leads to an increase in sensitivity of the calculated function, however, the stocastic omission of some observations leads to a boost in the functional privacy-loss parameter. We now show that with more careful design and analysis, under zCDP, these two factors can exactly cancel each other, that is any increase in sensitivity can be exactly offset by the boost in the functional privacy-loss parameter.

Consider the following implementation of the bootstrap. Instead of building one bootstrapped dataset conventionally by resampling, we are going to partition the original data into subsets according to how many times that observation has been selected to appear in the bootstrapped version. Conceptually, it is as if in figure 4 all the observations in each column are put in separate datasets.

We build a set of datasets $\mathbf{X} = \{X_0, X_1, \ldots\}$. Let R be an N-dimensional draw from the multinomial distribution as:

$$\mathbf{r} = \{r_1, \cdots, r_N\}; r_i = \text{Mult}(N, \pi_i); \pi_i = 1/N \ \forall\ i \tag{19}$$

We partition each observation into one of the datasets by this multinomial draw.

$$y_j \in X_i \iff |\mathbf{r} = j| = i \tag{20}$$

[3] That is, of the class $f(\mathbf{X}) = f(\mathbf{X_1}) + f(\mathbf{X_2})\ \forall \mathbf{X_1} \uplus \mathbf{X_2} = \mathbf{X}$. This includes all functions of the form $f(\mathbf{X}) = \sum_{i=1}^{N} f(x_i)$ including counts and means.

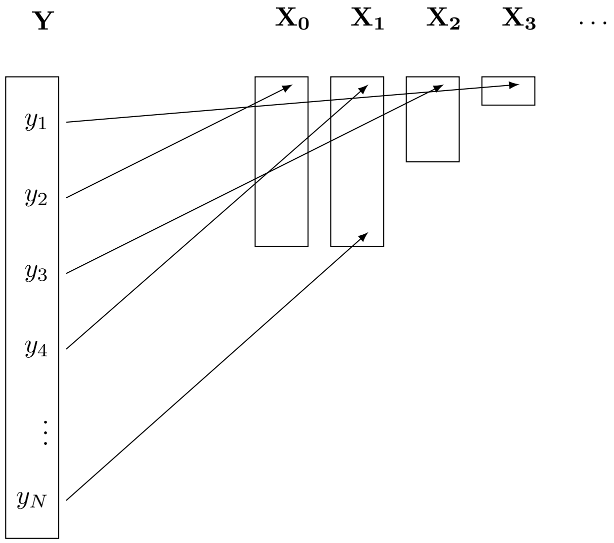


Figure 5: *Schematic of partition of original dataset Y across sequence of datasets X_0, X_1, ...*

Thus the probability that the i-th row is placed in the j-th partition is:

$$\Pr(y_j \in X_i) = \binom{N}{i} (1/N)^i \, (1 - 1/N)^{N-i} \tag{21}$$

This is shown schematically in figure 5. The original dataset Y is divided among datasets X_i, with the probability (and thus size of each X_i) following the density of figure 4.

We partition ρ as a function of i and p_i to give us a functional parameter ρ' as:

$$\rho_i = ip_i\rho \Rightarrow \rho'_i = ip_i\rho(1/p_i^2) = i\rho/p_i \tag{22}$$

which still crucially results in $\sum \rho_i = \rho$. That is, the partition exactly exhausts the privacy loss budget.

The i-th dataset now contains observations that would be repeated i times in a particular bootstrapped dataset. Thus we can calculate the function on each partition and multiply by i to see the partial contributions to the bootstrapped value.[4] On the i-th dataset we calculate:

$$M_i = i \cdot f(X_i) + \mathcal{N}(0, \sigma_i^2) \tag{23}$$

$$\text{where} \qquad \sigma_i^2 = \frac{(i\Delta)^2}{2\rho'_i} = \frac{p_i(i\Delta)^2}{2i\rho} = ip_i\frac{\Delta^2}{2\rho} \tag{24}$$

Which gives us the partial contributions of each subdataset to the grand sum. We then compute:

$$M = \sum_i M_i \tag{25}$$

Which itself has variance:

$$\sigma^2(M) = \sum_i \sigma^2(M_i) = \sum_i ip_i\frac{\Delta^2}{2\rho} = \frac{\Delta^2}{2\rho} \tag{26}$$

Thus the variance from the function run on our bootstrapped construction is the same variance as if we had run the function on the original data. This is a central result so we set it out as:

[4] In the case of the mean, it's important to note the function $f(X_i)$ is still the partial contribution to the mean $1/N \sum_{y_j \in X_i} y_j$ and not the mean of X_i e.g. $1/N_i \sum_{y_j \in X_i} y_j$.

Remark 3 *Consider two zero Concentrated Differentially Private estimators of an additive function, m and m', where m is the query run on a dataset X using ρ and m' is the query also using ρ on a bootstrap of dataset X. Then $\sigma^2(m) = \sigma^2(m')$.*

6.4 Key Result

In summary, under zCDP, by remark 3 the sensitivity of a function on the original data, and the sensitivity of the function on a bootstrap of the data are the same. By remark 1 the expected error of asking a query once using all ρ, and the error of asking k queries using ρ/k each, is the same. Therefore, when using the Gaussian mechanism on an additive function:

- Releasing a single query at ρ, has the same privacy loss as releasing k queries at ρ/k on k bootstraps of the original data.
- The Average of those k bootstrapped answers has the same utility in squared error as the single query.
- However, the distribution of the bootstraps also provides a standard error estimate at no additional loss in privacy.

More succinctly, we set this as our first result as follows:

Result 1 *Consider a differentially private release, m, of an additive function on sample dataset X using zCDP with some ρ, and $m^\star$, the average of k such queries each using ρ/k on a bootstrap of X. Then $\sigma^2(m) = \sigma^2(m^\star)$.*

7 Inference with Confidence Intervals

If an estimate has been calculated on a sample, the inferential task of the confidence interval is to provide bounds, with some associated guarantee, depicting the uncertainty of the population value that has been estimated from the available sample. Commonly, confidence intervals for means (drawing on the Normal approximation of the sampling distribution given by the Central Limit Theorem) are given by:

$$\mathrm{CI}_{1-\alpha}(\bar{x}) = \bar{x} \pm z_{\frac{\alpha}{2}} s(\bar{x}) \tag{27}$$

Where $1 - \alpha$ is the *coverage*, that is, the fraction of the time the confidence interval is expected to contain the population value, and $z_{\frac{\alpha}{2}}$ is a constant from the cumulative normal and $s(\bar{x})$ is the estimated standard error. Thus an intermediate step is to calculate a standard error, which is an estimate of the variance that would be observed across sample estimates computed on repeated samples.

Again, a primary use of the bootstrap is to estimate the standard error by the standard deviation of the bootstrapped values. However, we now tackle the problem that the standard error of privacy preserving bootstrapped values have been inflated by the necessary noise of the Gaussian mechanism.

7.1 Notation

As is common notation, let $\sigma^2(z)$ be the variance of a random variable z, and $\mathrm{s}^2(z)$ be the sample variance of k observed realizations.

$$s^2(z) = \frac{\sum_{i=1}^{k}(z_i - \bar{z})^2}{N-1} \qquad \text{where} \quad \bar{z} = \frac{1}{N}\sum_{i=1}^{k} z_i \tag{28}$$

Let $f = \{f_1, f_2, \cdots, f_k\}$ be the values of a function, such as the mean, computed on k bootstraps of the data, and similarly $m = \{m_1, m_2, \cdots, m_k\}$ the zCDP releases each using the Gaussian mechanism with privacy parameter ρ/k, all having mean $\bar{m}$. Following the notation originally introduced in equation 10, let the bootstrap averaged (bagged) estimator be $m^* = \bar{m}$. Let $n = \{m_1 - f_1, m_2 - f_2, \cdots, m_k - f_k\}$ be the errors which we know are drawn from a mean zero Gaussian distribution with variance $(\Delta k/2\rho)$.

7.2 Decomposition of the Variance

Since n is independent of f, the variances are additive, so we know:

$$\sigma^2(m) = \sigma^2(f) + \sigma^2(n|\rho/k) = \sigma^2(f) + \frac{\Delta^2 k}{2\rho} \tag{29}$$

The observed variance of m across k bootstraps, $s^2(m)$, is the feasible estimator of the variance of the bootstrap releases, $\sigma^2(m)$. That gives us an (unbiased) estimator of the variance of f as:

$$\hat{\sigma}^2(f) = \mathrm{s}^2(m) - \frac{\Delta^2 k}{2\rho} \tag{30}$$

Since for the mean, $\sigma^2(f) = sd(x)/\sqrt{N}$, then this also gives us an estimate of the standard deviation of the individual-level variable in the population, which may be a summary statistic of interest to the analyst.

We know that in any particular set of bootstraps, $\mathrm{s}^2(n)$ might be larger or smaller than $\sigma^2(n)$. On occasions where it was small in a particular set of bootstraps, then the estimator in 30 will underestimate $\sigma^2(f)$. As we will see, underestimating this may result in narrower confidence intervals with lower than promised coverage. So rather than using the expectation of $\sigma^2(m)$ we also consider a conservative lower bound.

The sampling distribution of the sample variance of the Normal is Chi-squared, with $k-1$ degrees of freedom,[5] specifically:

$$\mathrm{s}^2(n) \sim \chi^2_{k-1}\frac{\sigma^2(n)}{k-1} \tag{31}$$

We can compute the critical value of the Chi-squared, c_α defined such that:

$$\int_0^{c_\alpha} \chi^2_{k-1} = \alpha \tag{32}$$

And use:

$$\hat{\sigma^2}(n) = c_\alpha \frac{\sigma^2(n)}{k-1} \tag{33}$$

As a conservative bound of the $\mathrm{s}^2(n)$. That is, $1-\alpha$ percent of the time, the sample variance of the noise added in the k bootstrap releases will be larger than the value in equation 33.

[5] This follows directly from the Chi-squared being the distribution of the sum of the squares of Normal draws.

Using this conservative estimate, and drawing on eq. 30, we can create a revised and conservative estimate of $\sigma^2(f)$ as:

$$\hat{\sigma}^{2\star}(f, c_{\alpha'}) = \mathrm{s}^2(m) - \frac{\Delta^2 k}{2\rho}\frac{c_{\alpha'}}{k-1} \tag{34}$$

Where here conservative has a meaning adjusted by a parameter such that we expect α' fraction of the time that the actual $\sigma^2(f)$ is less than $\hat{\sigma}^{2\star}(f, \alpha')$. Note that for this conservative estimator, the adjustment term $c_{\alpha'}/(k-1)$ is the only difference from the previous unbiased estimator in eq 30. For $c_{\alpha'} = k-1$ we obtain the unbiased estimator in 30, while the most conservative this estimator could be would be $c_\alpha = 0$ which reduces our estimator to $\mathrm{s}^2(m)$ and simply uses the standard deviation of the noisy privacy preserving bootstrapped releases as the standard error. Values $k-1 > c_{\alpha'} > 0$, correspond to other values of α' as given by equation 32.

7.3 Confidence Intervals

While we are releasing k draws of m, our best estimator for the mean is $m^* = \bar{m}$, which has its own variance. This can be estimated as:

$$\hat{\sigma}^{2\star}(m^*, c_{\alpha'}) = \hat{\sigma}^{2\star}(f, c_{\alpha'}) + \sigma^2(n|\rho) = \left[\mathrm{s}^2(m) - \frac{\Delta^2 k}{2\rho}\frac{c_{\alpha'}}{k-1}\right] + \frac{\Delta^2}{2\rho} = \mathrm{s}^2(m) - \frac{\Delta^2}{2\rho}\left(\frac{k c_{\alpha'}}{k-1} - 1\right) \tag{35}$$

Since m^* is composed of $\bar{f}$ which is asymptotically Normal, and n which is exactly Normal, then m^* is asymptotically Normal and we can use a conventional confidence interval of:

$$CI_{1-\alpha}(m^*) = m^* \pm z_{\frac{\alpha}{2}}\sqrt{\hat{\sigma}^{2\star}(m^*, c_{\alpha'})} \tag{36}$$

Where again, α sets the desired coverage of the confidence interval and α' sets our conservative parameter, that is, how often our estimate of $\sigma^2(f)$ may be under reported; increasing values of α' lead to decreasing values of $c_{\alpha'}$ and wider confidence intervals.

We turn now to compare the coverage performance at three specifications: [1] our *unbiased* estimator with $c_{\alpha'} = k-1$, [2] a *conservative* estimator where $\alpha' = 0.05$, and [3] a *most conservative* estimator where $\alpha' = 0$ ($\Rightarrow c_{\alpha'} = 0$). We simulate k bootstrapped samples from a sample of size 500, and on each release a differentially private mean using the Gaussian mechanism with privacy loss under zCDP of ρ/k. In Figure 6 we show one example of the distribution of $k = 50$ bootstraps, each using $\rho = 0.5/50$. The true population mean is shown as the red dashed line at zero, and the true sample mean the red solid line. The average of the bootstraps, m^* is shown in blue, along with the confidence interval we can generate using equation using the unbiased variant. Here in this exemplar both the true population mean and the sample mean are contained within the confidence interval.

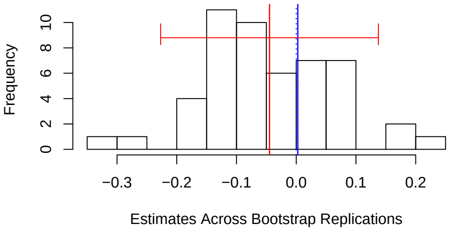
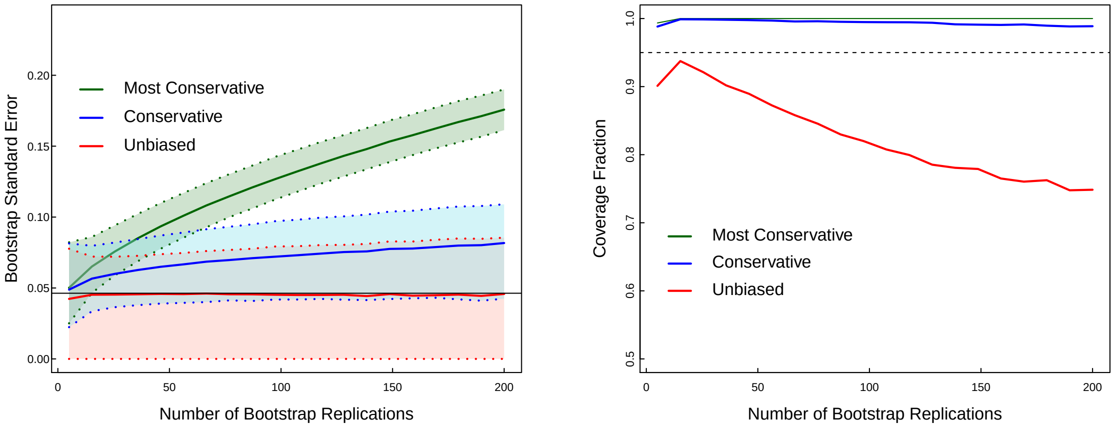


Figure 6: Simulation of fifty differentially private bootstrap means with resulting confidence interval (red) and true population value (blue).

In the left in Figure 7a, we show the estimates of the standard error across 500 simulations at variable numbers of private bootstrap replications, for our unbiased, conservative, and most conservative confidence intervals. The dotted lines represent the 90% range over the simulations. The horizontal blue line is the true standard error of the mean. We see the unbiased estimator hits this line on average, but often gives standard errors far too small. At approximately 50 bootstrap replications, we observe that the conservative and most conservative estimates are never smaller than the true value, thus any CI's generated would not have coverage that was too small.



(a) (b)

Figure 7: *Private bootstrap estimate of the mean and its standard error.*

In the right in Figure 7b we show the coverage of the three confidence intervals, that is, the fraction of the time the simulated confidence interval contains the true value. Here we depict 95 percent confidence intervals. The interval constructed with our unbiased estimator covers the true value substantially less than 95 percent of the time and this grows worse as the number of bootstraps increases. However our conservative confidence interval has greater than 95 percent coverage (as does the most conservative variant). The conservative interval, in this example for a mean of 500 observations is around 1.5 times larger than the unbiased variant, so we see that greater than promised coverage can be obtained for a modest increase in the confidence interval span.

8 Example: Releasing Mean Age from US Census Data

We demonstrate the privacy-preserving bootstrap by estimating the mean of 500 observations sampled from the US Census Public Use Micro Sample (PUMS) 5 percent file for California. We use the variable *age* clipped to the interval 0 to 100. We use $\rho = 0.50$ and perform the algorithm with 50 bootstrap replications, meaning that the privacy preserving mean learned from any one bootstrap sample is obtained with $\rho = 0.01$.

In Figure 8 we create 1000 "oneshot" releases of the mean of age using the full $\rho = 0.50$ and compare the distribution to 1000 simulations of the bootstrap averaged answer (again of 50 bootstraps in each simulation) to show visually our key result from section 6.4 that the distributions of the released answer are the same, for the same privacy loss parameter, regardless of whether the bootstrap or the common one-shot approach is employed.

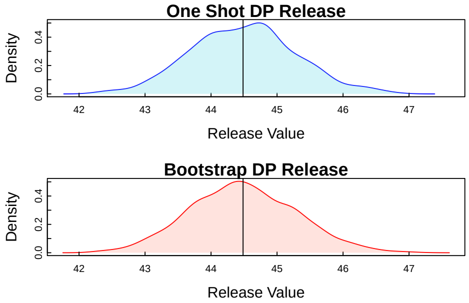
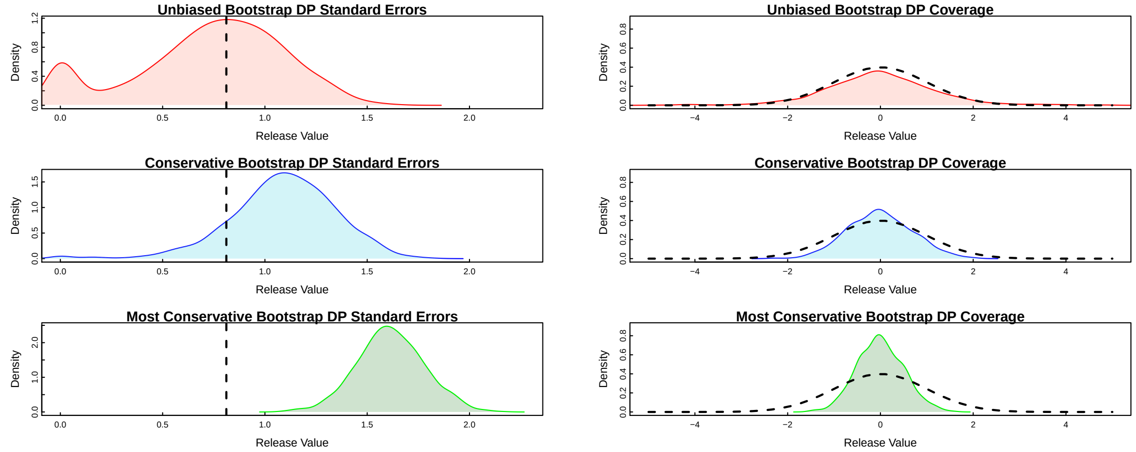


Figure 8: *Distribution of Bootstrap generated releases against simple one-shot releases.*

However, crucially, the bootstrap also allows us to estimate the standard error in this mean, and thus derive confidence intervals. In Figure 9a we show the distribution of standard errors, that is, $\sqrt{\hat{\sigma}^{2\star}(m^*, c_{\alpha'})}$, for our [1] *unbiased* estimator ($c_{\alpha'} = k-1$), [2] *conservative* estimator ($\alpha' = 0.05$), and [3] a *most conservative* estimator ($\alpha' = 0$). The "true" standard error[6] is shown as the vertical dashed line. We see the unbiased

[6] Computed using the variance of the *age* variable in all the PUMS California observations and the known variance of the

standard error (in red) often has standard error below this value. There is even a mode of observations at zero where the right term in equation 30 is greater than the left term. The density of the conservative estimator (blue) is shifted to the right, as we are being more conservative in how much observed variance in the bootstrap distribution we are attributing to the differentially private mechanism. The most conservative estimate (green) shifts this further still.



(a) (b)

Figure 9: *Distribution of DP mean releases from oneshot release and from the bootstrap release.*

In Figure 9b we depict the z-scores of "true" mean in the PUMS California dataset, against differentially private estimates generated from samples of size 500 and privacy loss parameter $\rho = 0.5$. That is, if m^T is the mean in the dataset, from which a sample of 500 is taken, and then those 500 observations are bootstrapped to create 50 datasets from which the bootstrap average m^* and standard error are computed, then the z-score:

$$z = \frac{m^* - m^T}{\sqrt{\hat{\sigma}^{2\star}(m^*, c_{\alpha'})}} \tag{37}$$

tells us how many standard errors the true value is from the estimated value, or in other words, how many standard errors wide our confidence interval would have to be to contain the true value. If for example, $z > 1.96$ then a 95% confidence interval using this estimate and standard error would not contain the true value. We plot the entire distribution of z-scores for 1000 simulations against the standard normal as a reference distribution. If the z-score distribution matched the reference distribution, any confidence intervals, at any desired level of confidence, would have exactly the right coverage. If the distribution is wider than the reference distribution we have lower than desired coverage, and if more condensed than the reference we have conservative coverage. For the unbiased distribution, the small standard errors we previously saw, are creating more mass in the tails (more high valued z-scores) than a standard normal. For example, for 95% confidence intervals constructed using the unbiased standard errors, only 80.1% of the confidence intervals contain the true value, so we have improper coverage.

In contrast, the conservative z-values are slightly more condensed than the reference distribution, so the confidence intervals created will be slightly larger than needed. In this example, across the 1000 simulations the 95% confidence interval contained the true value 97.1% of the time. The condensing of the z-scores is can be seen to be even more profound for the most conservative estimator. Here the 95% confidence intervals contain the true value 100% of the time.

Gaussian noise in the differentially private mechanism.

9 Conclusion

Differential Privacy makes releases of queries on sensitive datasets available to analysts, with formal bounds on the privacy loss from the query. However, often an analyst is not interested in the exact value in the sample data, but rather an inference back to the true value in the population. Confidence intervals, and other related techniques for inference, typically require a measure of uncertainty in the central estimate, such as the variance/standard error. If we see this as a separate function to be estimated on the data, a conventional differentially private approach would require us to partition the privacy loss parameter, to spend some on this on the calculation of uncertainty[7] and accordingly result in a more noisy release of the original quantity of interest.

Instead, here we show that the variance/standard error can be obtained by post processing on the original function computed on bootstrapped datasets. Under zCDP, this results in an answer that has no utility loss compared to the one-shot approach, but gives a standard error "for free", that is, at no cost to the privacy parameter, and from which we can construct confidence intervals with conservative coverage. Because we are only spending the privacy budget on the original function of interest (albeit divided across bootstrapped datasets), and deriving the uncertainty by post processing, we can obtain not just the differentially private release, but a valid confidence interval for that value.

References

[1] Micah Altman, Alexandra Wood, David R OBrien, and Urs Gasser. Practical approaches to big data privacy over time. *International Data Privacy Law*, 2018.

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

[3] Mercè Crosas. The dataverse network: An open-source application for sharing, discovering and preserving data. *D-Lib Magazine*, 17:1–2, 2011. doi:1045/january2011-crosas.

[4] Merce Crosas, Gary King, James Honaker, and Latanya Sweeney. Automating open science for big data. *The ANNALS of the American Academy of Political and Social Science*, 659(1):260–273, 2015.

[5] Aref N. Dajani, Amy D. Lauger, Phyllis E. Singer, Daniel Kifer, Jerome P. Reiter, Ashwin Machanavajjhala, Simson L. Garfinkel, Scot A. Dahl, Matthew Graham, Vishesh Karwa, Hang Kim, Philip Leclerc, Ian M. Schumutte, William N. Sexton, Lars Vilhuber, and John M. Abowd. The modernization of statistical disclosure limitation at the U.S. Census Bureau.

[6] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proc. 22^{nd} PODS ACM*, pages 202–210. ACM, 2003.

[7] Vito D'Orazio, James Honaker, and Gary King. Differential privacy for social science inference. 2015.

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In 3^{rd} *Theory of Crypt. Conf.*, pages 265–284, 2006.

[9] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[10] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 650–669. IEEE, 2015.

[11] Bradley Efron and Robert J Tibshirani. *An Introduction to the Bootstrap.* CRC press, 1994.

[7] For which there are few existing mechanisms in the literature.

[12] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[14] Marco Gaboardi, James Honaker, Gary King, Jack Murtagh, Kobbi Nissim, Jonathan Ullman, and Salil Vadhan. Psi ({\Psi}): a private data sharing interface. *arXiv preprint arXiv:1609.04340*, 2016.

[15] Marco Gaboardi, Hyun-Woo Lim, Ryan M Rogers, and Salil P Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*. JMLR, 2016.

[16] Andy Greenberg. Apples differential privacy is about collecting your databut not your data. *Wired, June*, 2016.

[17] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genetics*, 4(8), 2008.

[18] James Honaker. Efficient use of differentially private binary trees. 2015.

[19] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Vertica*, 1:1000.

[20] Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

[21] Gary King. An introduction to the dataverse network as an infrastructure for data sharing. *Sociological Methods and Research*, 36:173–199, 2007.

[22] Kobbi Nissim, Thomas Steinke, Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, David R OBrien, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Working Group Privacy Tools Sharing Res. Data, Harvard Univ., Boston, MA, USA, Tech. Rep. TR-2017-03*, 2017.