Direct Sampling in Bayesian Regression Models with Additive Disclosure Avoidance Noise

Andrew M. Raim

Center for Statistical Research and Methodology
U.S. Census Bureau

CSRM Seminar 4/6/2021

Disclaimer

This article is released to inform interested parties of ongoing research and to encourage discussion of work in progress. Any views expressed are those of the author and not those of the U.S. Census Bureau.

Overview

- This work revisits the direct sampler proposed by Walker, Laud, Zantedeschi, and Damien (2011).
- It was motivated by Bayesian regression models where the data have added noise for the purpose of disclosure avoidance.
- Differential privacy (DP) has become increasingly popular for its ability to mathematically bound risks to unwanted disclosure in the released data (Dwork and Roth, 2014).
- The U.S. Census Bureau is evaluating use of DP for public release of the data collected in the 2020 Decennial Census (Abowd, 2018; Garfinkel et al., 2018).
- Several noise mechanisms under consideration achieve privacy by adding noise variates.
- Relatively simple regression models including DP noise may lead to conditional distributions which are nontrivial to sample.
- The direct sampler will provide a reliable way to draw from these conditionals and therefore construct a Gibbs sampler.

Overview

- The most basic implementation of direct sampling (BD sampler) described in Walker et al. (2011) is easy to implement, but poor performance is possible under the situations we encounter.
- Raim (2021b) proposes a “customized direct sampler” (CD sampler), which addresses some of these issues. It assumes certain restrictions on target distributions.
- We will:
  1. Briefly review DP preliminaries.
  2. Review the BD sampler and problematic situations.
  3. Discuss the proposed CD sampler.
  4. Show the sampler in the context of a regression modeling application.
  5. Walk through a simulation study comparing inference based on a noisy release and the original sensitive data.

Acknowledgement

Teammates on disclosure avoidance modeling project are:

- Scott Holan (ADRM),
- Kyle Irimata (CSRM),
- Ryan Janicki (CSRM), and
- James Livsey (CSRM).

Differential Privacy Preliminaries

- The Laplace mechanism Dwork and Roth (2014, Chapters 2–3) is one of the fundamental mechanisms in differential privacy (DP).

- A few definitions:
  1. Let $X \sim \text{Lap}(\mu, \lambda)$ denote a random variable with Laplace distribution $g(x) = \frac{1}{2\lambda} e^{-|x-\mu|/\lambda}$.
  2. Privacy loss budget $\epsilon > 0$ quantifies how much protection the data will receive.
  3. Histogram $\boldsymbol{x}$ contains distinct data values and their frequencies.
  4. A query f maps an $\boldsymbol{x}$ to $\mathbb{R}^k$.
  5. The L_1 sensitivity of a query f is

$$\Delta f = \max \|f(\boldsymbol{x}) - f(\boldsymbol{y})\|_1, \quad \text{s.t. } \|\boldsymbol{x} - \boldsymbol{y}\|_1 = 1.$$

- Given a privacy loss budget ϵ, a histogram $\boldsymbol{x}$, and query function f, the Laplace mechanism is

$$M_{\text{Lap}}(\boldsymbol{x} \mid f, \epsilon) = f(\boldsymbol{x}) + \boldsymbol{\xi}, \quad \xi_1, \ldots, \xi_k \overset{\text{iid}}{\sim} \text{Lap}(0, \Delta f/\epsilon).$$

Differential Privacy Preliminaries

- A randomized algorithm M is said to be (ϵ, δ)-differentially private if

$$\mathrm{P}[M(\boldsymbol{x}) \in S] \leq e^{\epsilon}\, \mathrm{P}[M(\boldsymbol{y}) \in S] + \delta$$

  for all $S \subseteq \mathrm{range}(M)$ and all histograms $\boldsymbol{x}, \boldsymbol{y}$ such that $\|\boldsymbol{x} - \boldsymbol{y}\|_1 \leq 1$.

- **Theorem.** The Laplace mechanism is $(\epsilon, 0)$ differentially private.

Differential Privacy Preliminaries

Proof. Let $\boldsymbol{x}, \boldsymbol{y}$ be such that $\|\boldsymbol{x} - \boldsymbol{y}\| \leq 1$. Let p_x and p_y denote the density functions of $M_{\text{Lap}}(\boldsymbol{x} \mid f, \epsilon)$ and $M_{\text{Lap}}(\boldsymbol{y} \mid f, \epsilon)$. For any $\boldsymbol{\xi} \in \mathbb{R}^k$,

$$\begin{aligned}
\frac{p_x(\boldsymbol{\xi})}{p_y(\boldsymbol{\xi})} &= \prod_{i=1}^{k} \frac{\exp[-\epsilon|f(\boldsymbol{x})_i - \xi_i|/\Delta f]}{\exp[-\epsilon|f(\boldsymbol{y})_i - \xi_i|/\Delta f]} \\
&= \prod_{i=1}^{k} \exp\left\{ \frac{|f(\boldsymbol{y})_i - \xi_i| - |f(\boldsymbol{x})_i - \xi_i|}{\Delta f/\epsilon} \right\} \\
&\leq \prod_{i=1}^{k} \exp\left\{ \frac{|f(\boldsymbol{y})_i - f(\boldsymbol{x})_i|}{\Delta f/\epsilon} \right\} \\
&= \exp\left\{ \frac{\|f(\boldsymbol{y}) - f(\boldsymbol{x})\|_1}{\Delta f/\epsilon} \right\} \leq \exp(\epsilon).
\end{aligned}$$

Then for measureable $S \subseteq \mathbb{R}^k$,

$$p_x(\boldsymbol{\xi}) \leq e^{\epsilon} p_y(\boldsymbol{\xi}) \implies \mathrm{P}[M_{\text{Lap}}(\boldsymbol{x} \mid f, \epsilon) \in S] \leq e^{\epsilon}\, \mathrm{P}[M_{\text{Lap}}(\boldsymbol{y} \mid f, \epsilon) \in S].$$

Other Additive Noise Mechanisms

- Other mechanisms add randomly generated noise to a query to protect privacy.
- Gaussian mechanism (Dwork and Roth, 2014, Appendix A) adds $\text{N}(0, \tau^2)$ noise.
- Double Geometric mechanism (Ghosh et al., 2012) adds noise from $\text{DGeom}(\rho)$, whose density is $f(x) = \frac{\rho}{2-\rho}(1-\rho)^{|x|} \cdot \text{I}(x \in \mathbb{Z})$.
- Discrete Gaussian mechanism (Canonne et al., 2020) adds noise from the density $f(x) \propto \exp\left\{-x^2/2\tau^2\right\} \cdot \text{I}(x \in \mathbb{Z})$.
- Proofs for other cases are more involved than Laplace mechanism, and more complicated criteria are usually obtained.
- A user of the protected data has full knowledge of the mechanism, including parameters (Gong, 2020).

Weighted Densities

- To draw from a weighted density

$$f(x) = \frac{w(x)g(x)}{\psi}, \quad x \in \Omega.$$

- $\psi = \int_\Omega w(x)g(x)d\nu(x)$ is the normalizing constant.
- Ω is the support of random variable $x \sim f(x)$.
- $\nu(\cdot)$ is a dominating measure so that x may be discrete or continuous.
- f can be considered a modified version of a base distribution g. The weight function $w : \Omega \to [0, \infty)$ emphasizes or deemphasizes parts of the space.

General Bayesian Example

- Consider a standard Bayesian model

$$\boldsymbol{y} \sim f(\boldsymbol{y} \mid \boldsymbol{\theta}), \quad \boldsymbol{\theta} \sim f(\boldsymbol{\theta}).$$

- The posterior distribution

$$f(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{f(\boldsymbol{y} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\boldsymbol{y})}$$

  is a weighted density.

- Here it seems most natural to take $f(\boldsymbol{\theta})$ as the base distribution and $f(\boldsymbol{y} \mid \boldsymbol{\theta})$ as the weight function.

Disclosure Avoidance Noise Example

- Consider a Bayesian regression model in the form of

$$z_i = y_i + \xi_i, \quad \xi_i \overset{\text{ind}}{\sim} \text{Lap}(0, \lambda_i),$$
$$\log y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \gamma_i, \quad \gamma_i \overset{\text{iid}}{\sim} \text{N}(0, \sigma^2),$$

for $i = 1, \ldots, n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)$ has prior $\boldsymbol{\beta} \sim \text{N}(\boldsymbol{0}, \sigma_\beta^2 \boldsymbol{I})$ and $\sigma^2 \sim \text{IG}(a_\sigma, b_\sigma)$.

- Laplace density with known λ_i,

$$f_{\text{Lap}}(\xi \mid \lambda_i) = \frac{1}{2\lambda_i} e^{-|\xi|/\lambda_i}, \quad \xi \in \mathbb{R},$$

comes from the DP noise.

- The density of a Lognormal random variable $y \sim \text{LN}(\mu, \sigma^2)$ is

$$f_{\text{LN}}(y \mid \mu, \sigma^2) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2\sigma^2} (\log y - \mu)^2 \right\}, \quad y > 0.$$

Disclosure Avoidance Noise Example

Routine Gibbs Steps

- Given $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$, draws for $\boldsymbol{\theta}$ may be derived routinely in two additional Gibbs sampling steps, and are found to have familiar forms.

- $[\beta \mid —] \sim \mathrm{N}_d(\boldsymbol{\vartheta}, \boldsymbol{\Omega}^{-1})$

$$\boldsymbol{\Omega} = \sigma^{-2} \boldsymbol{X}^\top \boldsymbol{X} + \sigma_\beta^{-2} \boldsymbol{I}_d, \quad \boldsymbol{\vartheta} = \boldsymbol{\Omega}^{-1} \left(\sigma^{-2} \sum_{i=1}^n \boldsymbol{x}_i \log y_i \right),$$

where $\boldsymbol{X} = (\boldsymbol{x}_1 \cdots \boldsymbol{x}_n)^\top$ and $\boldsymbol{I}_d$ is the $d \times d$ identity matrix.

- $[\sigma^2 \mid —] \sim \mathrm{IG}(a^*, b^*)$

$$a^* = a_\sigma + \frac{n}{2}, \quad b^* = b_\sigma + \frac{1}{2} \sum_{i=1}^n (\log y_i - \boldsymbol{x}_i^\top \beta)^2.$$

Disclosure Avoidance Noise Example

Weighted Densities

- The joint distribution of all random quantities is

$$f(\boldsymbol{z}, \boldsymbol{\xi}, \beta, \sigma^2) = \left[\prod_{i=1}^n f_{\text{LN}}(z_i - \xi_i \mid \mu_i, \sigma^2) f_{\text{Lap}}(\xi_i \mid 0, \lambda_i) \right] f(\boldsymbol{\theta}),$$

where $\mu_i = \boldsymbol{x}_i^\top \beta$.

- The conditional distribution of $[\xi_i \mid \text{—}]$ is then

$$\begin{aligned} f(\xi_i \mid \text{—}) &\propto f_{\text{LN}}(z_i - \xi_i \mid \mu_i, \sigma^2) f_{\text{Lap}}(\xi_i \mid 0, \lambda_i) \\ &\propto \underbrace{\frac{1}{z_i - \xi_i} \exp\left\{ -\frac{1}{2\sigma^2} [\log(z_i - \xi_i) - \mu_i]^2 \right\} \cdot \text{I}(z_i > \xi_i)}_{w(\xi_i \mid z_i, \mu_i, \sigma^2)} \underbrace{\frac{1}{2\lambda_i} e^{-|\xi_i|/\lambda_i}}_{g(\xi_i \mid \lambda_i)}. \end{aligned}$$

- Its normalizing constant is

$$\int_{-\infty}^{z_i} \frac{1}{z_i - v} \exp\left\{ -\frac{1}{2\sigma^2} [\log(z_i - v) - \mu_i]^2 \right\} \frac{1}{2\lambda_i} e^{-|v|/\lambda_i} dv.$$

Some Relevant Literature

- Bowen and Liu (2020) review noise mechanisms for disclosure avoidance, including some non-additive mechanisms.
- Charest (2011) considers Bayesian modeling under a DP mechanism for binary data. Metropolis-Hastings is used to sample sensitive data within a Gibbs sampler.
- Klein and Sinha (2019) consider generation and analysis of multiply imputed data under noise from a Laplace mechanism, taking very large and very small values to be censored.
- Gong (2019) uses Approximate Bayesian Computation and Monte-Carlo Expectation Maximization to analyze data with additive DP noise.
- For simple linear regression, Gong (2020) provides some theoretical insight about biases when noise mechanism is ignored.
- Evans and King (2020+) propose a version of ordinary least squares regression where estimators are consistent under added DP noise.
- Bernstein and Sheldon (2019) formulate a Gibbs sampler for linear regression with noise from a Laplace mechanism. Noise is drawn as augmented data via a scale mixture of Normals.

Direct Sampling Idea

- Back to our weighted density $f(x) = w(x)g(x)/\psi \cdot \mathrm{I}(x \in \Omega)$.
  1. Let $c = \sup_{x \in \Omega} w(x)$.
  2. Let $A_u = \{x \in \Omega : w(x) > uc\}$.

- **Objective**: augment a random variable u so that $[x, u]$ is easier to draw than x. Especially, avoid computing ψ.

- Assume that $[u \mid x] \sim \text{Uniform}(0, w(x)/c)$, so that

$$f(u \mid x) = \frac{c}{w(x)} \mathrm{I}(0 < u < w(x)/c) = \frac{c}{w(x)} \mathrm{I}(x \in A_u).$$

- The joint density of $[x, u]$ is then

$$f(x, u) = \frac{c}{\psi} g(x) \, \mathrm{I}(x \in A_u).$$

- The marginal density of u is then

$$p(u) = \frac{c}{\psi} \mathrm{P}(A_u), \quad u \in [0, 1], \quad \text{where } \mathrm{P}(A_u) = \int \mathrm{I}(x \in A_u) g(x) d\nu(x).$$

- The distribution of $[x \mid u]$ is then

$$f(x \mid u) = \frac{g(x)}{\mathrm{P}(A_u)} \mathrm{I}(x \in A_u).$$

Direct Sampling Idea

- Now we can take a draw from $[x, u]$ using

$$u \sim p(u) = \frac{c}{\psi} \mathrm{P}(A_u), \quad x \sim f(x \mid u) = \frac{g(x)}{\mathrm{P}(A_u)} \mathrm{I}(x \in A_u).$$

- Here are a few important features about the density $p(u)$.
  1. The support of u is bounded in $[0, 1]$.
  2. $\mathrm{P}(A_u)$ is monotonically nonincreasing in u.
  3. $A_0 \equiv \operatorname{supp} w$ so that $\mathrm{P}(A_0) = \int_\Omega \mathrm{I}(w(x) > 0) g(x) d\nu(x)$.
  4. A_1 is an empty set so that $\mathrm{P}(A_1) = 0$.

Basic Direct Sampler

Drawing from $p(u)$

- To draw $u \sim p(u)$, consider the following.
- Using knot points $u_k = k/N$, compute

$$q(u_k) = \frac{\mathrm{P}(A_{u_k})}{\sum_{\ell=0}^{N} \mathrm{P}(A_{u_\ell})}, \quad k = 0, 1, \ldots, N.$$

  N is prespecified.
- Sample $k \sim \text{Discrete}\Big((0, 1, \ldots, N), (q(u_0), \ldots, q(u_N))\Big)$.
- Given k, sample $u \sim \text{Beta}(k+1, N-k+1)$.
- The density of u is then proportional to

$$\sum_{k=0}^{N} \frac{u^k (1-u)^{N-k}}{B(k+1, N-k+1)} q(u_k) \propto \sum_{k=0}^{N} q(u_k) \binom{N}{k} u^k (1-u)^{N-k},$$

- This can be seen as an approximation to $p(u)$ by Bernstein polynomials (Rivlin, 1981).

Basic Direct Sampler

Drawing from $f(x \mid u)$

- Given u, we must draw x from

$$f(x \mid u) = \frac{g(x)}{\mathrm{P}(A_u)} \mathrm{I}(x \in A_u).$$

- Typically, it is easy to draw from the base distribution $g(x)$.
- Take candidate draws from $x^* \sim g(x)$ and reject until $x^* \in A_u$.

Bernstein Polynomials

- Recall Bernstein's version of the Weierstrass Approximation Theorem (e.g. Resnick, 1999). Let $q : [0, 1] \to \mathbb{R}$ be a continuous function and define the polynomial

$$B_n(x) = \sum_{k=0}^{n} q\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k},$$

  so that $B_n(x) = \mathrm{E}[q(S_n/n)]$ where $S_n = \sum_{i=1}^{n} T_i$ and $T_1, \ldots, T_n$ is an iid sample from $\mathrm{Ber}(x)$.
- Then $\sup_{x \in [0,1]} |B_n(x) - q(x)| \to 0$ as $n \to \infty$.

Some Issues with the BD Sampler

- The support of p may be contained in $[0, u_H]$ for a very small $u_H > 0$.
- The standard Bernstein approximation assumes knots are evenly spaced. This can be less than ideal; e.g., $p(u)$ can be like a step function.
- The simple rejection method to draw from the truncated g can be very inefficient, especially when A_u has small probability under g.

A Customized Direct Sampler

- To address these issues, we propose the following.
- A step function instead of Bernstein polynomials to approximate $p(u)$.
- Focus approximation effort on $[u_L, u_H] \subseteq [0, 1]$, where $p(u)$ is varying.
- Choose the knots sequentially so that each knot placement decreases the approximation error as much as possible.
- Use the CDF method to draw x from $f(x \mid u)$ without rejections.

A Customized Direct Sampler

- We make the following assumptions.
  1. w is unimodal, so that:
     a. We can identify the maximum value c.
     b. $A_u = \{x \in \Omega : w(x) > uc\}$ is an interval with endpoints $\{x_1(u), x_2(u)\}$.
  2. For g,
     a. Exact draws can readily be generated.
     b. Quantiles can be identified.
- Ideally, these operations can be computed with little work.
- These assume a univariate w and g.

United States Census Bureau

Bisection Search

Bisection Search Algorithm.

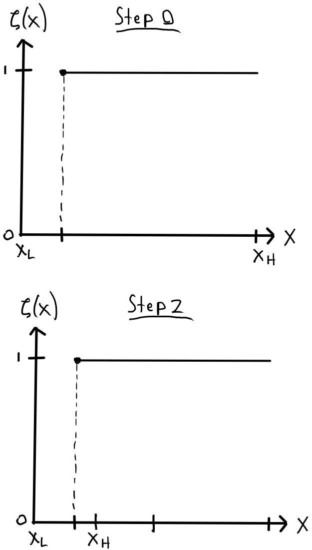
$x = \text{mid}(x_L, x_H)$
while $\text{dist}(x_L, x_H) > \delta$ **do**
  $x_L = \zeta(x) \cdot x_L + [1 - \zeta(x)] \cdot x$
  $x_H = \zeta(x) \cdot x + [1 - \zeta(x)] \cdot x_H$
  $x = \text{mid}(x_L, x_H)$
return x

- $\zeta(x)$ is a step function that activates between the given $[x_L, x_H]$.
- $\text{mid}(x, y)$ is a midpoint function, such as $f(x) = (x + y)/2$.
- $\text{dist}(x, y)$ is a distance function, such as $f(x) = |x - y|$.
- To find the activation point $x^* = \min\{x \in [x_L, x_H] : \zeta(x) = 1\}$.

This is used to find $[u_L, u_H]$ containing the "descent" of $\text{P}(A_u)$.

- u_L is the smallest $u \in [0, 1]$ such that $\text{P}(A_u) < \text{P}(A_0)$.
- u_H is the smallest $u \in [0, 1]$ such that $\text{P}(A_u) = 0$.

Bisection Search



Step Function

- Let $u_0 < \cdots < u_N$ be knot points with $u_0 \equiv u_L$ and $u_N \equiv u_H$.
- To approximate the unnormalized $\mathrm{P}(A_u)$, consider the function

$$h^*(u) = \mathrm{P}(A_{u_0}) \cdot \mathrm{I}(0 \leq u < u_0) + \sum_{j=0}^{N-1} \mathrm{P}(A_{u_j}) \cdot \mathrm{I}(u_j \leq u < u_{j+1}).$$

- A density is obtained using $h(u) = h^*(u)/a$ with

$$a = \int_0^1 h^*(u) du = \mathrm{P}(A_{u_0}) \cdot u_0 + \sum_{j=0}^{N-1} \mathrm{P}(A_{u_j}) \cdot (u_{j+1} - u_j),$$

- Expressions for the CDF and quantile function of h can also be obtained.
- The quantile function can be used to draw from h.
- Because the quantile function is piecewise linear, bisection search can be used to quickly identify the piece containing a given probability.

Approximation Error Bound

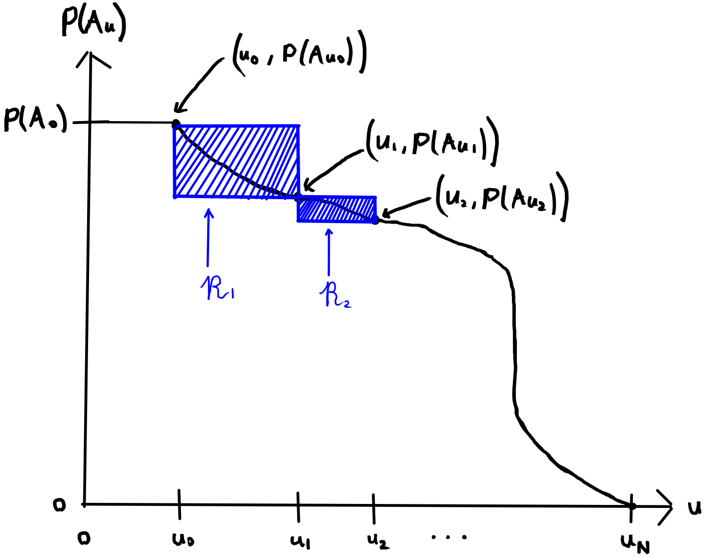
- We can bound the total variation distance between the h and p distributions.
- Let $\mathcal{R}_j$ represent the rectangle in $\mathbb{R}^2$ whose upper-left point is $(u_{j-1}, \mathrm{P}(A_{u_{j-1}}))$ and lower-right point is $(u_j, \mathrm{P}(A_{u_j}))$.
- The area of $\mathcal{R}_j$ is $|\mathcal{R}_j| = \left[\mathrm{P}(A_{u_{j-1}}) - \mathrm{P}(A_{u_j})\right](u_j - u_{j-1})$.

Result

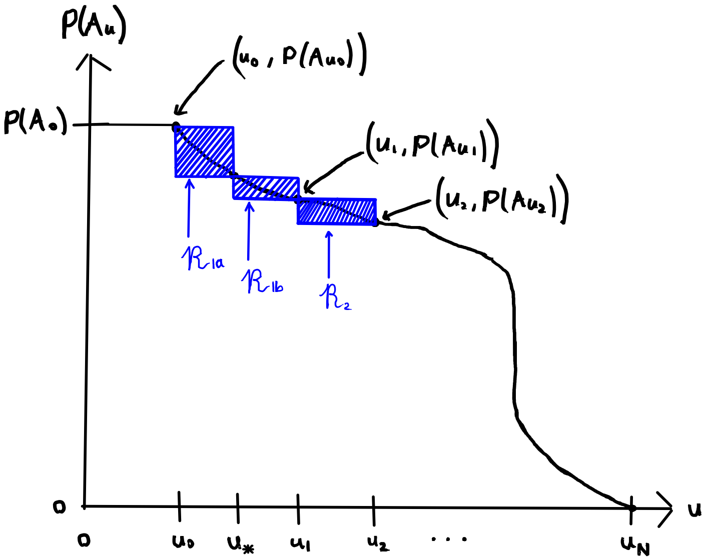
Let $\mathcal{B}$ denote the collection of measurable subsets of $[0,1]$; then

$$\sup_{B \in \mathcal{B}} \left| \int_B h(u)du - \int_B p(u)du \right| \leq \frac{c}{\psi} \sum_{j=1}^{N} |\mathcal{R}_j|.$$

Approximation Error Bound



Approximation Error Bound



Knot Selection

- Equally-spaced knots $u_j = u_L + (j/N)(u_H - u_L)$ are simple and easy to compute, but can fail to capture important features of $p(u)$.
- Our bound motivates selecting the knots $u_1, \ldots, u_{N-1}$ sequentially to reduce the largest $|\mathcal{R}_j|$. This motivates the following algorithm.

Small Rectangles Algorithm.

Let $u^{(0)} = u_L$, and $u^{(1)} = u_H$.
for $i = 1, \ldots, N-1$ **do**
  Let $u_0 < \ldots < u_i$ be sorted $u^{(0)}, \ldots, u^{(i)}$.
  Let $|\mathcal{R}_j| = \{\mathrm{P}(A_{u_{j-1}}) - \mathrm{P}(A_{u_j})\}(u_j - u_{j-1})$ for $j = 1, \ldots, i$.
  Let $j^* = \underset{j=1,\ldots,i}{\operatorname{argmax}} |\mathcal{R}_j|$.
  Let $u^{(i+1)} = \mathrm{mid}(u_{j^*-1}, u_{j^*})$.
Let $u_0 < \ldots < u_N$ be sorted $u^{(0)}, \ldots, u^{(N)}$.
return $(u_0, \ldots, u_N)$.

- The cost of this over equally-spaced knots is increased computation.
- To avoid repeated sorting of the $|\mathcal{R}_j|$'s, we can use a priority queue.

Knot Selection Example

- Recall our conditional distribution from the disclosure avoidance example.

$$
\begin{aligned}
f(\xi \mid —) &\propto f_{\text{LN}}(z - \xi \mid \mu, \sigma^2) f_{\text{Lap}}(\xi \mid 0, \lambda) \\
&\propto \underbrace{\frac{1}{z - \xi} \exp\left\{ -\frac{1}{2\sigma^2} [\log(z - \xi) - \mu]^2 \right\} \cdot \text{I}(z > \xi)}_{w(\xi | z, \mu, \sigma^2)} \underbrace{\frac{1}{2\lambda} e^{-|\xi|/\lambda}}_{g(\xi|\lambda)}.
\end{aligned}
$$

(Subscript i has been omitted.)

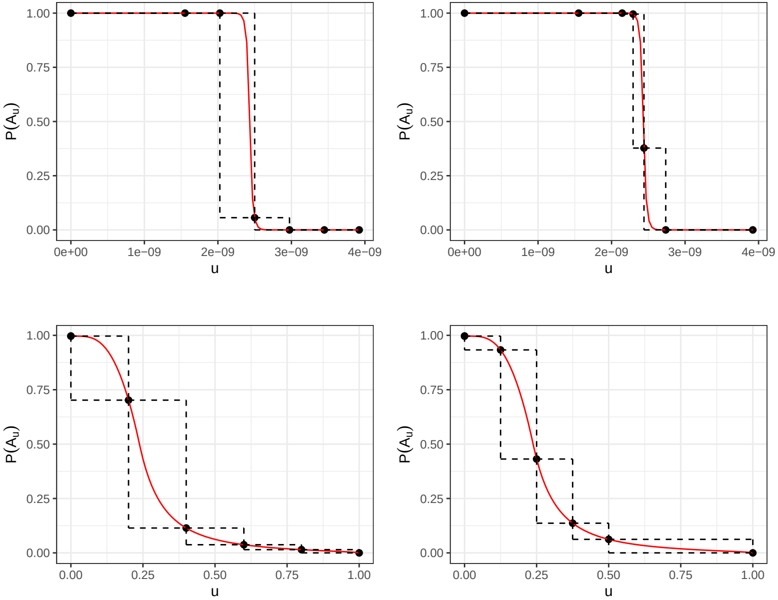


Figure: Plots of $\mathrm{P}(A_u)$ (—) using Lognormal(0, 1) weight function and Laplace(0, 0.4) base distribution. Top plots use $z = 200$ and bottom plots use $z = 2$. $N + 1 = 6$ knots (•) are shown with equal steps (left) and Small Rectangles (right).

Accept-Reject Algorithm

- We constructed h^* so that $h^*(u) \geq \text{P}(A_u)$ for all $u \in [0, 1]$.
- This motivates using h^* as an envelope for rejection sampling, to take exact draws from $p(u)$.
- For any $u \in [0, 1]$,

$$\frac{\text{P}(A_u)}{h^*(u)} \leq 1.$$

- Taking $v \sim \text{Uniform}(0, 1)$, the candidate $u \sim h(u)$ is accepted as a draw from $p(u)$ if $v < \frac{\text{P}(A_u)}{h^*(u)}$. Otherwise, repeat.
- Normalizing the ratio of densities yields:
  1. $\frac{\psi/c}{a}$ is the probability of accepting each proposed u.
  2. $\frac{a}{\psi/c}$ is the expected number of proposals needed for one acceptance.
- A rejected u may be added to the knot points to improve the envelope, at the cost of more bookkeeping.

Drawing from $[x \mid u]$

- Assuming unimodal w, $\mathrm{I}(x \in A_u) = \mathrm{I}(x_1(u) < x < x_2(u))$, and

$$f(x \mid u) = \frac{g(x)}{\mathrm{P}(A_u)} \mathrm{I}(x \in A_u) = \frac{g(x)\,\mathrm{I}(x_1(u) < x < x_2(u))}{G(x_2(u)-) - G(x_1(u))}.$$

- The associated CDF is

$$F(x \mid u) = \frac{G(x) - G(x_1(u))}{G(x_2(u)-) - G(x_1(u))}, \quad x_1(u) < x < x_2(u).$$

- The quantile function is

$$F^{-}(\varphi \mid u) = G^{-}((b - a)\varphi + a)$$

where G^{-} is the quantile function for density g, $a = G(x_1(u))$, and $b = G(x_2(u)-)$.

- An exact draw from $f(x \mid u)$ can be obtained via the inverse CDF method: draw $v \sim \mathrm{Uniform}(0, 1)$ and take $x = F^{-}(v \mid u)$.

Back to Disclosure Avoidance Example

- We wish to draw from

$$f(\xi) \propto \underbrace{\frac{1}{z-\xi} \exp\left\{-\frac{1}{2\sigma^2}[\log(z-\xi)-\mu]^2\right\} \cdot \mathrm{I}(z>\xi)}_{w(\xi|z,\mu,\sigma^2)} \underbrace{\frac{1}{2\lambda} e^{-|\xi|/\lambda}}_{g(\xi|\lambda)}.$$

- The maximum value of $w(\xi)$ is $c = \exp\{-(\mu - \sigma^2/2)\}$, attained when $\xi = z - \exp\{\mu - \sigma^2\}$.
- The set $A_u = \{\xi \in \Omega : w(\xi) > uc\}$ is an interval with endpoints

$$\{\xi_1(u), \xi_2(u)\} = z - \exp\left\{(\mu - \sigma^2) \pm \left[\sigma^4 - 2\mu\sigma^2 + 2\sigma^2 \log(cu)\right]^{1/2}\right\}.$$

- CDF and quantile functions of g are respectively

$$G(\xi \mid \lambda) = \frac{1}{2} + \frac{1}{2}\,\mathrm{sgn}(\xi)[1 - e^{-|\xi|/\lambda}], \quad \text{and}$$
$$G^{-}(\varphi \mid \lambda) = -\lambda\,\mathrm{sgn}\left(\varphi - \frac{1}{2}\right) \log\left(1 - 2\left|\varphi - \frac{1}{2}\right|\right).$$

- Exact draws can be generated from g using standard software libraries.

Code Example

- Use `DirectSampling` package (Raim, 2021a) to draw from

$$f(\xi) \propto \underbrace{\frac{1}{z-\xi} \exp\left\{ -\frac{1}{2\sigma^2} [\log(z-\xi) - \mu]^2 \right\} \cdot \mathrm{I}(z > \xi)}_{w(\xi \mid z, \mu, \sigma^2)} \underbrace{\frac{1}{2\lambda} e^{-|\xi|/\lambda}}_{g(\xi \mid \lambda)}.$$

  Defaults in the following are: `N = 100` and `method = "small_rects"`.

- Set up: the two `get` functions are shown in the next slides.

```
library(DirectSampling)

w = get_lognormal_weight(z = 100, mu = 5, sigma2 = 3^2)
g = get_laplace_base(lambda = 0.2)
```

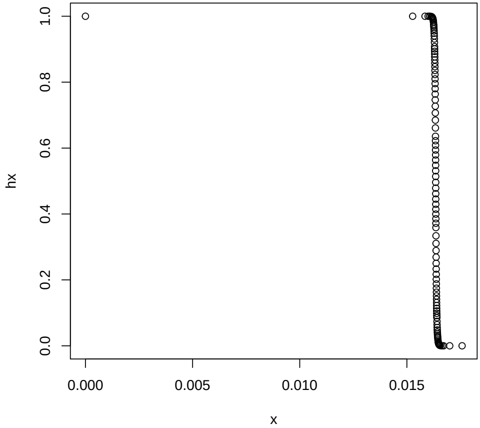
- Draw a sample.

```
R> direct_sampler(n = 20, w, g)
 [1] -0.11185683  0.03854990 -0.01441820 -0.22186401 -0.10575271 -0.06173714
 [7]  0.17074268 -0.06516397  0.11328199  0.11903198 -0.05490415  0.13957360
[13]  0.02124918  0.05218875  0.06050365 -0.07118187 -0.27217647 -0.27952874
[19]  0.13179049  0.75207048
```

Code Example

- Get the step function approximation.

```
step = Stepdown$new(w, g)
x = exp( step$get_log_x_vals() )
hx = exp( step$get_log_h_vals() )
plot(x, hx)
```



```
get_lognormal_weight = function(z, mu, sigma2)
{
    # The maximum value of the function log w(x)
    log_c = -(mu - sigma2 / 2)

    # Evaluate the weight function
    eval = function(x, log = FALSE) {
        n = length(x)
        out = rep(-Inf, n)
        idx = which(x < z)
        out[idx] = -log(z-x[idx]) - (log(z-x[idx]) - mu)^2 / (2*sigma2)
        if (log) { return(out) } else { return(exp(out)) }
    }

    # Return the roots of the equation w(x) = a in increasing order.
    roots = function(log_a) {
        x1 = z - exp((mu - sigma2) + sqrt(sigma2 * (sigma2 - 2*(mu + log_a))))
        x2 = z - exp((mu - sigma2) - sqrt(sigma2 * (sigma2 - 2*(mu + log_a))))
        c(x1, x2)
    }

    ret = list(log_c = log_c, roots = roots, eval = eval)
    class(ret) = "weight"
    return(ret)
}
```

```
get_laplace_base = function(lambda)
{
    density = function(x, log = FALSE) {
        d_laplace(x, 0, lambda, log)
    }

    # Compute Pr(x1 < X < x2) probability where X ~ Laplace(0, lambda)
    pr_interval = function(x1, x2) {
        p_laplace(x2, 0, lambda) - p_laplace(x1, 0, lambda)
    }

    # Quantile function of Laplace truncated to (x_min, x_max)
    q_truncated = function(p, x_min = -Inf, x_max = Inf) {
        p_min = p_laplace(x_min, 0, lambda)
        p_max = p_laplace(x_max, 0, lambda)
        x = q_laplace((p_max - p_min)*p + p_min, 0, lambda)
        max(x_min, min(x, x_max))
    }

    r_truncated = function(n, x_min = -Inf, x_max = Inf) {
        u = runif(n)
        x = numeric(n)
        for (i in 1:n) {
            x[i] = q_truncated(u[i], x_min, x_max)
        }
        return(x)
    }

    ret = list(pr_interval = pr_interval, q_truncated = q_truncated,
        r_truncated = r_truncated, density = density)
    class(ret) = "base"
    return(ret)
}
```

Regression Model Application

- We can now formulate a Gibbs sampler for a regression model with agency noise.
- The following scenario uses a Double Geometric noise mechanism for the outcome and a Laplace mechanism for the first covariate x_{i1}. The second covariate x_{i2} is observed without noise:

$$\begin{aligned} \tilde{y}_i &= y_i + \xi_i^y, \quad \xi_i^y \stackrel{\text{ind}}{\sim} \text{DGeom}(\rho_i^y), \\ \tilde{x}_{i1} &= x_{i1} + \xi_i^x, \quad \xi_i^x \stackrel{\text{ind}}{\sim} \text{Lap}(0, \lambda_i^x), \\ \log y_i &= x_{i1}\beta_1 + x_{i2}\beta_2 + \gamma_i, \quad \gamma_i \stackrel{\text{iid}}{\sim} \text{N}(0, \sigma^2), \end{aligned}$$

  for $i = 1, \ldots, n$.
- To complete the model specification, take the prior to be

$$\boldsymbol{\beta} \sim \text{N}_2(\mathbf{0}, \sigma_\beta^2 \boldsymbol{I}_2), \quad \sigma^2 \sim \text{IG}(a_\sigma, b_\sigma)$$

  with $\sigma_\beta = 10$, $a_\sigma = 2$, and $b_\sigma = 10$.

Regression Model Application

- To derive a Gibbs sampler, consider the joint distribution of all random variables, factorized as

$$f(\tilde{\boldsymbol{y}}, \boldsymbol{y}, \tilde{\boldsymbol{X}}, \boldsymbol{\xi}^x, \boldsymbol{\theta}) = f(\tilde{\boldsymbol{y}} \mid \boldsymbol{y}) \cdot f(\boldsymbol{y} \mid \tilde{\boldsymbol{X}}, \boldsymbol{\xi}^x, \boldsymbol{\theta}) \cdot f(\boldsymbol{\xi}^x) \cdot f(\boldsymbol{\theta})$$

with

$$f(\tilde{\boldsymbol{y}} \mid \boldsymbol{y}) = \prod_{i=1}^{n} f_{\text{DGeom}}(\tilde{y}_i - y_i \mid \rho_i^y),$$

$$f(\boldsymbol{\xi}^x) = \prod_{i=1}^{n} f_{\text{Lap}}(\xi_i^x \mid 0, \lambda_i^x),$$

$$f(\boldsymbol{y} \mid \tilde{\boldsymbol{X}}, \boldsymbol{\xi}^x, \boldsymbol{\theta}) = \prod_{i=1}^{n} f_{\text{LN}}(y_i \mid \boldsymbol{x}_{i\cdot}^\top \beta, \sigma^2),$$

$$f(\boldsymbol{\theta}) = f_{\text{N}}(\beta \mid \boldsymbol{0}, \sigma_\beta^2 \boldsymbol{I}_2) \cdot f_{\text{IG}}(\sigma^2 \mid a_\sigma, b_\sigma).$$

Regression Model Application

- We routinely obtain the conditionals:
  1. $[\beta \mid —] \sim \text{N}_2(\boldsymbol{\vartheta}, \boldsymbol{\Omega}^{-1})$ with $\boldsymbol{\Omega} = \sigma^{-2}\boldsymbol{X}^\top \boldsymbol{X} + \sigma_\beta^{-2}\boldsymbol{I}_2$ and $\boldsymbol{\vartheta} = \boldsymbol{\Omega}^{-1}\left(\sigma^{-2}\sum_{i=1}^n \boldsymbol{x}_{i\cdot} \log y_i\right)$,
  2. $[\sigma^2 \mid —] \sim \text{IG}(a^*, b^*)$ with $a^* = a_\sigma + n/2$ and $b^* = b_\sigma + \frac{1}{2}\sum_{i=1}^n (\log y_i - \boldsymbol{x}_{i\cdot}^\top \beta)^2$.
- For the unobserved outcomes,

$$f(\boldsymbol{y} \mid —) \propto \prod_{i=1}^n f_{\text{LN}}(y_i \mid \boldsymbol{x}_{i\cdot}^\top \boldsymbol{\beta}, \sigma^2) \cdot f_{\text{DGeom}}(\tilde{y}_i - y_i \mid \rho_i^y)$$

$$\propto \prod_{i=1}^n \underbrace{\frac{1}{y_i} \exp\left\{-\frac{1}{2\sigma^2}\left[\log y_i - \boldsymbol{x}_{i\cdot}^\top \beta\right]^2\right\} \text{I}(y_i \geq 0)}_{w(y_i \mid \boldsymbol{x}_{i\cdot}^\top \boldsymbol{\beta}, \sigma^2)} \cdot \underbrace{\frac{\rho_i^y}{2 - \rho_i^y}(1 - \rho_i^y)^{|\tilde{y}_i - y_i|}}_{g(\tilde{y}_i - y_i \mid \rho_i^y)},$$

so that each y_i can be drawn independently within the Gibbs sampler via the direct sampler.

Regression Model Application

- To sample noise $\boldsymbol{\xi}^x$ for covariate $\boldsymbol{x}_{\cdot 1}$,

$$f(\boldsymbol{\xi}^x \mid \text{—}) \propto \prod_{i=1}^{n} f_{\text{LN}}(y_i \mid \boldsymbol{x}_{i\cdot}^\top \boldsymbol{\beta}, \sigma^2) \prod_{i=1}^{n} f_{\text{Lap}}(\xi_i^x \mid 0, \lambda_i^x)$$

$$\propto \prod_{i=1}^{n} \underbrace{\exp\left\{ -\frac{1}{2\tau^2} [(\tilde{x}_{i1} - \xi_i^x) - \vartheta_{i1}]^2 \right\}}_{w(\xi_i^x \mid \tilde{x}_{i1}, \vartheta_{i1}, \tau^2)} \underbrace{\frac{1}{2\lambda_i^x} \exp\left\{ -\frac{1}{\lambda_i^x} |\xi_i^x| \right\}}_{g(\xi_i^x \mid \lambda_i^x)},$$

  where $\tau^2 = \sigma^2 / \beta_1^2$ and $\vartheta_{i1} = \beta_1^{-1} (\log y_i - x_{i2}\beta_2)$.

- Now $\xi_1^x, \ldots, \xi_n^x$ may be drawn independently within this step of the Gibbs sampler via the direct sampler.

- **Note:** use of a transformed $\boldsymbol{x}$ in the regression will change the conditional distribution!

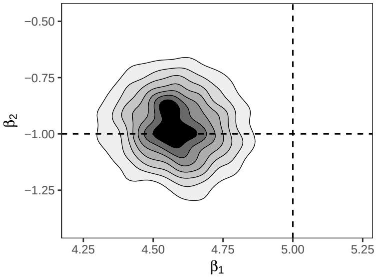
Simulation

- Using the Gibbs sampler, we can compare inference based on the noisy releases versus using the sensitive data:
    1. Algorithm 2: the full sampler we just derived.
    2. Algorithm 4: the sampler with $\boldsymbol{y}$ and $\boldsymbol{x}_{\cdot 1}$ observed.
- Settings: $n = 200$, $\boldsymbol{\beta} = (5, -1)$, $\sigma = 1$, with $\rho_i^y \equiv \rho \in \{0.01, 0.1, 0.4\}$ and $\lambda_i^x \equiv \lambda \in \{0.05, 0.10, 0.20\}$.
- Covariates $x_{ij} \sim \text{N}(0, 1)$ are generated independently for $j = 1, 2$ and $i = 1, \ldots, n$.
- Take the Lognormal regression model to be the (known) data-generating model, up to the parameter values.
- Algorithms 2 and 4 are used to produce a chain of 2,000 draws of $\boldsymbol{\theta}$, discarding the first 1,000 draws as burn-in and saving the remaining $R = 1{,}000$ draws.

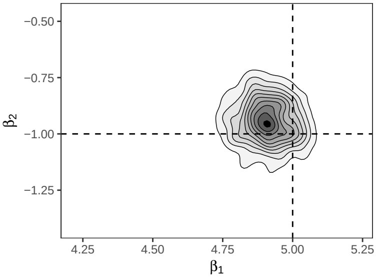
Simulation

- The simulation is repeated $S = 500$ times to produce realizations $\tilde{\boldsymbol{y}}^{(s)}$ and $\tilde{\boldsymbol{X}}^{(s)}$, and MCMC draws $\boldsymbol{\theta}^{(r,s)} = (\beta^{(r,s)}, \sigma^{2(r,s)})$ for $r = 1, \ldots, R$ and $s = 1, \ldots, S$ from each algorithm.
- Mean-squared error to summarize the posterior distribution of $\boldsymbol{\theta}$ relative to the true data-generating $\boldsymbol{\theta}_0$:

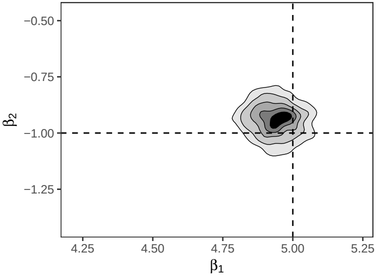
$$\text{MSE}^{(s)} = \frac{1}{R} \sum_{r=1}^{R} \|\boldsymbol{\theta}^{(r,s)} - \boldsymbol{\theta}_0\|^2 \approx \int \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2 f(\boldsymbol{\theta} \mid \tilde{\boldsymbol{y}}^{(s)}, \tilde{\boldsymbol{X}}^{(s)}) d\boldsymbol{\theta}.$$



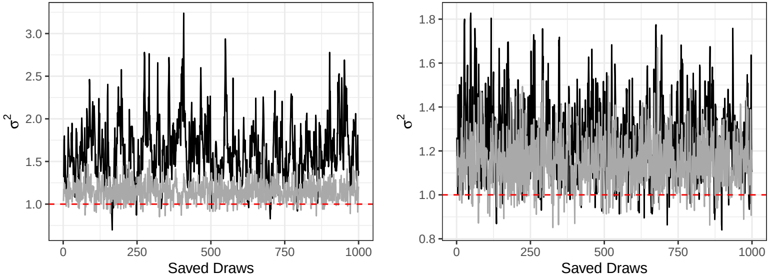
(a) $\rho = 0.01, \lambda = 0.2.$



(b) $\rho = 0.01, \lambda = 0.05.$



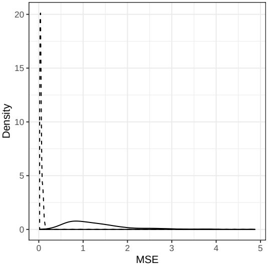
(c) Noise-free.



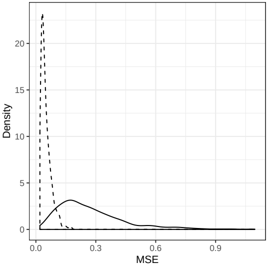
(a) $\lambda = 0.2$.

(b) $\lambda = 0.05$.

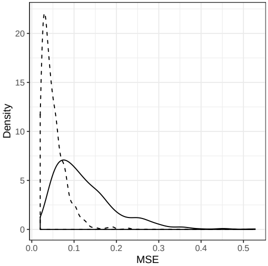
Figure: Traceplots of σ^2 draws for a particular data realization. The black and grey lines correspond to Algorithms 2 and 4, respectively, and red dashed lines mark true data-generating value $\sigma^2 = 1$.



(a) $\rho = 0.01, \lambda = 0.2$.

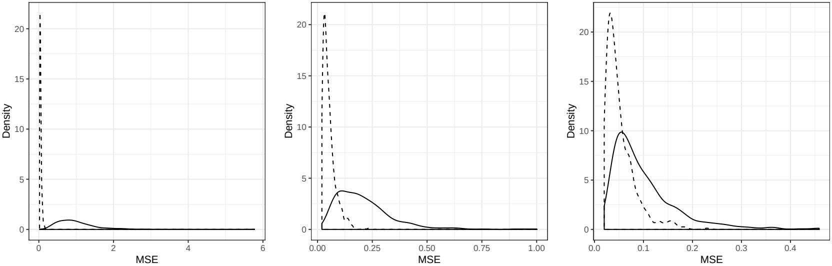


(b) $\rho = 0.01, \lambda = 0.1$.



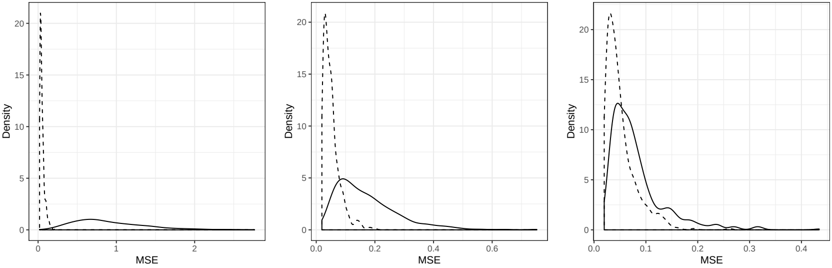
(c) $\rho = 0.01, \lambda = 0.05$.

Figure: The empirical density of $\text{MSE}^{(s)}$ over $S = 500$ simulation repetitions. Solid line and dashed lines represent Algorithms 2 and 4, respectively.



(a) $\rho = 0.1, \lambda = 0.2.$ (b) $\rho = 0.1, \lambda = 0.1.$ (c) $\rho = 0.1, \lambda = 0.05.$

Figure: The empirical density of $\text{MSE}^{(s)}$ over $S = 500$ simulation repetitions. Solid line and dashed lines represent Algorithms 2 and 4, respectively.



(a) $\rho = 0.4, \lambda = 0.2.$ (b) $\rho = 0.4, \lambda = 0.1.$ (c) $\rho = 0.4, \lambda = 0.05.$

Figure: The empirical density of $\text{MSE}^{(s)}$ over $S = 500$ simulation repetitions. Solid line and dashed lines represent Algorithms 2 and 4, respectively.

Conclusions

- We investigated some customizations to the direct sampler from Walker et al. (2011).
- This allowed us to implement a Gibbs sampler for Lognormal regression with additive DP noise for the outcome and/or covariates.
  - Avoids rejections.
  - Avoids manual tuning.
- Implementation is not trivial. E.g., care is required with floating point operations.
- Computations are somewhat heavy.
  - Timing for one run of the Gibbs sampler on our example.
  - Intel Core i7–2600 3.40 GHz workstation with four CPU cores.
  - 129.29 seconds total.
  - 70.29 seconds to draw $\boldsymbol{y}$'s.
  - 57.67 seconds to draw $\boldsymbol{\xi}^x$'s.
- Can we skip this and just use Stan (Carpenter et al., 2017)?
- Census 2020 data involves tabulations over geography, race, and other interesting relationships.

Thank You!

Andrew M. Raim
andrew.raim@census.gov

Andrew Raim. *Direct Sampling*, 2021a. R package version 0.1.0. https://github.com/andrewraim/DirectSampling.

Andrew M. Raim. Direct sampling in Bayesian regression models with additive disclosure avoidance noise. Research Report Series: Statistics #2021-01, Center for Statistical Research and Methodology, U.S. Census Bureau, 2021b. https://www.census.gov/library/working-papers/2021/adrm/RRS2021-01.html.

References I

John M. Abowd. The U.S. Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2867, New York, NY, USA, 2018. Association for Computing Machinery.

Garrett Bernstein and Daniel R Sheldon. Differentially private Bayesian linear regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 525–535. Curran Associates, Inc., 2019.

Claire McKay Bowen and Fang Liu. Comparative study of differentially private data synthesis methods. *Statistical Science*, 35(2):280–307, 2020.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete Gaussian for differential privacy, 2020. `https://arxiv.org/abs/2004.00010`.

Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76 (1):1–32, 2017.

Anne-Sophie Charest. How can we analyze differentially-private synthetic datasets? *Journal of Privacy and Confidentiality*, 2(2), 2011.

References II

Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers Inc, 2014.

Georgina Evans and Gary King. Statistically valid inferences from differentially private data releases, with application to the Facebook URLs dataset, 2020+. https://gking.harvard.edu/dpd.

Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. Issues encountered deploying differential privacy. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, WPES'18, pages 133–137, New York, NY, USA, 2018. Association for Computing Machinery.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6): 1673–1693, 2012.

Ruobin Gong. Exact inference with approximate computation for differentially private data via perturbations, 2019. https://arxiv.org/abs/1909.12237.

Ruobin Gong. Transparent privacy is principled privacy, 2020. https://arxiv.org/abs/2006.08522.

References III

Martin Klein and Bimal Sinha. Multiple imputation for parametric inference under a differentially private laplace mechanism. Technical Report Statistics #2019-05, Center for Statistical Research and Methodology, U.S. Census Bureau, 2019. `https://www.census.gov/library/working-papers/2019/adrm/RRS2019-05.html`.

Sidney I. Resnick. *A Probability Path*. Birkhäuser, 1999.

Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover, 1981.

Stephen G. Walker, Purushottam W. Laud, Daniel Zantedeschi, and Paul Damien. Direct sampling. *Journal of Computational and Graphical Statistics*, 20(3): 692–713, 2011.

Slice Sampler

For Continuous Univariate Target Distributions

- Suppose we wish to draw $X \sim \bar{f}(x)/C$ where $\bar{f}(x)$ is an unnormalized continuous density and $C = \int_{\mathbb{R}} \bar{f}(x)dx$.
- Consider the joint density

$$f(x, u) = \frac{1}{C} I(0 < u \leq \bar{f}(x)).$$

  We can verify that $\int f(x, u)du = f(x)$.

- From the joint density, we get conditionals

$$f(u \mid x) \propto I(0 < u \leq \bar{f}(x)),$$
$$f(x \mid u) \propto I(u \leq \bar{f}(x))$$

  Therefore $U \mid X \sim \text{Uniform}(0, \bar{f}(x))$ and $X \mid U$ follows a uniform dist'n on the set $\mathcal{S}_u = \{x \in \mathbb{R} : \bar{f}(x) \geq u\}$.

- A slice sampler is a Gibbs sampler which iterates between these steps.
- C does not need to be computed. The difficulty is usually to obtain $\mathcal{S}_u$.

Slice Sampler

For Discrete Univariate Target Distributions

- Now suppose we wish to draw $X \sim \bar{f}(x)/C$ where $\bar{f}(x)$ is an unnormalized discrete density and $C = \sum_{x \in \mathbb{Z}} \bar{f}(x) dx$.
- Again, start with the joint density

$$f(x, u) = \frac{1}{C} I(0 < u \leq \bar{f}(x)).$$

- From the joint density, we get conditionals

$$\begin{aligned} f(u \mid x) &\propto I(0 < u \leq \bar{f}(x)), \\ f(x \mid u) &\propto I(u \leq \bar{f}(x)) \end{aligned}$$

  As before, $U \mid X \sim \text{Uniform}(0, \bar{f}(x))$.
- Now $X \mid U$ follows a discrete uniform distribution on the set $\mathcal{S}_u = \{x \in \mathbb{Z} : \bar{f}(x) \geq u\}$.