Unbiased Statistical Estimation and Valid Confidence Intervals Under Differential Privacy

by

Christian Covington

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We present a method for producing unbiased parameter estimates and valid confidence intervals under the constraints of differential privacy, a formal framework for limiting individual information leakage from sensitive data. Prior work in this area is limited in that it is tailored to calculating confidence intervals for specific statistical procedures, such as mean estimation or simple linear regression. While other recent work can produce confidence intervals for more general sets of procedures, they either yield only approximately unbiased estimates, are designed for one-dimensional outputs, or assume significant user knowledge about the data-generating distribution. Our method induces distributions of mean and covariance estimates via the bag of little bootstraps (BLB) [26] and uses them to privately estimate the parameters' sampling distribution via a generalized version of the CoinPress estimation algorithm [6]. If the user can bound the parameters of the BLB-induced parameters and provide heavier-tailed families, the algorithm produces unbiased parameter estimates and valid confidence intervals which hold with arbitrarily high probability. These results hold in high dimensions and for any estimation procedure which behaves nicely under the bootstrap.

Acknowledgements

I would like to thank my academic advisors, Xi He and Gautam Kamath, for both their mentorship and their willingness to let me explore a variety of topics over the past two years, members of the Harvard Privacy Tools Project for introducing me to differential privacy and helping shape my research interests, and my family and partner for their personal and intellectual support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.0.1 Overview

Our society has experienced dramatic growth in large-scale data collection and analysis in recent history, which has led to a number of concerns around the role of privacy and security in the modern world. Our particular focus will be on statistical analysis and how it can leak information about individuals in the data set being analyzed.

Statistical agencies, in particular, have been concerned with what they called *statistical disclosure limitation*, using a variety of methods in an attempt to limit disclosure risk. [16, 17, 27], and [34] work on identifying and quantifying disclosure risk under different assumptions. There has also been substantial work developing disclosure limitation methods such as k-anonymity [40], t-closeness [30], l-diversity [31], and swapping [23], among others.

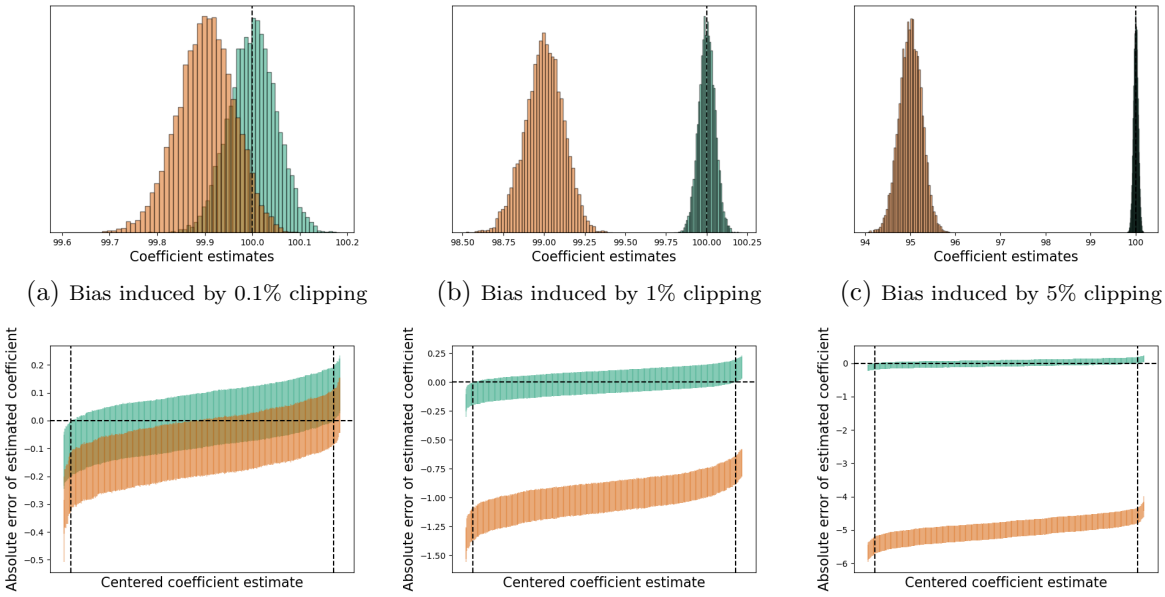
[12] developed a polynomial data reconstruction algorithm and used it to prove a result later coined in [20] as the *Fundamental Law of Information Recovery*. Roughly, this law states that an attacker can reconstruct a data set by asking a sufficiently large number of cleverly chosen queries of the data, even if the query answers are noised before being returned to the attacker. This inspired the invention of *differential privacy* (DP) [19]. DP is a definition which requires that, for any two data sets differing in one row, the distribution of answers to any possible query doesn't change much between the two data sets.

DP has become a popular tool in many corners of industry but has not been widely applied to research in many fields that often analyze sensitive data (social sciences, medicine, etc.). We suggest that this is, in part, because of a lack of DP algorithms that effectively meet the needs of these fields. First, DP algorithms typically require the user to, *without looking at the data*, specify bounds to which the data will be censored/clipped. We claim that doing this well is a difficult problem in general, and potentially introduces substantial error into the DP pipeline which is difficult to account for. Second, DP estimators do not typically admit basic statistical guarantees that many applied researchers desire; namely unbiasedness and valid confidence intervals. Our goal is to provide a framework for converting non-private estimators to DP estimators in a way that jointly addresses both of these concerns.

1.0.2 Problem Demonstration

DP estimators are not, in general, unbiased with respect to the non-private estimator they are attempting to approximate, nor do they typically admit valid confidence sets. Although many DP mechanisms take the form "non-private statistic plus zero-mean noise", this obscures the fact that these mechanisms typically require that the input data reside in a bounded data domain. In practice, analysts are required to specify this domain without looking at the data, which introduce a potential trade-off in the analyst's decision calculus. The analyst generally wants to find tight bounds, as tighter bounds typically require less noise addition to guarantee privacy. However, if the analyst's bounds are too tight, they risk censoring the data and biasing statistics they calculate.

Consider the case where $X = \{X_1, \ldots, X_n\}$ with $X_i \sim N(0, 1)$ and $y = X\beta + \epsilon$ for $\beta = 100, \epsilon \sim N(0, 10^2)$. We estimate β and get associated 95% confidence intervals using OLS and test the effect of various levels of censoring on the estimates and confidence intervals. Specifically, we leave X uncensored and censor the top $\{0, 0.1, 1, 5\}$ percent of y. Note that we will often use the term "clipping" in place of "censoring", as it is the more familiar term within the DP literature.



(a) Bias induced by 0.1% clipping (b) Bias induced by 1% clipping (c) Bias induced by 5% clipping

(d) CI coverage induced by 0.1% clipping (e) CI coverage induced by 1% clipping (f) CI coverage induced by 5% clipping

Figure 1.1: Distribution of OLS coefficient estimates (a-c) and 95% confidence intervals (d-f) under different levels of clipping of y. Non-clipped distribution in green, clipped distribution in orange.

Figure 1.1 shows results from 10,000 simulations of this process. The top plots show coefficient distributions under each level of clipping and compare it to the condition of no clipping. Note that at even a moderate level of 1% clipping, the distributions of estimates are completely non-overlapping. The bottom plots show the estimates, arranged in increasing order on the x-axis, with vertical bars representing the 95% confidence interval for that estimate. Each estimate is centered as if the true coefficient value were 0. The black dotted vertical lines on the left and right sides of each plot show the 0.025 and 0.975

quantiles, where we expect perfectly calibrated confidence intervals to cross the x-axis. At 0.1% clipping, approximately half of our confidence intervals do not contain the true parameter value; at higher levels of clipping, none of them do.

We ultimately want to construct confidence intervals for our private estimator rather than the non-private one, which introduces extra complexity not represented in Figure 1.1. At a high-level, our private estimator has variance from two sources; the inherent variance of the non-private estimator, which we privately estimate, and the noise we add for privacy, which we often know analytically for DP mechanisms. We need to take both into account for valid confidence intervals. Even if we don't clip any data points, a private estimate of the variance of the non-private estimator (privatized with symmetric, zero-mean noise) will be smaller than its non-private counterpart with probability $\frac{1}{2}$. Plugging in a variance estimated in this way will lead to overly narrow confidence intervals, so we need to ensure that we can produce a private variance estimate that is at least as large as its non-private equivalent.

It's worth pointing out that we chose a very simple regime for the experiments above; one-dimensional OLS with a Gaussian covariate, Gaussian error, clipping in only the outcome variable, and no attempt to respect DP. In more complex settings, the effect of clipping on the coefficients could be worse, and would certainly be harder to predict and reason about.

1.0.3 Contributions

We introduce a general-purpose meta-algorithm that allows an analyst to take any of a broad set of estimators that are unbiased in the non-private setting and produce a differentially private version that, with high probability, is unbiased and produces valid confidence intervals. In Section 2, we introduce our framework which takes its general structure from the Sample-Aggregate approach of [32]. We use the Bag of Little Bootstraps (BLB) algorithm (Algorithm 3 [26]) to approximate the sampling distribution of the non-private estimator and privately estimate the parameters of said distribution using a modified version of the CoinPress mean estimation algorithm (Algorithm 2 [6]). Under a few assumptions on the behavior of the bootstrap and initial user bounds, our use of CoinPress produces unbiased parameter estimates (Theorem 3.0.6). The estimated parameters and knowledge of the privatization process are combined to generate final private mean and covariance estimates (Theorems 4.0.2, 4.0.3, and 4.0.4), and valid confidence intervals (Theorem 4.0.5, and Corollary 4.0.6)). We credit [21] for first noting the compatibility of BLB with Sample-Aggregate.

These guarantees hold under the following assumptions. First, the BLB applied to the estimator must approximate the sampling distribution of the estimator well (Assumption 2.0.1). Then, for both the mean and covariance distribution induced by the BLB, the analyst must provide a distribution that is heavier-tailed (Definition 3.0.1 and Assumption 3.0.2), as well as give bounds on the mean (Assumption 3.0.3) and covariance (Assumption 3.0.4) of the induced distribution. We argue that these conditions are rather natural, and we demonstrate how the properties of the CoinPress algorithm allow the analyst to get good performance even when they set the aforementioned bounds very conservatively.

We employ a multivariate extension of the precision-weighting technique to improve CoinPress' estimates (Theorem 4.0.1). While precision-weighting is a well-known technique in the meta-analysis literature, we give, to the best of our knowledge, the first proof of multivariate optimality, which may be of independent interest.

We believe that our framework is a promising step toward making differential privacy more practical for applied research. First, we believe that the problem of choosing good data bounds is significant in practice in that it is both generally difficult and that many DP algorithms are sensitive to poor choices. There is also currently an asymmetry in the failure modes, in that bounds that are too wide typically yield answers which are unbiased but very noisy, while bounds that are too narrow risk "silent failure", where the DP result looks precise but is not actually representative of the non-private answer. Our framework, through use of the CoinPress algorithm, gives users more leeway to err on the side of conservatism, thus mitigating the possibility of getting DP results that appear precise but are systematically incorrect.

Second, our framework is general enough to be applied to any estimators for which the BLB does a "good" job estimating its sampling distribution. The bootstrap is broadly familiar to applied statisticians, and its properties for any particular estimator an analyst wishes to use can be tested on non-sensitive data. Thus, answering the question of whether or not our algorithm will be useful for a given problem can be done without much knowledge about DP.

1.0.4 Related Work

Differential privacy has grown in popularity in recent years, as has the literature exploring the intersection between statistics and DP. [18] point out how a handful of common robust statistical estimators could be extended to respect differential privacy. [46] compare DP mechanisms via convergence rates of distributions and densities from DP releases and frame DP in statistical language more broadly. [29] explores model selection under DP, while [33] explores model uncertainty. [42, 45, 24, 9], and [3] propose methods for DP hypothesis testing in various domains. [37, 43, 4], and [1] all address the problem of differentially private linear regression.

[25] gives nearly optimal confidence intervals for Gaussian mean estimation with finite sample guarantees. [15] proposes their own algorithms for the same problem and finds superior practical performance in some domains, and [6] develop an algorithm that works well at reasonably small sample sizes and without strong assumptions on user knowledge, while also scaling well to high dimensions. [14] explores non-parametric confidence intervals for calculating medians. [13] gives confidence intervals for a difference of means. [2] shows how to construct DP version of M-estimators, as well as associated confidence regions.

Our work continues in a line of recent work for constructing confidence sets for more general classes of differentially private estimators. [7] shows how to combine estimates from additive functions that respect zCDP to get confidence intervals at no additional cost. [44] provides confidence intervals for models trained with objective or output perturbation algorithms. These algorithms are quite general, but require solving the non-private ERM sub-problem optimally. [22] presents a very general approach based on privately estimating parameters of the data-generating distribution and bootstrapping confidence intervals by

repeatedly running the model of interest on samples from a distribution parameterized by the privately estimated parameters. This method is efficient with respect to its use of the privacy budget, but relies on significant knowledge of the structure of the data-generating process. With the exception of [25], these works either ignore the issue of bounding $\mathcal{X}$ effectively, or attempt to address it through bias-correction strategies which we believe are unlikely to work for complex problems. [5] tests a variety of DP algorithms for various tasks (ranging from tabular statistics to OLS regression) on real data and argue that existing methods for performing DP regression, "would struggle to produce accurate regression estimates and confidence intervals" ([5], p. 1).

[21] is the closest existing work to ours. They also start with Sample-Aggregate framework, note the similarity of Sample-Aggregate with the BLB algorithm, and use this to approximate the sampling distribution of the non-private estimator. The k estimates are then aggregated via a differentially private mean and confidence intervals are calculated using a differentially private variance estimate and CLT assumption. Because the estimates are projected into a bounded data domain to control the sensitivity of the mean, the resulting private mean could potentially be biased. [21] attempts to address this issue by privately estimating the proportion of the k estimates that are clipped by the projection and adjusting the private mean by the estimated clipping proportions. This method has the advantage of allowing users to specify overly tight clipping bounds in order to decrease the global sensitivity of their estimator, but is sensitive to how well the clipping proportions are estimated and, to our knowledge, has no multi-dimensional analogue.

1.0.5 Definitions

Differential Privacy

We begin with an introduction to the core definitions of DP.

Definition 1.0.1 (Neighboring data sets)**.** *Let $\mathcal{X}$ be a data universe and $D, D' \in \mathcal{X}^n$. We say that D, D' are neighboring if $\max(|D \setminus D'|, |D' \setminus D|) = 1$. We also define the set of all neighboring data sets as $\mathcal{D}_n = \{(D, D') \in \mathcal{X}^n \times \mathcal{X}^n : \max(|D \setminus D'|, |D' \setminus D|) = 1\}$.*

Definition 1.0.2 (Rényi divergence [35])**.** *Let P, Q be probability measures over a measurable space (Ω, Σ). Then we define the α-Rényi divergence between P, Q as*

$$H_\alpha(P\|Q) = \frac{1}{\alpha - 1} \ln \int_\Omega P(x)^\alpha Q(x)^{1-\alpha} dx.$$

Definition 1.0.3 (Zero-concentrated differential privacy (zCDP) [8])**.** *Let $\mathcal{M} : \mathcal{X}^n \to \Omega$ be a randomized algorithm where $(\Omega, \Sigma, \mathbb{P})$ is a probability space and $\rho \geq 0$. We say that $\mathcal{M}$ respects ρ-zCDP if*

$$\forall (D, D') \in \mathcal{D}_n, \forall \alpha \in (1, \infty) : H_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \leq \rho\alpha.$$

The parameter ρ represents an upper bound on the amount of information $\mathcal{M}$ leaks about the underlying data. Larger ρ implies more information leakage, or *privacy loss*, but also allows for the statistics returned by $\mathcal{M}$ to be more accurate.

Differential privacy has a few properties that will be useful for us later.

Lemma 1.0.4 (Composition of zCDP [8]). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ and $\mathcal{M}' : \mathcal{X}^n \to \mathcal{Z}$ such that $\mathcal{M}$ satisfies ρ-zCDP and $\mathcal{M}'$ satisfies ρ'-zCDP. Define $\mathcal{M}'' : \mathcal{X}^n \to \mathcal{Y} \times \mathcal{Z}$ by $\mathcal{M}''(x) = (\mathcal{M}(x), \mathcal{M}'(x))$. Then $\mathcal{M}''$ satisfies $(\rho + \rho')$-zCDP.*

Lemma 1.0.5 (Postprocessing of zCDP [8]). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ and $f : \mathcal{Y} \to \mathcal{Z}$ such that $\mathcal{M}$ satisfies ρ-zCDP. Define $\mathcal{M}' : \mathcal{X}^n \to \mathcal{Z}$ such that $\mathcal{M}'(x) = f(\mathcal{M}(x))$. Then $\mathcal{M}'$ satisfies ρ-zCDP.*

Definition 1.0.6 (Global Function Sensitivity). *Let $\mathcal{X}$ be a data domain, $\gamma : \mathcal{X}^n \to \mathbb{R}^d$, and $\mathcal{D}_n$ be the set of all neighboring data sets as in Definition 1.0.1. Then we write the global sensitivity of γ with respect to a distance metric d as $GS_d(\mathcal{X}^n, \gamma) = \max_{D,D' \in \mathcal{D}_n} d(\gamma(D), \gamma(D'))$.*

Algorithms can be made to respect DP in a variety of ways, but the most common way (as well as the approach we use in this work) is via an *additive noise mechanism*. This just entails running the algorithm as one would normally, and then adding random noise scaled relative to the algorithm's sensitivity.

Throughout this work, we use a popular additive noise mechanism called the *Gaussian mechanism*.

Lemma 1.0.7 (Gaussian Mechanism). *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ have global ℓ_2 sensitivity $GS_{\ell_2}(\mathcal{X}^n, f)$. Then the Gaussian mechanism*

$$\mathcal{M}_f(D) = f(D) + N\left(0, \left(\frac{GS_{\ell_2}(\mathcal{X}^n, f)}{\sqrt{2\rho}}\right)^2 I_d\right)$$

satisfies ρ-zCDP.

Note that it is often necessary to bound the data domain $\mathcal{X}$ to ensure that $GS_{\ell_2}(\mathcal{X}^n, f) < \infty$. For example, let $\mathcal{X} = \mathbb{R}$, $D = (D_1, \ldots, D_n)$ with $D_i \in \mathbb{R}$, and $f : \mathbb{R}^n \to \mathbb{R}$ be such that $f(D) = \frac{1}{n}\sum_{i=1}^n D_i$. If we let $D' = (\infty, D_2, \ldots, D_n)$, then D, D' are neighbors (they differ only in the first element), but $\|f(D) - f(D')\|_2 = \infty$. If instead $\mathcal{X} = [0, 1]$, then the D, D' that induce the largest difference in f are $D = (1, D_2, \ldots, D_n)$ and $D' = (0, D_2, \ldots, D_n)$. In this scenario, $GS_{\ell_2}(\mathcal{X}^n, f) = \frac{1}{n}$. These bounds must be set without looking at the particular D_i, and are generally chosen by a data analyst based on public metadata and/or their beliefs about the data-generating process.

Statistical Inference

This need to bound $\mathcal{X}$ introduces complications for doing statistical inference under DP, while maintaining the types of guarantees we often want from non-private estimators. We focus specifically on unbiased estimators and valid confidence sets.

Definition 1.0.8 (Unbiased Estimator). *Let $\theta \in \mathbb{R}^d$ be a model parameter we wish to estimate. We collect data $D \sim \mathcal{D}$ and estimate θ with a random variable $\hat{\theta} : \mathcal{D} \to \mathbb{R}^m$. We say that $\hat{\theta}$ is an unbiased estimator of θ if*

$$\mathbb{E}\left(\hat{\theta}(D)\right) = \theta,$$

with randomness taken over the sampling of $D \sim \mathcal{D}$, as well as any other randomness in $\hat{\theta}$.

Many applied statisticians, particularly those interested in estimating causal effects using linear models, prize unbiased parameter estimation and are willing to sacrifice on other fronts to achieve it. For example, the standard OLS estimator (which is the minimum-variance unbiased estimator under the assumptions of the Gauss-Markov theorem) is used for estimating parameters of a linear regression model in favor of other biased estimators, such as the James-Stein estimator [38, 39], which dominate it in terms of ℓ_2 error of the parameter estimates.

Definition 1.0.9 (Confidence Set)**.** *Let $\theta \in \mathbb{R}^d$ be a model parameter we wish to estimate using data $D \sim \mathcal{D}$. For arbitrary $\alpha \in [0, 1]$, a $(1-\alpha)$-level confidence set for θ is a random set $S \subseteq \mathbb{R}^d$ such that*

$$\mathbb{P}(\theta \in S) = 1 - \alpha,$$

with randomness taken from the sampling of D and any other randomness in the construction of S.

Ideally, we would be able to find a perfectly-calibrated confidence set, where the coverage probability (i.e. $\mathbb{P}(\theta \in S)$) is exactly $1 - \alpha$. However, this is often impossible to compute exactly and so practitioners tend to default to being overly conservative instead. In this setting, we require $\mathbb{P}(\theta \in S) \geq 1 - \alpha$ and call S a *valid confidence set*. In this work, we will focus on *confidence intervals*, which are one-dimensional and contiguous confidence sets.

We can simplify the general problem of constructing confidence sets by restricting our attention to estimators for which each marginal belongs to a symmetric location-scale family.

Definition 1.0.10 (Location-Scale Family)**.** *A set of probability distributions $\mathcal{F}$ is a location-scale family if for any distribution function $F \in \mathcal{F}$, the distribution function $F'(x) = F(\mu + \sigma x) \in \mathcal{F}$ for any $\mu \in \mathbb{R}, \sigma > 0$.*

Our restriction to location-scale families ensures that estimating the mean and (co)variance of the estimator is sufficient for constructing confidence intervals.

Chapter 2

Algorithm Overview

We present a high-level overview of our method in Algorithm 1, eliding some parameters from the subroutines (which we replace with ...) to make the presentation simpler. Let $\xi(a_1, \ldots, a_k) = \frac{1}{k}\sum_{i=1}^{k} a_i$ and $\xi'(a_1, \ldots, a_k) = Cov\left(\{a_i\}_{i\in[k]}\right)$.

Algorithm 1 General Valid DP (GVDP)

Input: data set $X \in \mathbb{R}^{n\times m}$, estimator $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$ families of distributions $Q_{\tilde{\theta}}, Q_{\tilde{\Sigma}}$, privacy budgets $\rho^{\tilde{\theta}}, \rho^{\tilde{\Sigma}} > 0$

Output: parameter estimate $\tilde{\theta}$ and associated confidence intervals/set which respect $\left(\rho^{\tilde{\theta}} + \rho^{\tilde{\Sigma}}\right)$-zCDP

procedure GVDP$(X, \hat{\theta}, Q_{\tilde{\theta}}, Q_{\tilde{\Sigma}}, \rho^{\tilde{\theta}}, \rho^{\tilde{\Sigma}})$

  $\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]} = \text{BLB}\left(X, \hat{\theta}, \xi', \ldots\right)$ ▷ Algorithm 3 – get estimates of parameter covariance

  $\{\hat{\theta}_i^{BLB}\}_{i\in[k]} = \text{BLB}\left(X, \hat{\theta}, \xi, \ldots\right)$ ▷ Algorithm 3 – get estimates of parameter means

  $\{\tilde{\Sigma}_m\}_{m\in[t]} = \text{MVMREC}\left(\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]}, \ldots, Q_{\tilde{\Sigma}}, \ldots, \rho^{\tilde{\Sigma}}, \ldots\right)$ ▷ Algorithm 2 – privately estimate parameter covariance

  $\{\tilde{\theta}_m\}_{m\in[t]} = \text{MVMREC}\left(\{\hat{\theta}_i^{BLB}\}_{i\in[k]}, \ldots, Q_{\tilde{\theta}}, \ldots, \rho^{\tilde{\theta}}, \ldots\right)$ ▷ Algorithm 2 – privately estimate parameter means

  Combine $\{\tilde{\Sigma}_m\}_{m\in[t]}$ to get $\tilde{\Sigma}$ ▷ Theorem 4.0.1

  $\tilde{\Sigma} = \text{PSDPROJECTION}(\tilde{\Sigma})$ ▷ Algorithm 5 – ensure $\tilde{\Sigma}$ is PSD

  Combine $\{\tilde{\theta}_m\}_{m\in[t]}$ to get $\tilde{\theta}$ ▷ Theorem 4.0.1

  Use $\tilde{\theta}, \tilde{\Sigma}$, and $Q_{\tilde{\theta}}$ to get confidence intervals/set S. ▷ E.g. Theorem 4.0.5

  return $\{\tilde{\theta}, S\}$

Let $\mathcal{X}$ be our data universe, $\mathcal{D}$ a distribution over the universe, and $X = \{x_1, \ldots, x_n\}$ where x_i are drawn i.i.d. from $\mathcal{D}$. For shorthand, we say that $X \sim \mathcal{D}^n$. We say that the analyst wants to run some model over the data, which has an associated parameter vector $\theta \in \mathbb{R}^d$. The analyst specifies the estimator they would have liked to run in the non-private setting $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$.

As stated earlier, differentially private algorithms typically require specification of the global sensitivity of the function whose outputs are being privatized (see Definition 1.0.6 and Lemma 1.0.7). This can become arbitrarily complex for complicated models even assuming a bounded input domain. The Sample-Aggregate paradigm introduced in [32] suggests that one way to deal with this is to run the algorithm of interest non-privately over k disjoint subsets of the data, bound the outputs, and then aggregate the results using

a function with a sensitivity that is easier to reason about (e.g. the mean). Because each element in the original data contributes to one subset of the partition, its effect on the aggregation is localized to one of its k inputs and we can treat this just like privatizing the mean of a data set with k elements.

In that vein, our algorithm begins (lines 2 and 3 of Algorithm 1) by randomly partitioning the data X into k disjoint subsets $\{X_1, \ldots, X_k\}$, with k chosen by the analyst and applying the BLB algorithm over the partition. On each subset X_i, the BLB algorithm scales the subset back up to size n and runs $\hat{\theta}$ r times, producing estimates $\{\hat{\theta}_{i,a}\}_{a\in[r]}$. It then aggregates these into an arbitrary assessment of estimator quality. For our purposes we use the mean and covariance, so for each $i \in [k]$ we get

$$\hat{\theta}_i^{BLB} = \frac{1}{r}\sum_{a=1}^{r} \hat{\theta}_{i,a}$$
$$\hat{\Sigma}_i^{BLB} = Cov\left(\{\hat{\theta}_{i,a}\}_{a\in[r]}\right).$$

These sets of estimates are now empirical approximations to the parameter distributions induced by BLB, which we assume are good approximations of the actual sampling distributions. We make this last assumption explicit as follows.

Assumption 2.0.1. *Let the estimator $\hat{\theta} \sim G(\theta, \Sigma)$ where the marginals of G each belong to a location-scale family. For $\hat{\theta}_i^{BLB}$ and $\hat{\Sigma}_i^{BLB}$ generated by applying BLB to our estimator $\hat{\theta}$, let $\hat{\theta}^{BLB} = \frac{1}{k}\sum_{i=1}^{k} \hat{\theta}_i^{BLB}$ and $\hat{\Sigma}^{BLB} = \frac{1}{k}\sum_{i=1}^{k} \hat{\Sigma}_i^{BLB}$. We assume that $\mathbb{E}\left(\hat{\theta}^{BLB}\right) = \mathbb{E}\left(\hat{\theta}\right) = \theta$ and $\mathbb{P}\left(\hat{\Sigma}^{BLB} \succeq \hat{\Sigma}\right) = 1$.*

For the rest of this work, our goal is going to be to privately estimate $\hat{\theta}^{BLB}$ and $\hat{\Sigma}^{BLB}$ and argue that these allow us to get unbiased estimates of $\hat{\theta}$ and valid confidence intervals. Assumption 2.0.1 provides the link that lets us do this. We are going to construct a private estimator that is unbiased with respect to $\hat{\theta}_{BLB}$, which in turn gives us an unbiased estimator of $\hat{\theta}$. Likewise, we are going to create a private covariance estimate $\tilde{\Sigma}$ such that $\tilde{\Sigma} \succeq \hat{\Sigma}_{BLB}$. By transitivity of the Löwner order, Assumption 2.0.1 also then implies that $\tilde{\Sigma} \succeq \hat{\Sigma}$.

The Löwner condition on $\hat{\Sigma}^{BLB}$ is potentially onerous (especially in high dimensions) and, frankly, unlikely to hold. In practice however, this condition can be dropped at the cost of a bit of extra fuzziness in the results. As stated above, we later make claims about our private estimator relative to $\hat{\Sigma}^{BLB}$, which under Assumption 2.0.1 also hold relative to $\hat{\Sigma}$. This generalization to $\hat{\Sigma}$ is a higher bar than is typically set in applications of the bootstrap, where the bootstrap approximation is simply treated as a "good-enough" approximation of the sampling distribution.

At this point we use the CoinPress mean estimation algorithm (lines 4 and 5) to generate private estimates of the mean of each empirical distribution. CoinPress assumes that the analyst has given a heavier-tailed distribution than the distribution it is estimating, as well as bounds on the mean and covariance of the distribution.

CoinPress then procedes by iteratively privately estimating the mean of the BLB distribution (Algorithm 4 in the appendix) and using the updated estimate to tighten the

bounds on the mean. At each step, CoinPress has access to bounds that, with high probability, contain the true mean of the distribution. It takes these bounds and pushes them out based on the supposed distribution; the idea being that if you have a distribution of a known family with bounded mean and covariance, you can with high probability upper bound the ℓ_2 norm of an arbitrary number of draws from said distribution. That is, we set bounds that with high probability do not clip any of the k estimates from the BLB distribution. The global sensitivity of the estimator is calculated using these new bounds and we use this to get a private estimate. Because we are not clipping any points with high probability, we know that the form of our estimator is "empirical mean + noise", where the noise comes from a well-specified distribution (line 9). Thus, we can use this to produce a set which, with high probability, contains the true empirical mean (lines 9 and 10). This set becomes the bound on the mean at the next step and the process continues. We perform this process for both distributions induced by the BLB, yielding t mean estimates $\{\tilde{\theta}_m\}_{m\in[t]}$ and t covariance estimates $\{\tilde{\Sigma}_m\}_{m\in[t]}$.

We combine the t estimates of each parameter via a precision-weighting argument (lines 6 and 8) to get final private estimates of the mean and covariance of our distribution, ensuring in line 7 that the resulting covariance estimate is PSD. We then use these estimates, along with Assumption 3.0.2, to get valid confidence intervals (line 9).

Chapter 3

General Private Mean Estimation

Notice that lines 4 and 5 of Algorithm 1 rely on differentially private mean estimation, so we first need to build a framework for doing this. We imagine a d-dimensional distribution of interest with mean $\mu \in \mathbb{R}^d$, from which we have k realizations $\{\hat{\mu}_i\}_{i\in[k]}$. We call $\hat{\mu} = \frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i$.

For our mean estimation algorithm we need three assumptions, the first of which requires us to define a way of comparing how heavy-tailed two distributions are.

Definition 3.0.1. *Let $\mathcal{B}(\mu,\Sigma)$ and $\mathcal{C}(\mu,\Sigma)$ be families of distributions and B, C be random variables drawn from each such that $\mathbb{E}(B) = \mathbb{E}(C) = \mu$ and $Cov(B) = Cov(C) = \Sigma$. Let PSD_d be the set of all $d \times d$ PSD matrices. We say that $\mathcal{B}$ is* heavier-tailed *than $\mathcal{C}$ if $\forall\mu \in \mathbb{R}^d, \forall\Sigma \in PSD_d, \forall v \in \mathbb{R}^d$ such that $\|v\| = 1, \forall z > 0$:*

$$\mathbb{P}\left(v^T(B-\mu) \le z\right) \le \mathbb{P}\left(v^T(C-\mu) \le z\right).$$

Assumption 3.0.2. *The user provides a family of distributions $Q(\mu, \Sigma_{\tilde{\mu}})$ with heavier tails than the distribution of $\hat{\mu}$ as described in Definition 3.0.1. Additionally, each dimension of Q is a symmetric location-scale family with finite first and second moments.*

This is a generalization of the treatment in [6], which requires Q to be multivariate Gaussian.

Assumption 3.0.3. *The user of the algorithm provides $\tilde{\mu}_0 \in \mathbb{R}^d$ and $r_0 \in \mathbb{R}$ such that $\hat{\mu} \in B_2(\tilde{\mu}_0, r_0)$, where $B_2(\tilde{\mu}_0, r_0) \in \mathbb{R}^d$ is the ℓ_2 ball centered at $\tilde{\mu}_0$ with radius r_0.*

Assumption 3.0.4. *The user of the algorithm provides $\Sigma_\mu^U \in \mathbb{R}^{d\times d}$ such that $\hat{\Sigma}_\mu \preceq \Sigma_\mu^U$.*

These three assumptions provide the backbone of the iterative bound improvement in CoinPress. The algorithm tries to find the tightest possible bounds that, with high probability, do not clip any points. Assumption 3.0.3 ensures that the algorithm starts with sufficiently conservative bounds that the algorithm can tighten them and still contain $\hat{\mu}$. Assumptions 3.0.2 and 3.0.4 allow the algorithm to convert bounds on $\hat{\mu}$ to high-probability bounds on the $\hat{\mu}_i$.

We now present our mean estimation algorithm, which is a generalization of the CoinPress mean estimation algorithm [6] to distributions other than sub-Gaussians. For the remainder of the section, we assume that our three assumptions hold.

Algorithm 2 Modified CoinPress [6]

Input: $\hat{M} = (\hat{\mu}_1, \ldots, \hat{\mu}_k)$ from a distribution with mean μ and covariance Σ, $\tilde{\Sigma}$ such that $\hat{\Sigma} \preceq \tilde{\Sigma}$, $B_2(\tilde{\mu}_0, r_0)$ containing $\hat{\mu}$, family of distributions $Q_{\tilde{\mu}}(\cdot, \Sigma_{\hat{\mu}})$ with heavier tails than $\hat{\mu}$, $t \in \mathbb{N}^+$, $\rho^{\tilde{\mu}} > 0, \beta^{\tilde{\mu}} > 0$

Output: t estimates of μ that jointly respect $\rho^{\tilde{\mu}}$-zCDP

procedure MVMREC($\hat{\mu}_{1\ldots k}, \tilde{\mu}_0, r_0, Q, t, \rho^{\tilde{\mu}}, \beta^{\tilde{\mu}}$)

  $S = \tilde{\Sigma}^{1/2}$

  $\tilde{\mu}_0 = S^{-1}\tilde{\mu}_0$

  $r_0 = \max\left(diag\left(S^{-1}\right)\right) \cdot r_0$

  Define $\bar{M} \in \mathbb{R}^{k \times d}$ such that $\forall j \in [d], \forall m \in [k] : \bar{M}_{m,j} = \frac{1}{k}\sum_{m'=1}^{k} \hat{\mu}_{m',j}$. Note that each row $\bar{\mu}_{m,:}$ is equal to the d-dimensional empirical mean of $\hat{M}$

  $\hat{M} = \left(\hat{M} - \bar{M}\right) S^{-1}$

  for $m \in [t-1]$ **do**

    $(\tilde{\mu}_m, r_m, \sigma_m) = \text{MVM}(\hat{M}, \tilde{\mu}_{m-1}, r_{m-1}, Q_{\tilde{\mu}}(0, I_d), \frac{\rho^{\tilde{\mu}}}{2(t-1)}, \frac{\beta^{\tilde{\mu}}}{t})$ ▷ Algorithm 4

  $(\tilde{\mu}_t, r_t, \sigma_t) = \text{MVM}(\hat{M}, \tilde{\mu}_{t-1}, r_{t-1}, Q_{\tilde{\mu}}(0, I_d), \frac{\rho^{\tilde{\mu}}}{2}, \frac{\beta^{\tilde{\mu}}}{t})$

  $\forall m \in [t] : \tilde{\mu}_m = (S\tilde{\mu}_m) + \bar{\mu}_{1,:}$ ▷ convert mean estimates to proper scale

  $\forall m \in [t] : \vec{\sigma}_m^2 = \text{diag}\,(S\sigma_m)^2$ ▷ convert private noise variances to proper scale

  return $\{(\tilde{\mu}_m, \vec{\sigma}_m^2)\}_{m \in [t]}$

This algorithm respects zCDP and comes with a high-probability guarantee of unbiasedness. Proofs of Theorems 3.0.5 and 3.0.6 can be found in Appendices .2.2 and .2.3, respectively.

Theorem 3.0.5. *Algorithm 2 respects $\rho^{\tilde{\mu}}$-zCDP.*

Theorem 3.0.6. *Algorithm 2 produces t mean estimates $\{\tilde{\mu}_m\}_{m \in [t]}$ such that*

$$\mathbb{P}\left(\forall m \in [t] : \mathbb{E}\left(\tilde{\mu}_m\right) = \mu\right) \geq 1 - \beta^{\tilde{\mu}}.$$

In Section 4, we use this general mean estimation framework to estimate the means of both the $\{\hat{\theta}_i^{BLB}\}_{i \in [k]}$ and $\{\hat{\Sigma}_i^{BLB}\}_{i \in [k]}$.

Chapter 4

Parameter Estimation

Now that we have a way of privately estimating the means of the BLB-induced distributions, $\hat{\theta}^{BLB}$ and $\hat{\Sigma}^{BLB}$, we can move to lines 6, 8, and 9 of Algorithm 1, where we use the private estimates we get from CoinPress to produce our final parameter estimates and confidence intervals.

We start by noting a theorem that will be used throughout this section, whose proof can be found in Appendix .3.1. This is the multivariate version of the inverse-variance weighting argument commonly found in the meta-analysis literature, which gives the optimal (i.e. minimum (co)variance) way to combine a set of unbiased estimators.

Theorem 4.0.1. *For a parameter τ, say we are given a series of independent estimates $\{\hat{\tau}_m\}_{m\in[t]}$ such that $\mathbb{E}(\hat{\tau}_m) = \tau$ and $Cov(\hat{\tau}_m) = S_m$ for some positive definite (PD) S_m.*[1] *Then the minimum variance unbiased linear weighting of the $\{\hat{\tau}_m\}_{m\in[t]}$ is given by*

$$\hat{\tau} = \left(\Sigma_{m=1}^{t} S_m^{-1}\right)^{-1} \left(\Sigma_{m=1}^{t} S_m^{-1}\hat{\tau}_m\right),$$

which has $\mathbb{E}(\hat{\tau}) = \tau$ and $Cov(\hat{\tau}) = (\Sigma_{m=1}^{t} S_m^{-1})^{-1}$.

Specifically, by "minimum variance" we mean that any other unbiased linear weighting $\hat{\tau}'$ of the $\hat{\tau}_m$ will have $Cov(\hat{\tau}) \preceq Cov(\hat{\tau}')$.

4.0.1 Privately estimating $\hat{\Sigma}$

We now explore how to aggregate our $\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]}$ to get a differentially private estimate of $\hat{\Sigma}$. We consider two separate cases; one where we estimate only the diagonal of $\hat{\Sigma}^{BLB}$ and one where we estimate the full matrix.

Estimating diagonal of $\hat{\Sigma}$

We first consider the case where we care only about the diagonal of $\hat{\Sigma}^{BLB}$; that is, we don't care about the covariance structure of our parameters. Thus, without loss of generality,

[1] Note that this is slightly stronger than the PSD assumption we have been making.

we write $\hat{\Sigma}_i^{BLB} = \hat{V}_i I_d$ where $\hat{v}_{i,j}$ is the j^{th} element of $\hat{V}_i$. Though $\hat{V}$ is a random variable, we abuse notation in the typical way and also let $\hat{V} = \frac{1}{k}\sum_{i=1}^{k} \hat{V}_i$. Because we are assuming the off-diagonal elements of Σ are 0, estimating Σ reduces to estimating V.

We assume the user provides initial conditions such that Assumptions 3.0.2, 3.0.3, and 3.0.4, hold and appeal to Algorithm 2, substituting $\hat{V}_i$ for the general $\hat{\mu}_i$ from those statements. This yields the following result, which we prove in Appendix .3.2.

Theorem 4.0.2. *Applying Algorithm 2 to* $\hat{M} = (\hat{V}_1, \ldots, \hat{V}_k)$ *yields* $\{\tilde{V}_m, \vec{\sigma}_m^2\}_{m\in[t]}$.

We combine these into an estimator

$$\tilde{V} = \frac{\sum_{m=1}^{t} \tilde{V}_m/\vec{\sigma}_m^2}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2}. \tag{4.1}$$

Define Φ^{-1} *as the quantile function of the standard Gaussian and let*

$$\vec{\gamma} = \frac{1}{\sqrt{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2}} \Phi^{-1}\left(1 - \beta^{ub}/d\right) \tag{4.2}$$

for arbitrary $\beta^{ub} \in (0,1)$. *Then, for* $\tilde{\Sigma} = (\tilde{V} + \vec{\gamma})I_d$ *we have* $\mathbb{P}\left(\forall j \in [d] : \hat{\Sigma}_{j,j} \leq \tilde{\Sigma}_{j,j}\right) \geq 1 - \beta^{\tilde{V}} - \beta^{ub}$.

For the sake of generality, we will refer to $\beta^{\tilde{V}}$ as $\beta^{\tilde{\Sigma}}$ and refer to $\hat{\Sigma}$ as if it were diagonal such that if $\forall j \in [d] : \hat{\Sigma}_{j,j} \leq \tilde{\Sigma}_{j,j}$, then $\hat{\Sigma} \preceq \tilde{\Sigma}$.

Estimating full $\hat{\Sigma}$

In the case where we care about the covariance structure of our parameters, we instead estimate the entire $\hat{\Sigma}^{BLB}$ matrix. We start by creating a flattened upper triangular of each $\hat{\Sigma}_i^{BLB}$, which we call $\hat{S}_i^{\flat} \in \mathbb{R}^{\frac{d(d+1)}{2}}$. Our analysis then proceeds much like it did in Section 4.0.1. The primary difference is that we no longer treat upper bounding the diagonal of $\hat{\Sigma}^{BLB}$ as sufficient for valid confidence intervals. Instead, we need a full Löwner upper bound on $\hat{\Sigma}^{BLB}$. Again, we assume the user provides initial conditions such that Assumptions 3.0.2, 3.0.3, and 3.0.4 hold and appeal to Algorithm 2. We also use Algorithm 6 (HPUB) from Appendix .3.2, which produces a high-probability upper bound on a specified random variable.

Theorem 4.0.3. *Applying Algorithm 2 to* $\hat{M} = (\hat{S}_1^{\flat}, \ldots, \hat{S}_k^{\flat})$ *yields* $\{\tilde{S}_m^{\flat}, \vec{\sigma}_m^2\}_{m\in[t]}$.

We combine these into a single estimate

$$\tilde{S}^{\flat} = \frac{\sum_{m=1}^{t} \tilde{S}_m^{\flat}/\vec{\sigma}_m^2}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2}$$

and unflatten it back to a matrix $\tilde{S} \in \mathbb{R}^{d\times d}$. *Let*

$$\gamma = HPUB\left(\|\sigma_M^2\|_2, \beta^{ub}\right)$$

where $\sigma_M^2 \in \mathbb{R}^{d\times d}$ *is the unflattened version of* $\frac{1}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2}$.

Then, for $\tilde{\Sigma} = \tilde{S} + \gamma I_d$ *we have* $\mathbb{P}\left(\hat{\Sigma} \preceq \tilde{\Sigma}\right) \geq 1 - \beta^{\tilde{S}} - \beta^{ub}$.

See Appendix .3.2 for a proof. As before, for generality we refer to $\beta^{\tilde{S}}$ as $\beta^{\tilde{\Sigma}}$.

4.0.2 Privately estimating $\hat{\theta}$

To privately estimate $\hat{\theta}$, we again use Algorithm 2, substituting $\hat{\theta}_i$ for $\hat{\mu}_i$ in the general statement. We again assume the user provides initial conditions such that Assumptions 3.0.2 and 3.0.3 hold.

We could assume that the user provides bounds that respect Assumption 3.0.4, but we have another option available to us which will often give much tighter estimates; relating the private estimate $\tilde{\Sigma}$ to the covariance of the $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$.

Although we are scaling up our subsets to the original data size within the BLB to get correct overall covariance estimates, this does not imply that the covariance of our $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$ is appropriately scaled. In fact, this covariance will often be roughly the same as if $\hat{\theta}$ were simply run on subsets of size $\frac{n}{k}$. So, the covariance of the $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$ should be roughly $\frac{r(n/k)}{r(n)}\hat{\Sigma}$, where r is the convergence rate of the estimator in question. For example, the covariance of OLS coefficients decays with $\frac{1}{n}$, so if $\hat{\theta}$ represents OLS estimation we would say the covariance is $\frac{1/(n/k)}{1/n}\Sigma = k\hat{\Sigma}$. We upper bound this with $k\tilde{\Sigma}$. Under Assumption 2.0.1, $k\tilde{\Sigma}$ will be a Löwner upper bound on $k\hat{\Sigma}$ with probability $1-\beta^{\tilde{\Sigma}}$ provided we privately estimate the entire covariance matrix as in Section 4.0.1. If we estimate only the diagonal as in Section 4.0.1, we instead have a $1-\beta^{\tilde{\Sigma}}$ probability guarantee that $k\tilde{\Sigma}$ will upper bound the empirical variance in each dimension.

Theorem 4.0.4. *Applying Algorithm 2 to $\hat{M} = (\hat{\theta}_1, \ldots, \hat{\theta}_k)$ yields $\{\tilde{H}_m, \vec{\sigma}_m^2\}_{m\in[t]}$.*

We then construct a combined estimator $\tilde{\theta}$ where

$$\tilde{\theta} = \frac{\sum_{m=1}^{t} \tilde{H}_m/\vec{\sigma}_m^2}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2}. \tag{4.3}$$

This new estimator has covariance $\Sigma_{\tilde{\theta}} = \frac{1}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2} I_d$.

In particular, we say that $\mathbb{P}\left(\tilde{\theta} \sim N\left(\theta, \Sigma_{\tilde{\theta}}\right)\right) \geq 1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$.

See Appendix .3.3 for a proof.

4.0.3 Confidence Intervals

We now have estimates of the mean and covariance of $\hat{\theta}$; $\tilde{\theta}$ and $\tilde{\Sigma}$ respectively, such that $\tilde{\theta} \sim N\left(\hat{\theta}, \Sigma_{\tilde{\theta}}\right)$ and $\hat{\Sigma} \preceq \tilde{\Sigma}$ with probability $1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$, as well as a distribution $Q_{\tilde{\theta}}$ we assume to be heavier tailed than that of $\hat{\theta}$. We can represent our approximation of the sampling distribution as the compound distribution

$$Q_{\tilde{\theta}}\left(\hat{\theta} + N\left(0, \Sigma_{\tilde{\theta}}\right), \tilde{\Sigma}\right),$$

where we interpret $\hat{\theta}$ and $\tilde{\Sigma}$ as realizations of their random variables.

We appeal to Algorithm 8 in Appendix .6, which in turn uses Algorithm 6 to get an upper/lower bounds on quantiles of a random variable. This yields two different notions of confidence intervals; one for which the confidence interval is valid with high probability, and one which is valid with probability 1. See Appendices .5.1 and .5.2 for proofs.

Theorem 4.0.5 (General confidence intervals (valid with high probability))**.** *We apply Algorithm 8 to each dimension of $Q_{\tilde{\theta}}$ at levels $\{\alpha_j\}_{j\in[d]}$ and get $\{(ci_j^l, ci_j^u)\}_{j\in[d]}$. Then, with probability $1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$,*

$$\forall j \in [d] : \mathbb{P}\left(\hat{\theta}_j \in \left(ci_j^l, ci_j^u\right)\right) \geq 1 - \alpha_j.$$

Corollary 4.0.6 (General confidence intervals (valid with probability 1))**.** *For all $j \in [d]$, let $\alpha'_j = \alpha_j - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$ for some $\{\alpha_j\}_{j\in[d]}$. Then, provided that $\forall j \in [d] : \alpha'_j > 0$, we apply Algorithm 8 to each dimension of $Q_{\tilde{\theta}}$ at levels $\{\alpha'_j\}_{j\in[d]}$ and get $\{(ci_j^l, ci_j^u)\}_{j\in[d]}$. Then,*

$$\forall j \in [d] : \mathbb{P}\left(\hat{\theta}_j \in \left(ci_j^l, ci_j^u\right)\right) \geq 1 - \alpha_j.$$

Note that although we give these results based on a general $Q_{\tilde{\theta}}$, the compound distribution behaves much more nicely for many distributions. Most notable is that if $Q_{\tilde{\theta}}$ is multivariate Gaussian, the resulting compound distribution is $N\left(\tilde{\theta}, \tilde{\Sigma} + \Sigma_{\tilde{\theta}}\right)$. In this case, we note that $\forall j \in [d] : \tilde{\theta}_j \sim N\left(\hat{\theta}_j, \tilde{\Sigma}_{j,j} + \Sigma_{\tilde{\theta}_{j,j}}\right)$, and so we can calculate confidence intervals directly from the CDF of the univariate Gaussian.

Chapter 5

Empirical Evaluation

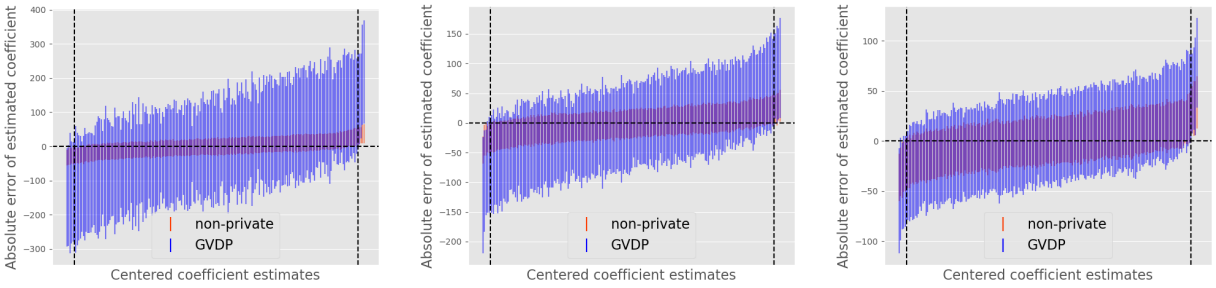
5.0.1 Demonstration of unbiasedness and valid confidence intervals

We start with empirical demonstrations of our core result, showing that we can produce unbiased parameter estimates and valid confidence intervals when the requisite assumptions hold. For every evaluation, we aim to get valid dimension-wise confidence intervals rather than a single valid confidence set. Additionally, inside of the GVDP algorithm, we always run CoinPress for $t = 5$ iterations.

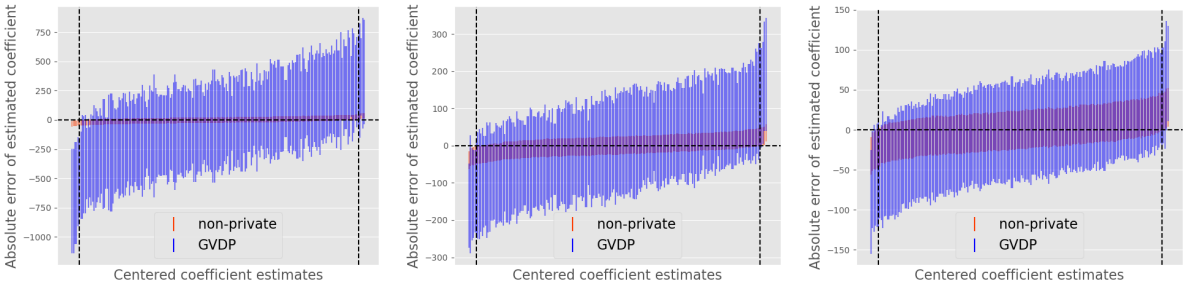
Demonstration: OLS regression

We begin by testing on a straightforward task, parameter estimation for an OLS model. We ran experiments over sample sizes $n = \{100000, 200000, 500000, 1000000\}$, numbers of partition subsets $k = \{500, 1000, 2500, 5000\}$, and number of explanatory variables $d = \{10, 50\}$. Each set of experiments was run for 20 iterations. In a single iteration, we generate data from a linear model $y = X\beta + \epsilon$ with Gaussian covariates, Gaussian error, and correlation structure such that the effective rank of the resulting data is $\approx d - 1$. Note that because of how we are generating the data, the non-private confidence intervals are basically constant across values of n. We then privately estimate the values of the d coefficients and their associated standard errors. We assume the sampling distribution of the coefficients is multivariate Gaussian and imagine that the user sets all upper bounds ≈ 100 times larger than the tightest possible upper bounds. The width of the confidence intervals produced by GVDP scale with this bound, so tighter (looser) upper bounds will lead to tighter (looser) confidence intervals. We present these results in Figure 5.1.

Each plot consists of coefficient estimates centered around their true values and presented in increasing order, with vertical bars representing the 95% confidence interval for that estimate. We expect properly calibrated confidence intervals to cross the x-axis at the vertical dotted black lines, placed at the 2.5^{th} and 97.5^{th} quantiles, which is the behavior we observe in each plot. Note that the scale of the y-axis changes for each plot, as our primary goal is to clearly show our algorithm's performance relative to the theoretical guarantees and non-private results, rather than how it changes over the different parameters.

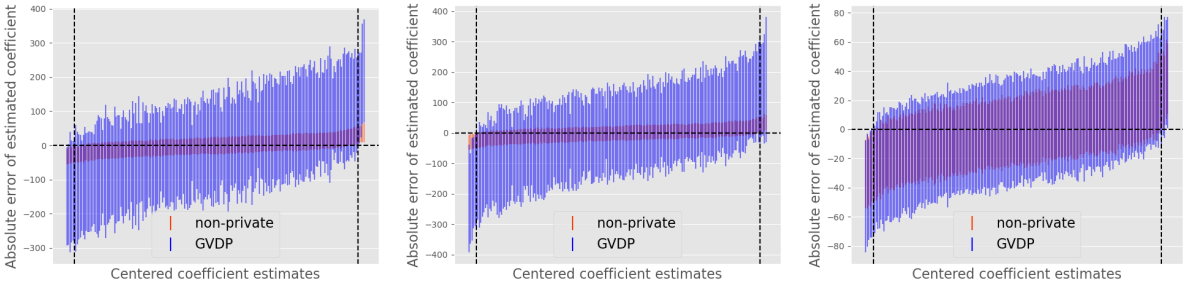


(a) $n = 100{,}000, k = 500, d = 10, \rho = .1$ (b) $n = 200{,}000, k = 1{,}000, d = 10, \rho = .1$ (c) $n = 500{,}000, k = 2{,}500, d = 10, \rho = .1$



(d) $n = 100{,}000, k = 500, d = 50, \rho = .1$ (e) $n = 200{,}000, k = 1{,}000, d = 50, \rho = .1$ (f) $n = 500{,}000, k = 2{,}500, d = 50, \rho = .1$

Figure 5.1: OLS: Distribution of coefficient estimates and 95% confidence intervals



(a) $n = 100{,}000, k = 500, d = 10, \rho = .1$ (b) $n = 1{,}000{,}000, k = 500, d = 10, \rho = .1$ (c) $n = 1{,}000{,}000, k = 5{,}000, d = 10, \rho = .1$

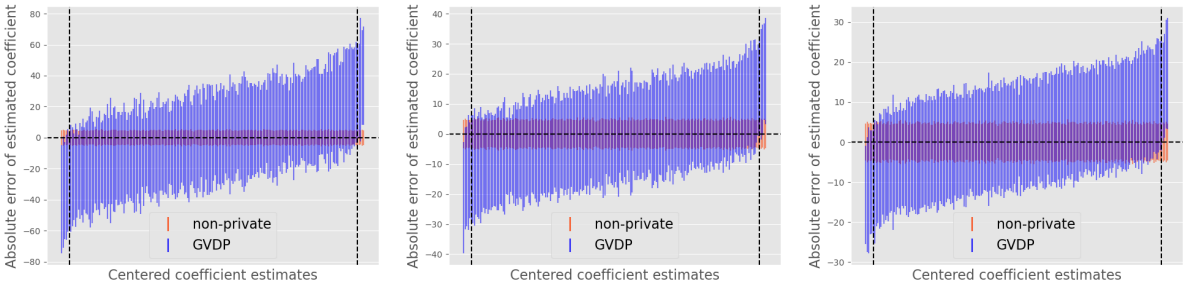
Figure 5.2: OLS: Distribution of coefficient estimates and 95% confidence intervals. Showing the effects of varying n vs. k

Figure 5.2 demonstrates the principle that the noise due to privacy in our algorithm scales with k rather than n. Note that plots (a) and (b) are essentially identical; the non-private confidence intervals are identical (even though n increased) as an artefact of our data-generating process, and the private confidence intervals are identical because, additionally, we didn't increase k. In general, the analyst should choose k to be as large as possible, subject to the constraint that the BLB approximates the sampling distribution well. We discuss this further in Section 6.0.1.

Demonstration: Logistic regression with imbalanced class output

In Figure 5.3 we show qualitatively similar results across a smaller set of experiments for a more challenging setting; logistic regression with imbalanced classes. We generate data

as before but run the outcome variable y through a scaled logistic function to get a new outcome variable $y' \in \{0,1\}^n$. Specifically, for $p_i = \frac{1}{1+\exp(-X_i\beta)}$ and $\bar{p} = \frac{1}{n}\sum_{i=1}^n p_i$, we have $\mathbb{P}(y_i = 1) = \frac{p_i}{\bar{p}} \cdot 0.05$. This induces a minority class that occurs with probability ≈ 0.05.



(a) $n = 100{,}000, k = 500, d = 10, \rho = .1$ (b) $n = 100{,}000, k = 1{,}000, d = 10, \rho = .1$ (c) $n = 100{,}000, k = 2{,}000, d = 10, \rho = .1$

Figure 5.3: Logistic Regression: Distribution of coefficient estimates and 95% confidence intervals

Demonstration: Logistic regression with fully sparse data

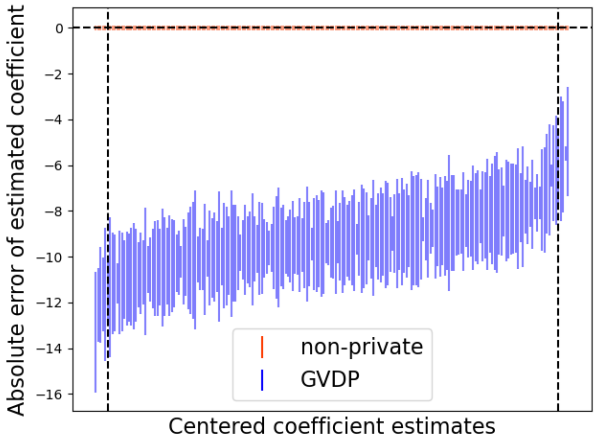
The requisite bootstrap assumptions obviously do not hold for all estimators and data distributions. We make our setting more difficult again for Figure 5.4, by generating a new set of covariates X' such that $\forall j \in [d] : X'_{i,j} = \mathbb{1}(X_{i,j} \geq z_j)$ where $z_j = \min_{r \in \mathbb{R}} \frac{1}{n}\sum_{i=1}^n \mathbb{1}(X_{i,j} > r) \leq 0.05$. That is, X' is itself now a binary matrix with highly imbalanced classes. The rightmost plot shows distributions of the d coefficient estimates induced by BLB, with a black dotted line at the value of the non-private coefficient. Note that at $n = 100{,}000$, the BLB distributions are essentially point masses at two extreme points, whereas at $n = 1{,}000{,}000$, they are much closer to being a symmetric distribution about the true coefficient value.

Generalizing CoinPress beyond multivariate sub-Gaussians

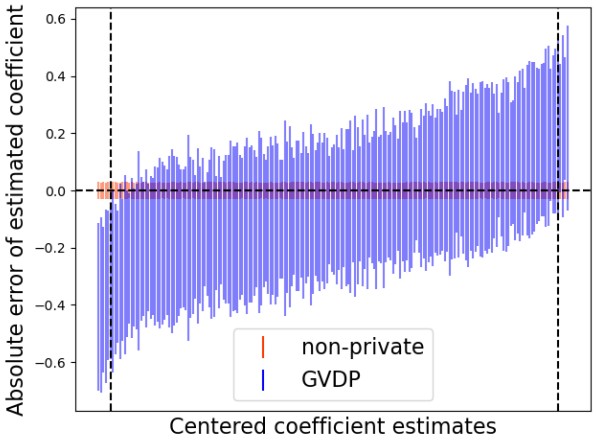
In Figure 5.5 we provide evidence that our generalization of CoinPress beyond sub-Gaussian distributions delivers on its promises. We pretend as if the estimator and data were such that the distributions induced by the BLB were are dominated by the multivariate Laplace (i.e. they are sub-Exponential), and the analyst overestimated the relevant parameters by a factor of 100. We show results corresponding to two different methods for calculating the clipping parameters at each step of CoinPress. The *analytic* solution calculates the bound using a theoretical bound given in Theorem .2.4, while the *approximate* solution calculates an approximate upper bound using Algorithm 7.

5.0.2 Comparison with AdaSSP [43]

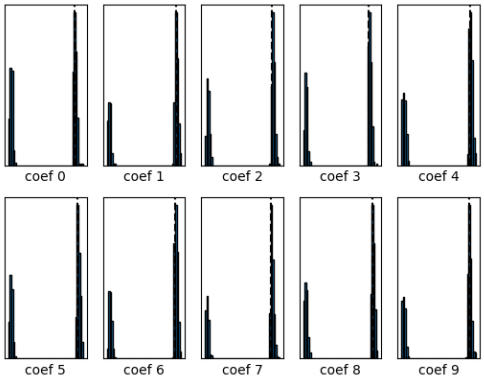
Although our method generalizes to a variety of models, we imagine it will often be used for OLS regression. So, we compare GVDP's performance to that of the Adaptive Sufficient Statistic Perturbation (AdaSSP) algorithm from [43], one of the best-performing algorithms for DP OLS.



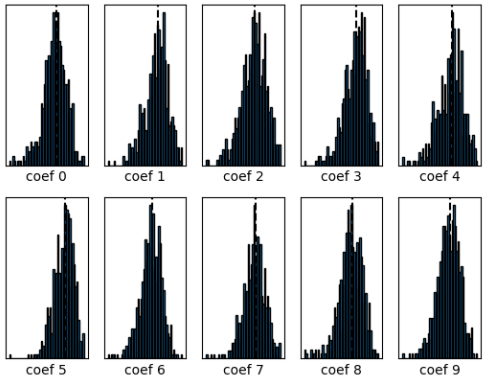
(a) $n = 100{,}000, k = 500, d = 10, \rho = .1$



(b) $n = 1{,}000{,}000, k = 500, d = 10, \rho = .1$



(c) BLB coefficient distributions for $n = 100{,}000, k = 500$



(d) BLB coefficient distributions for $n = 1{,}000{,}000, k = 500$

Figure 5.4: Logistic Regression with unbalanced binary fatures: Distribution of coefficient estimates and 95% confidence intervals

The two algorithms differ in a few key ways. First, AdaSSP does not attempt to do unbiased parameter estimation or give valid confidence intervals; instead, it is trying to estimate OLS coefficients with minimal ℓ_2 error. For purposes of comparison, we will ignore confidence intervals altogether and focus only on the parameter estimates. Second, AdaSSP assumes only bounds on the data, assuming that we can specify data domains $\mathcal{X}, \mathcal{Y}$ for our covariates and outcome, respectively, such that $\|\mathcal{X}\| = \sup_{x \in \mathcal{X}} \|x\|_2$ where $x \in \mathbb{R}^m$ and $\|\mathcal{Y}\| = \sup_{y \in \mathcal{Y}} |y|$. Ignoring the assumptions needed for confidence intervals for now, GVDP requires Assumptions 3.0.3, and 3.0.4 on the distribution of covariances induced by the bag of little bootstraps, as well as Assumption 3.0.3 on the distribution of means. It's worth noting the qualitative difference between these methods; AdaSSP requires the user to bound the data, GVDP requires the user to bound moments of the parameter distribution. For most analyses, we expect the AdaSSP bounds to be easier to specify tightly than those of GVDP. However, GVDP is designed to scale more gracefully under overly conservative bounds.

We generate data just as we did in Section 5.0.1 and compare AdaSSP and GVDP across a number of privacy budgets and what we call "overestimation factors". Say we

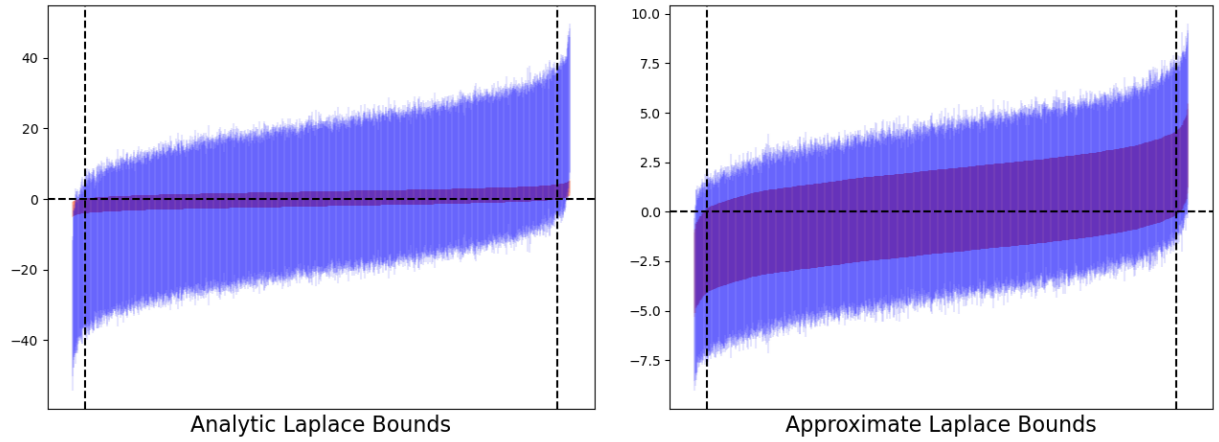


Figure 5.5: Distribution of coefficient estimates and 95% confidence intervals for $k = 5{,}000, d = 10, \rho = 0.1$ for multivariate Laplace distribution

OF	1	1.5	2	3	4	5	10	1000	10000
non-private	10.06	10.06	10.06	10.06	10.06	10.06	10.06	10.06	10.06
AdaSSP	10.34	12.96	28.65	139.51	463.21	6223.42	18689.97	20521.21	20064.88
GVDP	14.84	15.18	14.72	14.41	14.97	14.91	15.07	18.08	26.48

Table 5.1: Average absolute estimation error for each algorithm by overestimation factor (OF)

have realized data $X \in \mathbb{R}^{n \times d}, y \in \mathbb{R}^d$. For an overestimation factor of c, we set the bounds for AdaSSP to $c\left(\sup_{x \in X} \|x\|_2\right)$ and $c\left(\sup_{y \in Y} \|y\|\right)$. For GVDP, we perform the BLB step to get our $\{\hat{\theta}_i^{BLB}\}_{i \in [k]}$, which we'll say has empirical mean $\hat{\mu} \in \mathbb{R}^d$ and empirical covariance $\hat{\Sigma} \in \mathbb{R}^{d \times d}$. We set our ℓ_2 bounding ball for the mean of the distribution as $B_2\left(\hat{\mu}, c\left(\max_{j \in [d]} \hat{\mu}_j\right)\right)$ and our Löwner upper bound on the covariance as $c\left(diag(\hat{\Sigma}) I_d\right)$. Runs of non-private OLS are included for comparison, but the overestimation factor does not affect them.

All experiments were run with $n = 500{,}000, k = 2{,}500, d = 10$, and $\rho = 0.1$. The general trends were similar across other parameter combinations. We run each method over 100 simulations, estimating d coefficients at each iteration, so each method produces 1,000 coefficient estimates overall.

We see that AdaSSP performs well with slightly overestimated bounds but scales poorly with overly conservative bounds. GVDP performs a bit less well at low overestimation factors but scales much nicer.

5.0.3 Comparison with UnbiasedPrivacy [21]

In Figure 5.6, we compare our algorithm to the UnbiasedPrivacy (UP) algorithm from [21]. As stated in the intro, our framework (GVDP) and UP are similar in that both use the BLB to approximate the sampling distribution of the estimator, and then a private aggregation step to estimate its parameters. UP is designed for estimators with a univariate Gaussian sampling distribution, so we focus on that setting here.

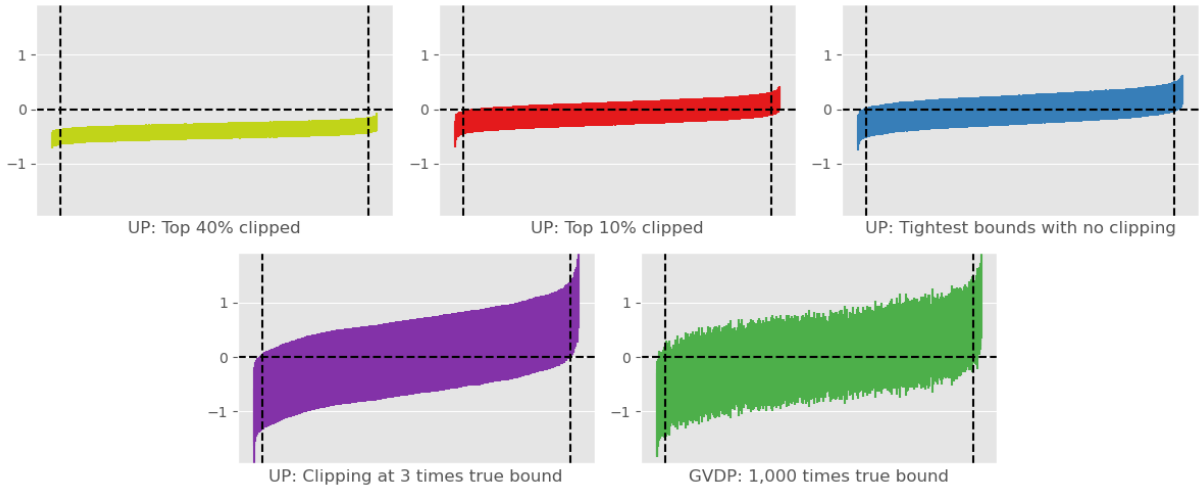


Figure 5.6: Comparison of UP [21] and GVDP: distribution of coefficient estimates and 95% confidence intervals for $n = 50{,}000, k = 500, d = 1, \rho = 0.1$

While the goal of GVDP is to let users set very conservative bounds (which the algorithm improves upon) and not clip any points in the aggregation step, UP requires that users set reasonably good bounds upfront. UP never tightens bounds that are too loose, but it will attempt to bias-correct the results if you analyst's bounds clip bounds (which our algorithm does not).

We test UP across four scenarios using the implementation provided by the authors.[1] As suggested in [21], we split the privacy budget evently between the mean estimation task and the estimation of the proportion of points that are clipped. We consider two cases in which we expect UP to perform bias correction, clipping the top 40% (yellow) and 10% (red) of the BLB estimates, and two in which we don't ,clipping bounds set as tightly as possible (blue) with no clipping and set three times larger than the true values of the parameters (purple). As a point of comparison, we show results from GVDP at bounds set 1,000 times larger than the relevant parameters (green).

Note that, in the 10% clipping case, UP essentially delivers as advertised; it gives (approximately) valid confidence intervals and does so at a lower error than UP does under no clipping. However, it is unable to achieve this when the top 40% of BLB estimates are clipped. Under the parameters we tested, GVDP at an overestimation factor of 1,000 performs more-or-less identically to UP at an overestimation factor of 3.

5.0.4 Replication of [10]

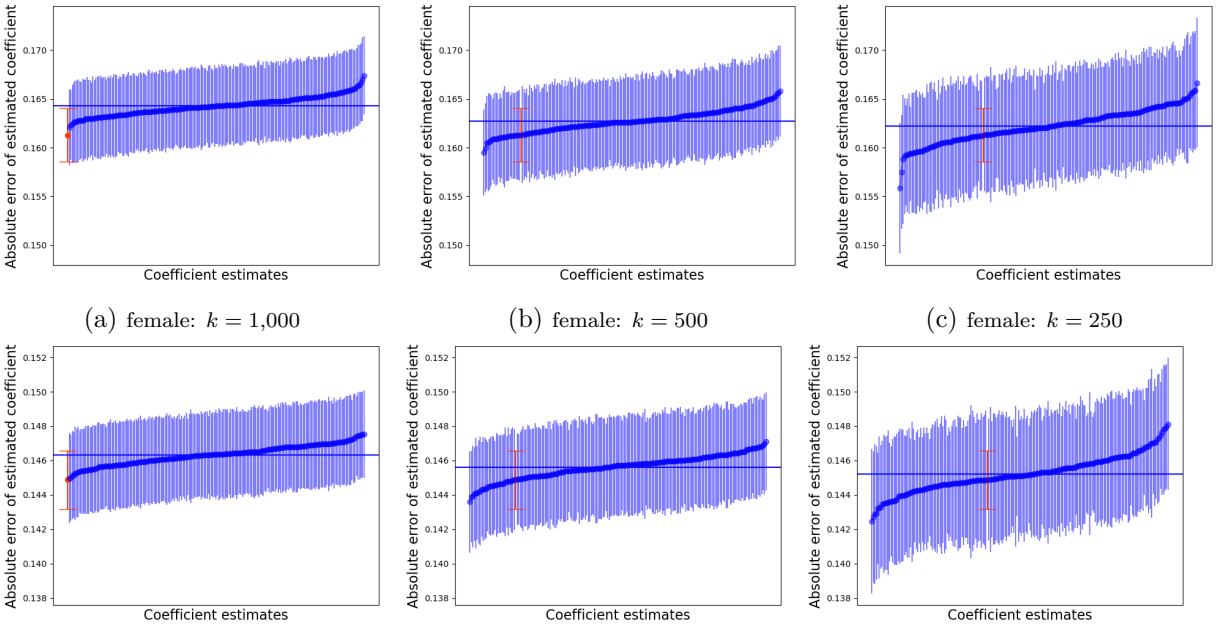
Inspired by the tests of [5], we attempt to replicate the core analysis of [10] under the constraints of DP. We use CPS ASEC data from 1994 to 1996 [36] and run OLS to estimate the following model:

$$\log(\text{inc\_wage}) = \beta_0 + \beta_1 \text{educ} + \beta_2 \text{PE} + \beta_3 \text{PE}^2 + \beta_4 \text{PE}^3 + \beta_5 \text{white} + \epsilon,$$

[1] Implementation can be found at https://github.com/schwenzfeier/udp.

where *inc_wage* is an individual's total pre-tax wage and salary income, *educ* is years of education, *PE* is potential years of work experience, and *white* indicated whether or not the individual identifies as white. The effect of education on income is our question of interest, with the other variables serving as controls. This allows us to use the full OLS model within the bootstrap, but release and privately estimate only the estimated mean/variance of β_1.

[10] runs this model separately for males and females; in Figure 5.7 we report the female results ($n = 95{,}177$) as well as for males and females combined ($n = 197{,}756$). To be consistent with [5], we report results in *approximate DP* with $(\epsilon, \delta) = (5, 1/k)$, which translates to $\rho \approx \{0.879, 1.06, 1.23\}$ for $k = \{1000, 500, 250\}$. All results are run with $t = 5$ CoinPress iterations and an overestimation factor of 100, and we run the GVDP estimation algorithm 200 times.



(a) female: $k = 1{,}000$ (b) female: $k = 500$ (c) female: $k = 250$

(d) combined: $k = 1{,}000$ (e) combined: $k = 500$ (f) combined: $k = 250$

Figure 5.7: Distribution of coefficient estimates and 95% confidence intervals for females only and both males and females (combined) with $\epsilon = 5$. The dot with a capped error bar represents the non-private estimate and confidence interval. The wider bars are the upper/lower bounds on the confidence intervals for the runs of GVDP. The horizontal line is the empirical mean of the GVDP estimates.

We note that our bootstrapped means do not always equal to the non-private mean in expectation, so although our algorithm's guarantees with respect to the bootstrapped distribution are met, they do not imply guarantees relative to the non-private answer as we hope they would. We see in these plots a clear trade-off. At $k = 1{,}000$, our confidence intervals are fairly tight, but are essentially centered around the upper end of the non-private confidence interval rather than the the true coefficient estimate. At $k = 250$ we minimize bias by generating more a representative bootstrap distribution, but do so at the cost of wider confidence intervals.

Chapter 6

Discussion

6.0.1 Choosing k

Recall from our explanation of Algorithm 1 that k is the number of subsets into which we partition our original data, which in turn becomes the number of elements fed into our private mean estimation algorithm, Algorithm 2. This presents a trade-off for the user; when k is large, the sensitivity of our aggregator decreases (Line 7 of Algorithm 4) and thus so does the variance of the noise we need to add for privacy. On the other hand, we assume that the mean and covariance estimates we get from the BLB reasonably approximate the mean and covariance of the true sampling distribution of the parameters, which is provably true only as $n \to \infty$ and $\frac{n}{k} \to \infty$ (and our estimator is Hadamard differentiable) [26].

Consider that once $\frac{n}{k}$ is large enough that we are in the asymptotics, there is no use in further increasing the ratio of n to k; we are better served by increasing k and reducing the noise needed for privacy. So, the best possible case for an analyst is that they choose the largest k such that the BLB estimates, operating over subsets of size $\frac{n}{k}$, approximates the true parameters of the sampling distribution. In cases where this is not possible, the guarantees of unbiasedness estimates and valid confidence intervals hold given that the expectation of the mean estimates we get from the BLB are equal to the expectation of the sampling distribution, and the mean of the covariance estimates from BLB is a Löwner upper bound on the covariance of the sampling distribution. If the analyst wants dimension-wise confidence intervals rather than a joint confidence set, the requirement of a Löwner upper bound can be relaxed such that each element of the diagonal of the mean BLB covariance estimate is larger than the corresponding diagonal element of the sampling distribution's covariance.

6.0.2 Future work

We believe our approach can be naturally split into three distinct sections, each of which could potentially be improved on in some way.

First, the algorithm runs the BLB algorithm over a partition of the data, comprising k disjoint subsets. As we stated in Section 6.0.1, we have a general description of what we want from our k; however, more work could be done on how to actually effectively choose

k for different models. Moreover, other versions of the bootstrap could be implemented with downstream changes to the aggregation and privacy accounting.

Second, the analyst must set bounds for the induced BLB distributions. Though our method scales reasonably well with overly conservative bounds, it's still worth trying to find tight bounds if possible. Our algorithm would benefit from methods that can privately find moderately conservative bounds on the empirical mean and covariance of collection of observations.

Finally, we have the private aggregation step. We use the generalized version of CoinPress, which has theoretical bounds for the multivariate sub-Gaussian and sub-Exponential settings, but currently relies on Monte Carlo sampling for other distributions. Analytical bounds, or more efficient means of calculating quantiles from known distributions, would help this become more practical computationally. Additionally, there is nothing in this work that necessitates the use of CoinPress; the framework requires only some DP aggregator. If other aggregation algorithms were developed that, for example, had fewer assumptions than CoinPress, they could likely be substituted without necessitating too many other changes to the framework.

Finally, we hope that this work will spur more interest in the assumptions the community makes of analysts and how, when not met, they can affect algorithms' guarantees. In this vein, we also hope to see more work developing algorithms that are robust to a lack of a priori knowledge about the data.

References

[1] Daniel Alabi, Audra McMillan, Jayshree Sarathy, Adam Smith, and Salil Vadhan. Differentially private simple linear regression. *arXiv preprint arXiv:2007.05157*, 2020.

[2] Marco Avella-Medina, Casey Bradshaw, and Po-Ling Loh. Differentially private inference via noisy optimization, 2021.

[3] Jordan Awan and Aleksandra Slavkovic. Differentially private inference for binomial data, 2019.

[4] Andrés F Barrientos, Jerome P Reiter, Ashwin Machanavajjhala, and Yan Chen. Differentially private significance tests for regression coefficients. *Journal of Computational and Graphical Statistics*, 28(2):440–453, 2019.

[5] Andrés F. Barrientos, Aaron R. Williams, Joshua Snoke, and Claire McKay Bowen. A feasibility study of differentially private summary statistics and regression analyses for administrative tax data, 2021.

[6] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. *Advances in Neural Information Processing Systems*, 33, 2020.

[7] Thomas Brawner and James Honaker. Bootstrap inference and differential privacy: Standard errors for free. 2018.

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

[9] Clément L Canonne, Gautam Kamath, Audra McMillan, Adam Smith, and Jonathan Ullman. The structure of optimal private tests for simple hypotheses. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 310–321, 2019.

[10] David Card. The causal effect of education on earnings. *Handbook of labor economics*, 3:1801–1863, 1999.

[11] C.J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 12 1934.

[12] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, page 202–210, New York, NY, USA, 2003. Association for Computing Machinery.

[13] Vito D'Orazio, James Honaker, and Gary King. Differential privacy for social science inference. *Sloan Foundation Economics Research Paper*, (2676160), 2015.

[14] Jörg Drechsler, Ira Globus-Harris, Audra McMillan, Jayshree Sarathy, and Adam D. Smith. Non-parametric differentially private confidence intervals for the median. *CoRR*, abs/2106.10333, 2021.

[15] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. Differentially private confidence intervals, 2020.

[16] George Duncan and Diane Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Association*, 81(393):10–18, 1986.

[17] George Duncan and Diane Lambert. The risk of disclosure for microdata. *Journal of Business & Economic Statistics*, 7(2):207–217, 1989.

[18] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

[19] 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.

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

[21] Georgina Evans, Gary King, Margaret Schwenzfeier, and Abhradeep Thakurta. Statistically valid inferences from privacy protected data, 2019.

[22] Cecilia Ferrando, Shufan Wang, and Daniel Sheldon. Parametric bootstrap for differentially private confidence intervals, 2021.

[23] Stephen E Fienberg and Julie McIntyre. Data swapping: Variations on a theme by dalenius and reiss. In *International Workshop on Privacy in Statistical Databases*, pages 14–29. Springer, 2004.

[24] Marco Gaboardi, Hyun Lim, Ryan Rogers, and Salil Vadhan. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *International conference on machine learning*, pages 2111–2120. PMLR, 2016.

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

[26] Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 795–816, 2014.

[27] Diane Lambert. Measures of disclosure risk and harm. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 9:313–313, 1993.

[28] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[29] Jing Lei, Anne-Sophie Charest, Aleksandra Slavkovic, Adam Smith, and Stephen Fienberg. Differentially private model selection with penalized and constrained likelihood. *arXiv preprint arXiv:1607.04204*, 2016.

[30] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[31] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[32] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[33] Víctor Peña and Andrés F Barrientos. Differentially private methods for managing model uncertainty in linear regression models. *arXiv preprint arXiv:2109.03949*, 2021.

[34] Jerome P Reiter. Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association*, 100(472):1103–1112, 2005.

[35] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[36] Steven Ruggles, Sarah Flood, Sophie Foster, Ronald Goeken, Jose Pacas, Megan Schouweiler, and Matthew Sobek. Ipums usa: Version 11.0 [dataset], 2021.

[37] Or Sheffet. Differentially private ordinary least squares. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3105–3114. PMLR, 06–11 Aug 2017.

[38] Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Proc. 3rd Berkeley Sympos. Math. Statist. Probability 1, 197-206 (1956)., 1956.

[39] Charles Stein and Willard James. Estimation with quadratic loss. In *Proc. 4th Berkeley Symp. Mathematical Statistics Probability*, volume 1, pages 361–379, 1961.

[40] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[41] Mariia Vladimirova, Stéphane Girard, Hien Nguyen, and Julyan Arbel. Sub-weibull distributions: Generalizing sub-gaussian and sub-exponential properties to heavier tailed distributions. *Stat*, 9(1), Jan 2020.

[42] Duy Vu and Aleksandra Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pages 138–143. IEEE, 2009.

[43] Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain, 2018.

[44] Yue Wang, Daniel Kifer, and Jaewoo Lee. Differentially private confidence intervals for empirical risk minimization. *Journal of Privacy and Confidentiality*, 9(1), Mar. 2019.

[45] Yue Wang, Jaewoo Lee, and Daniel Kifer. Revisiting differentially private hypothesis tests for categorical data. *arXiv preprint arXiv:1511.03376*, 2015.

[46] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

APPENDICES

.1 Bag of Little Bootstraps

This algorithm statement is adapted and simplified for our purposes; readers interested in the original version should consult [26]. We say that $\mathcal{X}$ is our data universe, $\mathcal{D}$ is a distribution over $\mathcal{X}$, and our realized data $X \in \mathbb{R}^{n \times m}$ are drawn from $\mathcal{D}^n$. For an arbitrary estimator $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$, we define $\hat{\theta}(\mathcal{D}) = \mathbb{E}_{X \sim \mathcal{D}^n}\left(\hat{\theta}(X)\right)$.

Algorithm 3 Bag of little bootstraps (BLB)

Input: data set $X \in \mathbb{R}^{n \times m}$, estimator $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$, estimator quality assessment ξ, k number of subsets of partition, r number of bootstrap simulations

Output: k estimates of $\hat{\theta}(\mathcal{D})$

procedure BLB($X, \hat{\theta}, k, r$)
  Randomly partition X into k subsets $\{X_i\}_{i \in [k]}$
  for $i \in [k]$ **do**
    $b = |X_i|$
    $\{\hat{\theta}_{i,c}\}_{c \in [r]} = \varnothing$
    for $c \in [r]$ **do**
      sample $(n_1, \ldots, n_b) \sim \text{Multinomial}(n, \mathbf{1}_b/b)$
      create $X_i^U \in \mathbb{R}^{n \times m}$ by including the j^{th} element of X_i n_j times
      $\hat{\theta}_{i,c} = \hat{\theta}(X_i^U)$
    $\hat{\theta}_i = \xi\left(\{\hat{\theta}_{i,c}\}_{c \in [r]}\right)$
  return $\{\hat{\theta}_i\}_{i \in [k]}$

.2 General mean estimation

.2.1 Modified CoinPress Algorithm - One Step Improvement

Algorithm 4 One Step Private Improvement of Mean Ball

Input: $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_k)$ from a distribution with mean 0 and covariance with smaller Löwner order than I_d, $B_2(\tilde{\mu}, r)$ containing $\hat{\mu}$, family of distributions $Q_{\tilde{\mu}}(\cdot, I_d)$, $\rho_m > 0, \beta_m > 0$

Output: A ρ_s-zCDP ball $B_2(\tilde{\mu}', r')$ and scale of the privatizing noise σ

1: **procedure** MVM($\hat{M}, \tilde{\mu}, r, Q, \tilde{\Sigma}, \rho_m, \beta_m$)
2: $\beta_s = \beta_m/2$
3: Let $R \sim Q_{\tilde{\mu}}(0, I_d)$
4: Set γ_1 such that $\mathbb{P}(\|R\|_2 > \gamma_1) \leq \frac{\beta_s}{k}$
5: Set γ_2 such that $\mathbb{P}(\|R\|_2 > \gamma_2) \leq \beta_s$
6: Project each $\hat{\mu}_i$ into $B_2(\tilde{\mu}, r + \gamma_1)$.
7: $\Delta = 2(r + \gamma_1)/k$.
8: $\sigma = \frac{\Delta}{\sqrt{2\rho_s}}$
9: Compute $\tilde{\mu}' = \frac{1}{k}\sum_i \hat{\mu}_i + Y$, where $Y \sim N\left(0, \sigma^2 I_d\right)$.
10: $r' = \gamma_2 \sqrt{\frac{1}{k} + \frac{2(r+\gamma_1)^2}{k^2 \rho_s}}$
11: **return** $(\tilde{\mu}', r', \sigma)$.

.2.2 Proof of Theorem 3.0.5

Proof. Algorithm 2 begins and ends by scaling the data to have empirical mean 0 and covariance which is Löwner upper bounded by I_d. The covariance scaling parameter is chosen independently of the data and the rest of the steps in the algorithm are invariant under location shift. So, our privacy analysis rests on the application of Algorithm 4 in lines 8 and 9 of Algorithm 2.

Algorithm 4 interacts with the raw data only in line 9, so privacy reduces to correct specification of Δ (the ℓ_2 sensitivity of the mean) and application of the Gaussian mechanism. The data are projected into $B_2(\tilde{\theta}, r + \gamma_1)$, and so the most a single data point can be changed in ℓ_2 norm is $2(r + \gamma_1)$. Because neighboring data sets X, Y differ in only one point (call it z), the ℓ_2 norm of the $k - 1$ other points remains the same and so

$$\left\| \frac{1}{k} \sum_{x \in X} x - \frac{1}{k} \sum_{y \in Y} y \right\|_2 = \left\| \frac{1}{k} z \right\|_2 = \frac{1}{k} \|z\|_2 \leq \frac{2(r + \gamma_1)}{k}$$

as desired. □

.2.3 Proof of Theorem 3.0.6

Proof. We start with Assumption 3.0.3 so we have $\mu \in B_2(\tilde{\mu}_0, r_0)$. Note that the clipping bounds, parameterized by γ_1, in line 4 of Algorithm 4 are set such that, with probability

$1 - \beta_s$, no points are affected by the bounding. Thus, with probability $\geq 1 - \beta_s$:

$$\begin{aligned}
\mathbb{E}(\mu') &= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i + Y\right) \\
&= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i\right) + \mathbb{E}(Y) \\
&= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i\right) \\
&= \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}(\hat{\mu}_i) \\
&= \mu.
\end{aligned}$$

We now consider γ_2, which is set as a $1 - \beta_s$ probability upper bound on the ℓ_2 norm of the privatized mean of k draws from $Q(0, \tilde{\Sigma})$. Conditional on no points being clipped so that $\tilde{\mu}' = \sum_{i=1}^{k}\hat{\mu}_i + Y$, we have

$$1 - \beta_s \leq \mathbb{P}\left(\left\|\frac{1}{k}\sum_{i=1}^{k}\hat{\mu}_i - \mu + Y\right\|_2 \leq \gamma_2\right) \tag{1}$$

$$= \mathbb{P}\left(\|\tilde{\mu}' - \mu\|_2 \leq \gamma_2\right). \tag{2}$$

So, having $\mu \in B_2(\tilde{\mu}_0, r_0)$ implies that $\mathbb{P}(\mu \in B_2(\tilde{\mu}', r')) \geq 1 - 2\beta_s = 1 - \beta_s$. Using the fact that $\sum \beta_s = \beta^\mu$ and a union bound, we proceed by induction over the t steps of the algorithm and see that with probability $1 - \beta^\mu$ we have

$$\forall m \in [t] : \mu \in B_2(\tilde{\mu}_m, r_m)$$

and

$$\forall m \in [t] : \mathbb{E}(\tilde{\mu}_m) = \mu.$$

□

.2.4 Setting γ_1, γ_2

This section is concerned with how to set γ_1, γ_2 in lines 4, 5 of Algorithm 4 for various $Q_{\tilde{\mu}}$. We start with a general statement that works for arbitrary $Q_{\tilde{\mu}}$.

Fact .2.1 (Chebyshev's Inequality)**.** *If X is a d-dimensional random vector with expected value $\mu = \mathbb{E}(X)$ and covariance $\Sigma = \mathbb{E}\left((X - \mu)(X - \mu)^T\right)$, then*

$$\mathbb{P}\left(\sqrt{(X - \mu)^T\Sigma^{-1}(X - \mu)} > t\right) \leq \frac{d}{t^2},$$

provided that Σ is positive definite.

Corollary .2.2. *For any R in Algorithm 4, $\mathbb{P}\left(\|R\|_2 > \sqrt{d/\beta}\right) \leq \beta$.*

Proof. By construction of R, we know that $\mu = 0$ and $\Sigma = I_d$. Let R_j be the j^{th} element of R. Then we can write

$$\sqrt{(R-\mu)^T\Sigma^{-1}(R-\mu)} = \sqrt{R^T R} = \left(\sum_{j=1}^{d} R_j^2\right)^{1/2} = \|R\|_2$$

We can set $t = \sqrt{d/\beta}$ and rewrite Chebyshev's Inequality as

$$\mathbb{P}\left(\|R\|_2 > \sqrt{d/\beta}\right) \leq \beta.$$

□

In practice, it is beneficial to set tighter bounds based on the specified $Q_{\tilde{\mu}}$. This can hypothetically be done via Monte Carlo sampling and empirical CDF inequalities (e.g. with Algorithm 8). However, this can be computationally expensive for γ_1 in particular, as you need at least k/β draws (and often far more) from the random variable to get a proper upper bound.

Some $Q_{\tilde{\mu}}$ also admit analytical bounds, which avoid the need for the costly computation. If $Q_{\tilde{\mu}}$ is multivariate Gaussian, we can use the following:

Fact .2.3 (Lemma 1 of [28])**.** *Let $Q_{\tilde{\mu}}$ be multivariate Gaussian such that $Q_{\tilde{\mu}}(\mu, \Sigma) = N(\mu, \Sigma)$. Then if $R \sim Q_{\tilde{\mu}}(0, I_d)$, we know that*

$$\forall \beta \in (0,1] : \mathbb{P}\left(\|R\|_2 > \sqrt{d + \sqrt{d\log(1/\beta)} + 2\log(1/\beta)}\right) \leq \beta.$$

We present a similar bound for when $Q_{\tilde{\mu}}$ is multivariate Laplace, based heavily on a result from Corollary 3.1 from [41].

Theorem .2.4. *Let $Q_{\tilde{\mu}}$ be multivariate Laplace with mean $\mu = 0$ and covariance $\Sigma = I_d$. Then if $R \sim Q_{\tilde{\mu}}(0, I_d)$, we know that*

$$\forall \beta \in (0,1] : \mathbb{P}\left(\|R\|_2 > \sqrt{e \cdot d\log^2(\beta)}\right) \leq \beta,$$

where $e \approx 2.718$ is Euler's number.

Proof. We start by noting that $\|R\|_2 = \left(\sum_{j=1}^{d} R_j^2\right)^{1/2}$. We know that the R_j are Laplace with mean 0 and variance 1, and thus $\forall j \in [d] : R_j^2 \sim Weibull(\lambda = 1/2, k = 1/2)$. For ease of notation, we'll call $X_j = R_j^2$.

We now define sub-Weibull random variables, as is done in [41]. We call a random variable X_j *sub-Weibull* with tail parameter θ if there exists $\theta, a, b > 0$ such that

$$\forall x > 0 : \mathbb{P}\left(|X_j| \geq x\right) \leq a\exp\left(-bx^{1/\theta}\right).$$

For context, sub-Gaussian random variables are sub-Weibull with $\theta = 1/2$, sub-Exponentials are sub-Weibull with $\theta = 1$, and Weibull random variables themselves are sub-Weibull with $\theta = 2$.

We can state an alternative condition, also from [41], that X_j is sub-Weibull with tail parameter θ if

$$\exists c > 0 \text{ s.t. } \forall t \geq 1 : \|X_j\|_t \leq ct^\theta.$$

Our goal is to find the smallest c that holds for Weibull random variables in particular. We recall that $X_j \sim Weibull(\lambda = 1/2, k = 1/2)$ and $\theta = 2$. Thus, for all $t \geq 1$:

$$\begin{aligned} & \|X_j\|_t \leq ct^\theta \\ \iff & \left(\mathbb{E}\left(|X_j|^t\right)\right)^{1/t} \leq ct^\theta \\ \iff & \lambda\Gamma\left(\frac{t}{k}+1\right)^{1/t} \leq ct^\theta \qquad (3) \\ \iff & \frac{1}{2t^2}\Gamma(2t+1)^{1/t} \leq c. \qquad (4) \end{aligned}$$

Line 3 follows by using the MGF of a Weibull random variable, and line 4 follows by plugging in the parameter values.

Our goal is to find the smallest c such that $\|X\|_t \leq ct^\theta$ for all $t \geq 1$. The lefthand side of line 4 is decreasing in t for $t \geq 1$, so finding the smallest possible c for $t = 1$ will be sufficient for all $t \geq 1$. Plugging in $t = 1$, we get $c = 1$.

We can finally appeal to Corollary 3.1 from [41], which states that if $X_1, \ldots, X_d$ are i.i.d. Weibull random variables with tail parameter θ, then for all $x \geq dK_\theta$ we have

$$\mathbb{P}\left(\left|\sum_{j=1}^d X_j\right| \geq x\right) \leq \exp\left(-\left(\frac{x}{K_\theta d}\right)^{1/\theta}\right)$$

for $K_\theta = ec$. Plugging in the $c = 1$ we found for Weibull random variables yields

$$\mathbb{P}\left(\left|\sum_{j=1}^d X_j\right| \geq x\right) \leq \exp\left(-\left(\frac{x}{e \cdot d}\right)^{1/\theta}\right).$$

We want the probability to be less than β, so we sub this in and get

$$\mathbb{P}\left(\left|\sum_{j=1}^d X_j\right| \geq e \cdot d \log^2(\beta)\right) \leq \beta.$$

We note that $\|X\|_2 = \sqrt{\left|\sum_{j=1}^d X_j\right|}$, so setting the bound at $\sqrt{e \cdot d \log^2(\beta)}$ gives our desired result. □

.3 Parameter Estimation

.3.1 Proof of Theorem 4.0.1

Proof. Our goal is to find weights $\{A_m\}_{m\in[t]}$ with $A_m \in \mathbb{R}^{d\times d}$ such that the Löwner order of $Cov\left(\sum_{m=1}^t A_m \hat{\tau}_m\right)$ is minimized. Because we want our weighted estimator to remain unbiased, we restrict ourselves to sets of A_m such that $\sum_{m=1}^t A_m = I_d$.

We note that the A_m are constants and $\hat{\tau}_m$ are independent, so

$$\begin{aligned} Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) &= \sum_{m=1}^{t} Cov\left(A_m \hat{\tau}_m\right) \\ &= \sum_{m=1}^{t} A_m^T Cov\left(\hat{\tau}_m\right) A_m. \end{aligned}$$

Assume $\hat{\tau}_m \in \mathbb{R}^d$ and let $\{B_m\}_{m\in[t]}$ with $B_m \in \mathbb{R}^{d\times d}$ be an arbitrary weighting. Then we can write

$$\begin{aligned} & Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) \preceq Cov\left(\sum_{m=1}^{t} B_m \hat{\tau}_m\right) \\ \iff & \forall v \in \mathbb{R}^d \setminus \{0\} : v^T Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) v \leq v^T Cov\left(\sum_{m=1}^{t} B_m \hat{\tau}_m\right) v. \end{aligned}$$

Note that the quantities on the righthand side of the statement above are scalars, so we have translated the problem of finding a minimal Löwner bound into minimizing a one-dimensional quantity.

Let $v \in \mathbb{R}^d \setminus \{0\}$ be arbitrary. We now have a one-dimensional constrained optimization problem; we want to find $\{A_m\}_{m\in[t]}$ which minimizes $v^T Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) v$ subject to $\sum_{m=1}^{t} A_m = I_d$. We can solve this using a Lagrange multiplier.

We write

$$\mathcal{L}\left(\{A_m\}_{m\in[t]}, \lambda\right) = v^T Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) v - \lambda v^T \left(\sum_{m=1}^{t} A_m - I_d\right) v$$

and differentiate with respect to A_m. Recall that $Cov\left(\hat{\tau}_m\right) = S_m$. Then we have

$$\begin{aligned} \frac{\partial \mathcal{L}\left(\{A_m\}_{m\in[t]}, \lambda\right)}{\partial A_m} &= \frac{\partial v^T Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right) v - \lambda v^T \left(\sum_{m=1}^{t} A_m - I_d\right) v}{\partial A_m} \\ &= \frac{\partial \left(\sum_{m=1}^{t} v^T A_m^T Cov\left(\hat{\tau}_m\right) A_m v\right) - \lambda v^T \left(\sum_{m=1}^{t} A_m - I_d\right) v}{\partial A_m} \\ &= S_m A_m v v^T + S_m^T A_m v v^T - \lambda v v^T \qquad (5) \\ &= 2\left(S_m A_m - \lambda I_d\right) v v^T. \end{aligned}$$

(5) comes from a matrix calculus identity that for vectors a, b and matrix C all independent of X, $\frac{\partial (Xa)^T C (Xb)}{\partial X} = CXba^T + C^T Xab^T$ and noting that the partial with respect to A_m influences the sum only in the m^{th} term.

We set this to 0 to find a stationary point.

$$\begin{aligned} 0 &= 2\left(S_m A_m - \lambda I_d\right) v v^t \\ \lambda I_d v v^T &= S_m A_m v v^t \\ A_m &= \lambda S_m^{-1} I_d v v^T (v v^T)^{-1} \\ &= \lambda S_m^{-1}. \end{aligned}$$

We know from our constraint that $\sum_{m=1}^{t} A_m = I_d$, so

$$\sum_{m=1}^{t} \lambda S_m^{-1} = I_d$$
$$\lambda = \left(\sum_{m=1}^{t} S_m^{-1} \right)^{-1},$$

and thus our stationary point is achieved at $A_m = \left(\sum_{m=1}^{t} S_m^{-1}\right)^{-1} S_m^{-1}$.

We have shown that choosing A_m in this way achieves a stationary point, but we want to show that it is a global minimum. For that, we need to check the second partial derivative test, which states that our stationary point is a global minumum if $\frac{\partial^2 \mathcal{L}\left(\{A_m\}_{m\in[t]}, \lambda\right)}{\partial^2 A_m}$ is PD.

We first note that

$$\begin{aligned}\frac{\partial^2 \mathcal{L}\left(\{A_m\}_{m\in[t]}, \lambda\right)}{\partial^2 A_m} &= \frac{\partial}{\partial A_m} 2\left(S_m A_m - \lambda I_d\right) vv^T \\ &= 2(vv^T) \otimes S_m,\end{aligned}$$

where $\otimes$ is the Kronecker product.

We know vv^T is PD, because $\forall z \in \mathbb{R}^d \backslash \{0\}$ we get $z^T vv^T z = (z^T v)(v^T z) = (v^T z)^T (v^T z) > 0$. The strict inequality comes because we know that both v and z are non-zero. We know S_m is PD by assumption and that, in general, if a matrix Y is PD then so is $2Y$. Finally, the Kronecker product of PD matrices is also PD, so $2(vv^T) \otimes S_m$ is PD and our second partial derivative condition is met. So $\mathcal{L}$ is convex and our local minimum is also a global minimum. Thus, our choice of A_m achieves the $Cov\left(\sum_{m=1}^{t} A_m \hat{\tau}_m\right)$ with minimal Löwner order. □

.3.2 Privately estimating $\hat{\Sigma}$

Proof of Theorem 4.0.2

Proof. From the proof in Appendix .2.3, we know that we have a $1 - \beta^{\tilde{V}}$ probability guarantee of not clipping any points during the t steps of our estimation algorithm. In this event, our private mean estimation just involves calculating the non-private mean and adding Gaussian noise, so for $Y_m \sim N\left(0, \vec{\sigma}_m^2 I_d\right)$ we say

$$\mathbb{P}\left(\forall m \in [t] : \tilde{V}_m = \hat{V} + Y_m\right) \geq 1 - \beta^{\tilde{V}}.$$

Now, we can view each $\tilde{V}_m$ as a realization of an unbiased estimator of $\hat{V}$ with multivariate Gaussian error and covariance $\vec{\sigma}_m^2 I_d$. Because we have t unbiased estimators, we use a standard precision-weighting argument to combine them into a new estimate in Equation 4.1 which has covariance $\frac{1}{\sum_{m=1}^{t} 1/\vec{\sigma}_m^2} I_d$. This estimator remains unbiased (with high-probability) by linearity of expectation and remains multivariate Gaussian because the sum of multivariate Gaussians is multivariate Gaussian. We imagine drawing from

the noise distribution of this new combined estimator and in Equation 4.2 get $1 - \beta^{ub}/d$ probability upper bounds on a draw from each dimension of the distribution. By a union bound over the d dimensions and adding the failure probability induced by any clipping, we then have

$$\mathbb{P}\left(\forall j \in [d] : \tilde{V}_j + \vec{\gamma}_j \geq \hat{V}_j\right) \geq 1 - \beta^{\tilde{V}} - \beta^{ub}. \tag{6}$$

Finally, recall from Assumption 2.0.1 that $\mathbb{P}\left(\hat{\Sigma}^{BLB} \succeq \hat{\Sigma}\right) = 1$. This implies that each diagonal element of $\hat{\Sigma}^{BLB}$ is at least as large as the corresponding diagonal element of $\hat{\Sigma}$, and so an upper bound on $\hat{V}_j$ is also an upper bound on $\hat{\Sigma}_{j,j}$. □

Proof of Theorem 4.0.3

Proof. We use the same proof structure as in Appendix .3.2, up until the point of upper bounding our covariance, as it is no longer sufficient to simply upper bound each element of the diagonal of $\hat{\Sigma}^{BLB}$; instead, we want a Löwner upper bound.

Let $\hat{S}_i$ be the i^{th} empirical covariance estimate and the unflattened version of $\hat{S}_i^\flat$. We know that

$$\mathbb{P}\left(\forall m \in [t] : \tilde{S}_m^\flat = \hat{S}^\flat + Y\right) \geq 1 - \beta^{\tilde{S}},$$

for $Y_m \sim N(0, \vec{\sigma}_m^2 I_d)$. All arguments that follow are conditional on this $1 - \beta^{\tilde{S}}$ probability event.

As before, we use our precision-weighting argument to create a combined estimator $\tilde{S}^\flat$ and by composition of Gaussians, we get that

$$\tilde{S}^\flat \sim N\left(\hat{S}^\flat, \frac{1}{\sum_{m=1}^t \vec{\sigma}_m^2} I_d\right).$$

We unflatten this to create $\tilde{S}$ where

$$\tilde{S} = \hat{S} + \sigma_M^2,$$

and $\sigma_M^2 \in \mathbb{R}^{d \times d}$ is the unflattened version of $\frac{1}{\sum_{m=1}^t \vec{\sigma}_m^2}$.

We then find γ such that

$$\mathbb{P}\left(\|\sigma_M^2\|_2 \leq \gamma\right) \geq 1 - \beta^{ub}$$

using Algorithm 6.

Letting $\lambda_i(X) \leq \ldots, \leq \lambda_d(X)$ be the eigenvalues of an arbitrary symmetric matrix X, we let Y be another symmetric matrix and say

$$\forall j \in [d] : \lambda_j(X) + \lambda_d(Y) \leq \lambda_j(X + Y).$$

This is a corollary of the Courant-Fischer min-max theorem.

Thus, we have

$$\begin{aligned}
&\tilde{S} = \hat{S} + \sigma_M^2 \\
\iff &\tilde{S} - \sigma_M^2 = \hat{S} \\
\iff &\tilde{S} + \sigma_M^2 = \hat{S} \quad (\sigma_M^2 \text{ is a r.v. symmetric about } 0) \\
\implies &\forall j \in [d] : \lambda_j\left(\tilde{S}\right) + \lambda_d\left(\sigma_M^2\right) \geq \lambda_j\left(\hat{S}\right) \\
\implies &\forall j \in [d] : \lambda_j\left(\tilde{S}\right) + \gamma \geq \lambda_j\left(\hat{S}\right) \\
\implies &\tilde{S} + \gamma I_d \succeq \hat{S}.
\end{aligned}$$

Reintroducing our two sources of failure and letting $\tilde{\Sigma} = \tilde{S} + \gamma I_d$, we get

$$\mathbb{P}\left(\tilde{\Sigma} \succeq \hat{\Sigma}^{BLB}\right) \geq 1 - \beta^{\tilde{S}} - \beta^{ub}.$$

We appeal to Assumption 2.0.1 and the transitivity of the Löwner order to replace $\hat{\Sigma}^{BLB}$ with $\hat{\Sigma}$ above and get our result. □

.3.3 Proof of Theorem 4.0.4

Proof. We know that we don't clip any points in Algorithm 2 with probability $1 - \beta^{\tilde{\theta}}$, so for $Y \sim N\left(0, \vec{\sigma}_m^2 I_d\right)$ we say

$$\mathbb{P}\left(\forall m \in [t] : \tilde{H}_m = \hat{\theta}^{BLB} + Y\right) \geq 1 - \beta^{\tilde{\theta}}.$$

Thus, we have

$$\mathbb{P}\left(\forall m \in [t] : \tilde{\theta}_m \sim N\left(\hat{\theta}^{BLB}, \vec{\sigma}_m^2 I_d\right)\right) \geq 1 - \beta^{\tilde{\theta}}.$$

We combine these t estimators into a single precision-weighted estimator using Fact 4.0.1. This estimator remains unbiased (with high-probability) by using linearity of expectation and that $\mathbb{E}\left(\hat{\theta}^{BLB}\right) = \mathbb{E}\left(\hat{\theta}\right) = \theta$ and remains multivariate Gaussian because the sum of multivariate Gaussians is multivariate Gaussian. □

.4 PSD Projection

It is possible that, after adding privatizing noise to our covariance estimate, it is no longer positive semidefinite (PSD). We show that, if this happens, we can project the matrix back to the PSD cone without losing the guarantees we need regarding the Löwner order of the covariance matrix.

Note that λ' is the set of eigenvalues of M'. We then have the following theorem.

Theorem .4.1. *Let $M \in \mathbb{R}^{m \times m}$. The PSD projection of M described in Algorithm 5 produces a matrix $M' \in \mathbb{R}^{m \times m}$ that is PSD and respects $M' \succeq M$.*

Algorithm 5 PSD Projection of a matrix

Input: Matrix $M \in \mathbb{R}^{m \times m}$, minimum eigenvalue $\epsilon \geq 0$
Output: PSD matrix $M' \in \mathbb{R}^{m \times m}$

procedure PSDProjection(M, ϵ)
  Calculate eigenvectors $\lambda = [\lambda_1, \ldots, \lambda_m]$ and matrix of eigenvectors $Q \in \mathbb{R}^{m \times m}$ such that $M = Q \operatorname{diag}(\lambda) Q^T$
  $\lambda' = [\lambda'_1, \ldots, \lambda'_m]$ where $\lambda'_i = \max(\epsilon, \lambda_i)$
  $M' \leftarrow Q \operatorname{diag}(\lambda') Q^T$
  return M'

Proof. The fact that M' is PSD follows immediately from the fact that we know $\min_{i \in [m]} \lambda'_i \geq 0$ by construction. Non-negative eigenvalues are a necessary and sufficient condition for a matrix to be PSD, so M' is PSD.

We say that $M' \succeq M$ if and only if $M' - M$ is a PSD matrix. Recall from Algorithm 5 that we can write $M = Q \operatorname{diag}(\lambda) Q^T$ and $M' = Q \operatorname{diag}(\lambda') Q^T$. Thus,

$$\begin{aligned} M' - M &= Q \operatorname{diag}(\lambda') Q^T - Q \operatorname{diag}(\lambda) Q^T \\ &= Q \left(\operatorname{diag}(\lambda') - \operatorname{diag}(\lambda) \right) Q^T \\ &= Q \operatorname{diag}(\lambda' - \lambda) Q^T. \end{aligned}$$

Because we know that $\forall i \in [m] : \lambda'_i - \lambda_i \geq 0$, we know that $M' - M$ has non-negative eigenvalues and thus $M' \succeq M$. □

We note that if $\epsilon > 0$, then the projection creates a positive definite matrix.

.5 Confidence Intervals

.5.1 Proof of Theorem 4.0.5

Proof. We assume that our estimation of $\tilde{\theta}$ and $\tilde{\Sigma}$ worked as described at the top of Section 4.0.3, which comes with a $1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$ probability guarantee.

Let $j \in [d]$ and $\alpha \in (0, 1)$ be arbitrary and let Z be drawn from the j^{th} dimension of $Q_{\tilde{\theta}}$. Then Algorithm 8 returns $\left(ci^l_j, ci^u_j\right)$ such that $\mathbb{P}\left(Z < ci^j_l\right) \leq \alpha/2$ and $\mathbb{P}\left(Z > ci^j_u\right) \leq \alpha/2$.

We know that $\mathbb{E}(Z) = \hat{\theta}_j$ and Algorithm 8 produces an interval around this expectation with length c on each side. So we have

$$\mathbb{P}\left(Z \in \hat{\theta}_j \pm c\right) \geq 1 - \alpha \iff \mathbb{P}\left(\hat{\theta}_j \in Z \pm c\right) \geq 1 - \alpha,$$

which gives us our confidence interval. □

.5.2 Proof of Corollary 4.0.6

Proof. This follows the same line of reasoning as the proof in Appendix .5.1, except we fold the failure probability directly into α_j. □

.6 Upper-Bounding Random Variables

.6.1 HPUB Algorithm

Algorithm 6 High-probability upper bound on random variable

Input: random variable X, failure probability α, number of simulations n, precision $\tau \in (0, 1)$
Output: u such that $\mathbb{P}(X \leq u) \geq 1 - \alpha$

procedure HPUB(X, α, n, τ)
  $n = \max(n, \lceil 1/\alpha \rceil))$
  while True **do**
    $\alpha_1^B = 0$ ▷ initialize target empirical CDF value
    $m = \lceil \frac{1}{\tau} \rceil - 1$
    for $j \in [m-1]$ **do** ▷ brute force search
      $\alpha_1 = \frac{j}{m}\alpha, \alpha_2 = \frac{\alpha - \alpha_1}{2}$
      $c = n(1 - \alpha_1)$
      $ci_l = F^{-1}_{B(c, n-c+1)}(\alpha_2)$ ▷ lower binomial confidence bound
      if $ci_l \geq 1 - \alpha - \alpha_2$ **then**
        $\alpha_1^B = \max(\alpha_1, \alpha_1^B)$
    if $\alpha_1^B = 0$ **then**
      $n = n + 1000$
    else
      $k \leftarrow \lceil n(1 - \alpha_1) \rceil$
  Sample $\{x_i\}_{i \in [n]}$ from X ▷ sample from random variable
  Sort $\{x_i\}_{i \in [n]}$ such that $x_i \leq x_j$ for $i < j$
  return x_k

Algorithm 7 Approximate upper bound on random variable

Input: random variable X, failure probability α, number of simulations n **Output:** u such that $\mathbb{P}(X \leq u) \overset{?}{\geq} 1 - \alpha$

procedure ApproxUB(X, α, n)
  $n = \max(n, \lceil 1/\alpha \rceil))$
  Sample $\{x_i\}_{i \in [n]}$ from X ▷ sample from random variable
  $k \leftarrow \lceil n(1 - \alpha_1) \rceil$
  Sort $\{x_i\}_{i \in [n]}$ such that $x_i \leq x_j$ for $i < j$
  return x_k

We sometimes refer to this as HPUB(X, α), leaving n, τ implicit. For very small α, running Algorithm 6 as written is unlikely to be computationally feasible. In this case, it is possible to use the Algorithm 7, which doesn't yield the desired high-probability guarantee but works pretty well in practice.

Lemma .6.1 (Binomial Confidence Interval [11])**.** *Let c be a realization of a random variable $C \sim Binomial(n, p)$. Let $B_{a,b} \sim Beta(a, b)$ be a random variable representing the Beta distribution with shape parameters a, b and let $F_{B(a,b)}$ be its CDF. Then, for any $\alpha \in (0, 1)$,*

$$\mathbb{P}\left(F^{-1}_{B(c, n-c+1)}(\alpha) \leq p\right) \geq 1 - \alpha.$$

Theorem .6.2. *For a random variable X and arbitrary $\alpha \in (0, 1)$, Algorithm 6 returns u such that*

$$\mathbb{P}(X \leq u) \geq 1 - \alpha.$$

Proof. Our goal is to find a u such that $\mathbb{P}(X \leq u) \geq 1 - \alpha$. Because we cannot sample infinitely from X, we'll split our failure probability α into two pieces, α_1 and α_2 such that $\alpha_1 + \alpha_2$ is strictly less than α.

Say we have $\{X_i\}_{i \in [n]}$ for $X_i \sim X$ and associated realizations of those random variables $\{x_i\}_{i \in [n]}$ such that $i \leq j \implies x_i \leq x_j$. Then if $F_X(x) = \mathbb{P}(X \leq x)$, we'll say that

$$\hat{F}_X(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_i \leq x).$$

Say we knew that $z = F^{-1}(1 - \alpha_1)$ for some z. Then,

$$\begin{aligned} z &= F^{-1}(1 - \alpha_1) \\ \iff F(z) &= 1 - \alpha_1 \\ \iff \hat{F}_X(z) &\sim Binomial(n, 1 - \alpha_1). \end{aligned}$$

We can now work backwards from this. Let $z' = x_{n(1-\alpha_1)}$ and note that $c := \hat{F}_X(z') = n(1 - \alpha_1)$ is a realization of a $C \sim Binomial(n, F_X(z'))$. We then appeal to Lemma .6.1 and say that

$$\mathbb{P}\left(F^{-1}_{B(c,n-c+1)}(\alpha_2) \leq F_X(z')\right) \geq 1 - \alpha_2.$$

In the algorithm, we search over combinations of α_1, α_2 to find the largest α_1 such that $F^{-1}_{B(c,n-c+1)}(\alpha_2)$ that is at least as large as $1 - \alpha + \alpha_2$. If, for a fixed n there is no satisfying combination of α_1, α_2 we increase n and look again.

So we end up with α_1, α_2 such that

$$\mathbb{P}(1 - \alpha + \alpha_2 \leq F_X(z')) \geq 1 - \alpha_2.$$

Composing these two failure probabilities by a union bound, our process yields a z' such that

$$1 - \alpha \leq \mathop{\mathbb{P}}_{X \sim \mathcal{X}, z' \sim \mathcal{Z}'}(X \leq z')$$

as desired.

□

Algorithm 8 Confidence Interval Simulation

Input: distribution $Q(\mu, \Sigma)$ from a symmetric location-scale family, desired confidence levels $\{\alpha_j\}_{j\in[d]} \in (0,1)^d$

Output: Interval that contains $1-\alpha$ of the mass in each dimension, with at most $\alpha/2$ outside on either end

procedure ConfidenceIntervalSimulation(Q, j, α)
    $Z \sim Q(\mu, \Sigma)$
    Let Z_j be the j^{th} element of the random vector Z
    for $j \in [d]$ **do**
        $ci_j^u = \text{HPUB}\left(Z_j, 1-\alpha_j/2\right)$
        $c = ci_j^u - \mu$
        $ci_j^l = \mu - c$
    return $\left\{ci_j^l, ci_j^u\right\}_{j\in[d]}$