dapper: Data Augmentation for Private Posterior Estimation in R

by Kevin Eng, Jordan A. Awan, Nianqiao Phyllis Ju, Vinayak A. Rao, and Ruobin Gong

Abstract This paper serves as a reference and introduction on using the dapper R package. It is an MCMC sampling framework which targets the exact posterior distribution given privitized data. The goal of this package is to provide researchers a tool to perform valid Bayesian inference on data protected by differential privacy. A strength of this framework is that it can be applied in situations where the likelihood is analytically intractable.

1 Introduction

Differential privacy (DP) provides a rigorous framework for protecting confidential information from re-identification attacks by using random noise to obscure the connection between the individual and data (Dwork et al. 2006). Its development was spurred on by successful attacks on anonymized data sets containing sensitive personal information. Prior to differential privacy, anonymization schemes did not always have sound theoretical guarantees despite appearing adequate. Differential privacy marks a leap forward in the science of privacy by putting it on rigorous footing and away from past ad hoc and obscure notions of privacy. Several recent high profile implementations of differential privacy include Apple (Tang et al. 2017), Google (Erlingsson, Pihur, and Korolova 2014), Microsoft (Ding, Kulkarni, and Yekhanin 2017), and the U.S. Census Bureau (Abowd 2018).

Many data sets amenable to differential privacy contain valuable information that stake holders are still interested in learning about. However, the noise introduced by differential privacy changes the calculus of inference. As an example, we can implement differential privacy for tabular data by directly adding independent, random error to each cell; the amount and type of which is determined by DP theory. When we fit a regression model to the noise infused data, this will correspond to having measurement errors in the covariates. This, unfortunately, violates the assumptions of most statistical models. In the presence of such errors, standard estimators can exhibit significant bias and incorrect uncertainty quantification (Gong 2022; Karwa, Kifer, and Slavković 2015; Wang et al. 2018). These issues are a serious concern for researchers (Santos-Lozada, Howard, and Verdery 2020; Kenny et al. 2021; Winkler et al. 2021). Therefore, developing privacy-aware statistical workflows are necessary in order for science and privacy to coexist.

Unfortunately making the necessary adjustments poses formidable mathematical challenges (Williams and Mcsherry 2010), even for seemingly simple models like linear regression. The difficulty lies in the marginal likelihood that results from correctly accounting for the injected privacy noise. This function is often analytically intractable and as a result, it is difficult or impossible to apply traditional statistical methods to derive estimators. In particular, the marginal likelihood can involve a complex integral where it is not even possible to evaluate the likelihood at a point. Tackling the problem by approximating the likelihood can be computationally infeasible since the integral is usually high dimensional. Few tools are available to researchers to address these issues, and their absence is a serious barrier to the wider adoption of differential privacy.

The **dapper** package provides a set of tools for conducting privacy-aware Bayesian inference. It serves as a R interface for the data augmentation framework proposed by Ju et al. (2022), allowing existing Bayesian models to be extended to handle noise infused data. The package is designed to integrate well with existing Bayesian workflows; results can by analyzed using tools from the **rstan** ecosystem in a drop-in fashion. Additionally, construction of a privacy-aware sampler is simplified through the specification of four independent modules. The benefits are twofold: several privacy mechanisms — these can even be from different formal privacy frameworks — can be compared easily by only swapping out relevant modules. And privacy mechanisms that have non-smooth transformations resulting in aforementioned intractable likelihoods (see example 3 which involves clamping) can be incorporated with little work. As a result, **dapper** may prove particularly useful to those engaged in studying the privacy utility trade-off or dealing with a privacy mechanism that involves multiple transformations.

The rest of this article is organized as follows: Section 2 covers the necessary background to understand the mathematical notation and ideas used throughout the paper. Section 3 goes over the main algorithm without going into mathematical detail– for specifics see Ju et al. (2022). Section 4 provides an overview of the **dapper** package and discusses important implementation details. Section 5 contains three examples of how one might use the package to analyze the impact of adding noise for privacy. The first example goes over a typical odds ratio analysis for a 2×2 table, the second example highlights the modular nature of **dapper** and reanalyzes the first example under a different privacy scheme, and the third example covers a linear regression model. Finally, section 6 discusses an important practical implication of using a small privacy budget with **dapper**.

2 Background

Let $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$ represent a confidential database containing n records. We will assume the data is generated by some statistical model $f(x \mid \theta)$. In many studies, scientist are interested in learning about θ because it provides important information about the subject under investigation.

In the Bayesian statistical framework, learning about θ is accomplished by using the data x to update the posterior $p(\theta \mid x) \propto f(x \mid \theta)p(\theta)$. Here, $p(\theta)$ is called the prior distribution, and represents the researcher's belief about θ before seeing any data. The posterior represents uncertainty around θ and is formed by using Baye's rule to fuse together the observed data and the research's prior belief. One major advantage of the Bayesian method is that, through the prior, it provides a mechanism for incorporating information not explicitly contained in the data at hand. This is especially useful in settings where there is considerable domain knowledge on the value of θ.

For large data sets, it is common to work with a summary statistic $s = s(x)$ that has much smaller dimension than the original data because doing so can greatly simplify calculations. Since summary statistics are easier to work with, database curators often publish them to efficiently communicate information contained in large data sets. This makes them a natural target for dissemination based privacy approaches.

Differential Privacy

While a summary statistic can already partially anonymize data, it is still possible to deduce information about an individual depending on how x is distributed. Differential privacy offers a more principled approach by introducing randomness such that the output distribution does not change much when one individual's data is changed. A common approach – and the one dapper is primarily designed to address– is to take a summary statistic s, and add noise to it to produce a noisy summary statistic s_{dp}.

While adding noise into confidential data is already a well established practice in statistical disclosure control (Dalenius and Reiss 1982), differential privacy provides a rigorous framework to specify where and how much noise to add. Most importantly, for the analyst, the specification of the differentially private noise mechanism can be made available without compromising privacy and thus incorporated into subsequent analyses.

The **dapper** package provides a flexible framework that can accommodate the many different flavors of differential privacy; the main requirement being that the DP mechanism has a closed-form density. However, for presentation, in this section we focus on the earliest and most common formulation of differential privacy, ϵ-differential privacy (ϵ-DP). The ϵ parameter is called the privacy loss budget. This parameter controls how strong the privacy guarantee is. Larger values of ϵ correspond to weaker privacy guarantees which in turn means less noise being added.

We now describe the ϵ-DP privacy framework in more detail. For the noisy summary statistic, we write $s_{dp} \sim \eta(\cdot \mid x)$. Here, η is the density of the privacy mechanism designed to meet a certain property: The privacy mechanism η is said to be ϵ-differentially private (Dwork et al. 2006) if for all values of s_{dp}, and all "neighboring" databases $(x, x') \in \mathcal{X}^n \times \mathcal{X}^n$ differing by one record (specifically we consider $d(x, x') \leq 1$ where d is the Hamming distance), the probability ratio is bounded:

$$\frac{\eta(s_{dp} \mid x)}{\eta(s_{dp} \mid x')} \leq \exp(\epsilon), \quad \epsilon > 0.$$

The differential privacy framework is used to create and verify privacy mechanisms. One such mechanism is the *Laplace mechanism*. It works by taking a deterministic statistic $s : \mathcal{X} \mapsto \mathbb{R}^m$ and constructs the privatized statistic $s_{dp} := s(x) + u$ where u is a m-dimensional vector of i.i.d. Laplace random variables. The amount of noise, u, is scaled proportionally to the *global sensitivity* (or just sensitivity) of the statistic s. We define the sensitivity of a statistic s as $\Delta(s) := \max_{(x,x') \in \mathcal{X}^n \times \mathcal{X}^n; d(x,x') \leq 1} \|s(x) - s(x')\|$. If we draw each $u_i \sim \text{Lap}(\Delta(s)/\epsilon)$, we can show s_{dp} is ϵ-differentially private for the the Laplace mechanism. Example 3, will cover an application of the Laplace mechanism to linear regression. Other common noise adding mechanisms include the Gaussian and the discrete Gaussian mechanisms, which also add noise scaled to the sensitivity; however these mechanisms satisfy a different privacy criteria called *zero concentrated differential privacy* (zCDP)

(Bun and Steinke 2016). [1]

3 Methodology

Given privatized data, s_{dp}, the goal of Bayesian inference is to sample from the private posterior distribution $p(\theta \mid s_{dp})$. Since the observed likelihood, $p(s_{dp} \mid \theta)$, often has no simple closed form expression (Williams and Mcsherry 2010), most standard approaches do not apply. To conduct privacy-aware Bayesian inference, the **dapper** package implements the data augmentation algorithm introduced in Ju et al. (2022) which allows us to sample from $p(\theta \mid s_{dp})$ without knowing a closed-form expression proportional to $p(s_{dp} \mid \theta)$. The algorithm accomplishes this by considering the joint distribution $p(\theta, x \mid s_{dp})$ and alternates sampling between the two distributions $p(\theta \mid x, s_{dp})$ and $p(x \mid \theta, s_{dp})$.

Since s_{dp} is derived from x, we have $p(\theta \mid x, s_{dp}) = p(\theta \mid x)$ which is just the usual posterior distribution given the confidential data x. The **dapper** package assumes the user has access to a sampler for $p(\theta \mid x)$. This can come from any R package such as **fmcmc** or constructed analytically via posterior conjugacy. For the second distribution, $p(x \mid \theta, s_{dp})$ may only be known up to a constant. The **dapper** package samples from this distribution by running a Gibbs-like sampler: Similar to a Gibbs sampler, each of the n components of x is individually updated. However unlike a Gibbs sampler, each component is updated using a Metropolis-Hastings algorithm. This method is sometimes called the Metropolis-within-Gibbs sampler (Robert and Casella 2004).

In some cases, sampling from $p(x \mid \theta, s_{dp})$ can be made more efficient when the privacy mechanism can be written as a function of s_{dp} and a sum consisting of contributions from each individual record. More precisely, we say the privacy mechanism satisfies the *record additivity* property if

$$\eta(s_{dp} \mid x) = g\left(s_{dp}, \sum_{i=1}^{n} t_i(x_i, s_{dp})\right)$$

for some known and tractable functions $g, t_1, \ldots, t_n$. The sample mean is a example of a summary statistic satisfying record additivity where $t_i(x_i, s_{dp}) = x_i$.

The data augmentation algorithm is described in the following pseudo code:

1. Sample θ^{t+1} from $p(\cdot \mid x^{(t)})$.
2. Sample from $p(x \mid \theta, s_{dp})$ using a three step process
    - Propose $x_i^* \sim f(\cdot \mid \theta)$.
    - If s satisfies the record additive property then update $s(x^*, s_{dp}) = t(x, s_{dp}) - t_i(x_i, s_{dp}) + t_i(x_i^*, s_{dp})$.
    - Accept the proposed state with probability $\alpha(x_i^* \mid x_i, x_{-i}, \theta)$ given by:

$$\alpha(x_i^* \mid x_i, x_{-i}, \theta) = \min\left\{1, \frac{\eta(s_{dp} \mid s(x_i^*, x_{-i}))}{\eta(s_{dp} \mid s(x_i, x_{-i}))}\right\} = \min\left\{1, \frac{g(s_{dp}, t(x^*, s_{dp}))}{g(s_{dp}, t(x, s_{dp}))}\right\}.$$

Theoretical results such as bounds on the acceptance probability as well as results on ergodicity can be found in Ju et al. (2022).

4 The Structure of dapper

The **dapper** package is structured around the two functions `dapper_sample()` and `new_privacy()`. The function, `dapper_sample()`, is used to generate MCMC draws from the private posterior. Targeting the correct private posterior requires a large set of inputs. In order to simplify the process of setting up the sampler, the `dapper_sample()` function uses a `privacy` object to encapsulate all information about the data generating process. The role of `new_privacy()` is to construct `privacy` objects. This separates inputs describing the data generating process from inputs describing simulation parameters, which decreases the chance for input related bugs.

[1] Example 2 will consider another privacy framework called (ϵ, δ)-differential privacy which is an extension of ϵ-differential privacy to the case where the ratio bound can fail with probability governed by δ. More specifically, we say a privacy mechanism, $\mathcal{M}$, satisfies (ϵ, δ)-differential privacy if for all neighboring databases where $d(x, x') = 1$, we have

$$P(\mathcal{M}(\S) \in S) \leq \epsilon P(\mathcal{M}(\S') \in S) + \delta$$

for any $S \subseteq \text{Range}(\mathcal{M})$ and $\delta \in [0, 1]$. Note setting $\delta = 0$ gives us back the pure ϵ-differential privacy condition.

Utility functions facilitating work with count data are also included. These center around the mass function and random number generators of the discrete Gaussian and discrete Laplacian distributions and are described in more detail in the Privacy Mechanisms for Count Data section.

Privacy Model

Creating a privacy model is done using the `new_privacy()` constructor. The main arguments consist of the four components as outlined in the methodology section.

```
new_privacy(post_f = NULL, latent_f = NULL, priv_f = NULL,
            st_f = NULL, npar = NULL)
```

To minimize the potential for bugs, there are a set of requirements the four main components must adhere to which are described below:

- `latent_f()` is a function that samples from the parametric model describing how to generate a new confidential database x given model parameters θ. Its syntax must be `latent_f(theta)` where `theta` is a numeric vector representing the model parameters. This function must work with the `init_par` argument of `dapper_sample()`. The output must be a $n \times p$ numeric matrix where n is the number of observations and p is the dimension of a record x. The matrix requirement is strict so even if $p = 1$, `latent_f()` should return a $n \times 1$ matrix and not a vector of length n.
- `post_f()` is a function which makes a one-step draw from the private posterior. It has the syntax `post_f(dmat, theta)`. Here `dmat` is a numeric matrix representing the confidential database and `theta` is a numeric vector which serves as the initialization point for a one sample draw. The easiest, bug-free way to construct `post_f()` is to use a conjugate prior. However, this function can also be constructed by wrapping a MCMC sampler generated from other R packages (e.g. **rstan**, **fmcmc**, **adaptMCMC**). Using this approach requires caution; **dapper** requires a valid draw and many sampler implementations violate this requirement. This is especially true for adaptive samplers like **rstan**'s HMC where the first few draws are used to initialize the gradient and do not necessarily correspond to draws from a valid MCMC chain. Additionally, some packages like **mcmc** will generate samplers that may be slow due to a large initialization overhead. For these reasons we recommend sticking with conjugate priors as they will be quick and avoid serious undetected semantic errors arising from specific implementation details of other R packages.
- `priv_f()` is a function that returns the log of the privacy mechanism density given the noise-infused summary statistics s_{dp} and its potential true value $s(x, s_{dp}) := \sum_{i=1}^{n} t_i(x_i, s_{dp})$. This function has the syntax `priv_f(sdp, sx)` where `sdp` and `sx` are numeric vectors or matrices representing the the value of s_{dp} and $s(x, s_{dp})$ respectively. The arguments must appear in the exact order with the same variables names as defined above. Finally, the return value of `priv_f()` must be a scalar value.
- `st_f()` is a function which calculates a summary statistic. It must be defined using the three arguments named `i`, `xi` and `sdp` in the stated order. The role of this function is to represent terms in the definition of record additivity with each of the three arguments in `st_f` corresponding the the similarly spelled terms in $t_i(x_i, s_{dp})$. Here `i` is an integer, while `xi` is a numeric vector and `sdp` is an numeric vector or matrix. The return value must be a numeric vector or matrix.
- `npar` is an integer equal to the dimension of θ.

Sampling

The `dapper_sample()` function essentially takes an existing Bayesian model and extends it to handle privatized data. The output of `dapper_sample()` contains MCMC draws from the private posterior. The function has syntax:

```
dapper_sample(data_model, sdp, init_par, niter = 2000, warmup = floor(niter / 2),
          chains = 1, varnames = NULL)
```

The parameters `data_model`, `sdp`, and `init_par` are required. The `data_model` input is a privacy model object that is constructed using `new_privacy()` (see section Privacy Model). The value of `sdp` is equal to the observed noise infused statistic. We require the object class of `sdp` to be the same as the output of `st_f()`. For example, if `st_f()` returns a matrix then `sdp` must also be a matrix. The provided starting value of the chain (`init_par`) must work with the `latent_f()` component. An error will be thrown if `latent_f()` evaluated at `init_par` does not return a numeric matrix.

The optional arguments are the number of MCMC draws (`niter`), the burn in period (`warmup`), number of chains (`chains`) and character vector that names the parameters. Running the chain without any warm up can be done by setting the value to 0. Running multiple chains can be done in parallel using the **furrr** package. Additionally, progress can be monitored using the **progressr** package. Adhering to the design philosophy of the two packages, we leave the setup to the user so that they may choose the most appropriate configuration for their system. The contingency table demonstration given in section 5 walks through a typical setup of **furrr** and **progressr**.

The return value of `dapper_sample()` is a list containing a `draw_matrix` object and a vector of acceptance probabilities of size `niter`. The `draw_matrix` object is described in more detail in the **posterior** package. The advantages with working with a `draw_matrix` object is that it is compatible with many of the packages in the **rstan** ecosystem. For example, any `draw_matrix` object can be plugged directly into the popular **bayesplot** package. Additionally, `dapper`'s basic summary function provides the same posterior summary statistics as those found when using **rstan**. Overall, this should make working with `dapper` easier for anyone already familiar with the **rstan** ecosystem.

Privacy Mechanisms for Count Data

The **dapper** package provides several utility functions for analyzing privatized count data. Currently, the U.S. Census Bureau is a major driver behind the deployment and research of privatized count data, and these functions were created for census oriented researchers in mind.

Pure ϵ-differential privacy is a strong requirement that can lead to poor data utility. For this reason the U.S. Census Bureau uses zero concentrated differential privacy which is a weaker criteria. The **dapper** package implements probability mass and sampling functions for the discrete Gaussian and discrete Laplace distributions (Canonne, Kamath, and Steinke 2021). These are two potential mechanisms for deploying concentrated differential privacy, and both are used in several current U.S. Census Bureau data products (Labs 2022).

Equations (1) and (2) in the panel below give the probability mass functions for the discrete Gaussian and discrete Laplace distributions respectively.

$$P[X = x] = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-(x-\mu)^2/2\sigma^2}} \tag{1}$$

$$P[X = x] = \frac{e^{1/t} - 1}{e^{1/t} + 1} e^{-|x|/t} \tag{2}$$

The support of both distributions is the set of all integers. The discrete Gaussian has two parameters $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}^+$ which govern the location and scale respectively. On the other hand, the discrete Laplace only has the scale parameter $t \in \mathbb{R}^+$. The functions `ddnorm` and `rdnorm` provide the density and sampling features for the discrete Gaussian distribution. The `ddnorm` function contains a calculation for the normalizing constant which is expensive. To speed up repeated execution, an optional logical argument named `normalize` is provided. When set to `FALSE`, it forgoes calculating the normalizing constant, however the default value is `TRUE`. Finally, the functions `ddlaplace` and `rdlaplace` provide similar features for the discrete Laplace distribution. The implementations of these functions are not the safest and are meant to provide users with a way to quickly explore the behavior of these DP mechanisms.

5 Examples

Example 1: 2x2 Contingency Table (Randomized Response)

As a demonstration, we analyze a subset of the UC Berkeley admissions data, which is often used as an illustrative example of Simpson's paradox (Bickel, Hammel, and O'Connell 1975). The question posed is whether the data suggest there is bias against females during the college admissions process. Table 1 shows the aggregate admissions result from six departments based on sex for a total of $N = 400$ applicants. The left sub-table shows the confidential data and the right shows the resulting counts after applying the random response privacy mechanism.

To see how the privacy mechanism works, we envision the record level data set as a $N \times 2$ matrix with the first column representing sex and the second column representing admission status. Thus each row in the matrix is the response of an individual. To anonymize the results, we apply a random

	Admitted	Rejected
Female	46	118
Male	109	127

	Admitted	Rejected
Female	74	102
Male	104	120

Table 1: The table on the left shows the confidential admissions data and the right show the perturbed data as a result of applying the response mechanism.

response scheme where for each answer we flip a fair coin twice.[2] As a concrete example, suppose Robert is a male who was rejected. To anonymize his response, we would first flip a coin to determine if his sex response is randomized. If the first flip is heads we keep his original response of being a male. If we see tails, then we would flip the coin again and change the answer to male or female depending on whether we see heads or tails respectively. We then repeat this process for his admission status. This anonymization scheme conforms to a mechanism with a privacy budget of at most $\epsilon = 2\log(3)$.

To set up dapper to analyze the anonymized admissions data, we first encode our anonymized record level data using a binary matrix where male and admit take the value 1. From this we can construct s_{dp} as the columns of the binary matrix stacked on top of each other.

1. `latent_f`: For each individual there are four possible sex/status responses which can be modeled using a multinomial distribution. To implement draws from the multinomial distribution we use the `sample` function to take samples from a list of containing the four possible binary vectors. Note the final line results in a 400×2 matrix.

```
latent_f <- function(theta) {
  tl <- list(c(1,1), c(1,0), c(0,1), c(0,0))
  rs <- sample(tl, 400, replace = TRUE, prob = theta)
  do.call(rbind, rs)
}
```

2. `post_f`: Given the confidential data, we can derive the posterior analytically using a Dirichlet prior. In this example, we use a flat prior which corresponds to Dirichlet(1) distribution. The code below generates samples from the Dirichlet distribution using random draws from the gamma distribution.

```
 post_f <- function(dmat, theta) {
  sex <- dmat[,1]
  status <- dmat[,2]

  #Male & Admit
  x1 <- sum(sex & status)
  x2 <- sum(sex & !status)
  x3 <- sum(!sex & status)
  x4 <- sum(!sex & !status)

  x <- c(x1, x2, x3, x4)

  t1 <- rgamma(4, x + 1, 1)
  t1/sum(t1)
}
```

3. `st_f`: The private summary statistic s_{dp} can be written as a record additive statistic using indicator functions. Let v_i be a binary vector of length $800 = 2 \times 400$ where the entries with index i and $400 + i$ are the only possible non zero entries. We let these two entries correspond to the sex and admission status response of the individual with record x_i. With this construction we have $s_{dp} = \sum_{i=1}^{400} v_i$.

```
st_f <- function(xi, sdp, i) {
  x <- matrix(0, nrow = 400, ncol = 2)
  x[i,] <- xi
  x
}
```

[2] The randomized response scheme predates the development of differential privacy and was first described by Warner (1965) as a means to reduce survey bias involving sensitive questions.

4. priv_f: The privacy mechanism is the result of two fair coin flips, so for each answer there is a 3/4 chance it remains the same and a 1/4 chance it changes. Hence the log likelihood of observing s_{dp} given the current value of the latent database, sx, is $\log(3/4)$ times the number of entries that match plus $\log(1/4)$ times the number of entries which differ.

```
priv_f <- function(sdp, sx) {
  t1 <- sum(sdp == sx)
  t1 * log(3/4) + (800 - t1) * log(1/4)
}
```

Below we load the data and create the noisy admissions table.

```
#Original UCBAdmissions data.
cnf_df <- tibble(sex = c(1, 1, 0, 0),
                 status = c(1, 0, 1, 0),
                 n = c(1198, 1493, 557, 1278)) %>% uncount(n)

set.seed(1)
ix <- sample(1:nrow(cnf_df), 400, replace = FALSE)
cnf_df <- cnf_df[ix,]

#Answers to be randomized
ri <- as.logical(rbinom(800, 1, 1/2))

#Randomized answers
ra <- rbinom(sum(ri), 1, 1/2)

#Create sdp
sdp <- as.matrix(cnf_df)
sdp[ri] <- ra
```

Once we have defined all components of the model we can create a new privacy model object using the new_privacy function and feed this into the dapper_sample function. Below we run four chains in parallel each with 5,000 posterior draws with a burn-in of 1000.

```
library(dapper)
library(furrr)
plan(multisession, workers = 4)

dmod <- new_privacy(post_f   = post_f,
                    latent_f = latent_f,
                    priv_f   = priv_f,
                    st_f     = st_f,
                    npar     = 4,
                    varnames = c("pi_11", "pi_10", "pi_01", "pi_00"))

dp_out <- dapper_sample(dmod,
                  sdp = sdp,
                  seed = 123,
                  niter = 6000,
                  warmup = 1000,
                  chains = 4,
                  init_par = rep(.25,4))
```

If the run time of dapper_sample is exceptionally long, one can use the progressr package to monitor progress. The progressor framework allows for a unified handling of progress bars in both the sequential and parallel computing case.

```
library(progressr)
handlers(global = TRUE)

handlers("cli")
dp_out <- dapper_sample(dmod,
                  sdp = sdp,
```

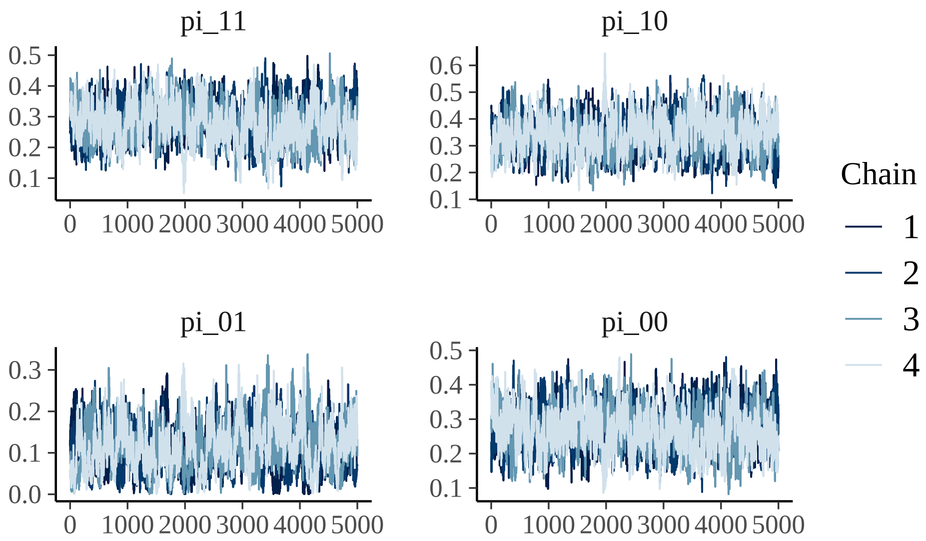

Figure 1: (Example 1) trace plots.

```
seed = 123,
niter = 6000,
warmup = 1000,
chains = 4,
init_par = rep(.25,4))
```

Results can be quickly summarized using the `summary` function which is displayed below. The `rhat` values in the table are close to 1, which indicates the chain has run long enough to achieve adequate mixing.

```
#> # A tibble: 4 x 10
#>   variable  mean median     sd    mad     q5   q95  rhat ess_bulk ess_tail
#>   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>    <dbl>    <dbl>
#> 1 pi_11    0.281  0.281 0.0610 0.0625 0.182  0.382  1.02     362.     818.
#> 2 pi_10    0.336  0.335 0.0638 0.0640 0.235  0.444  1.01     431.    1191.
#> 3 pi_01    0.111  0.108 0.0548 0.0563 0.0250 0.206  1.02     282.     504.
#> 4 pi_00    0.272  0.272 0.0601 0.0616 0.172  0.372  1.02     389.     875.
```

Diagnostic checks using trace plots can be done using the **Bayesplot** package as shown in figure 1. It is especially important to check for good mixing with **dapper** since sticky chains are likely to be produced when the amount of injected noise is high. See Discussion on Mixing and Privacy Loss Budget for a more detailed explanation.

To see if there is evidence of gender bias we can look at the odds ratio. Specifically, we look at the odds of a male being admitted to that of female. A higher odds ratio would indicate a bias favoring males. Figure 2 shows draws from the private posterior. The large odds ratio values would seem to indicate there is bias favoring the males. The “paradox” arises when analyzing the data stratified by university department where the odds ratio flips with females being favored over males.

For comparison, we run a standard Bayesian analysis on the noise infused table ignoring the privacy mechanism. This will correspond exactly to the model defined in the `post_f` component. Figure 3 shows a density estimate for the odds ratio under the confidential and noisy data. The posterior distribution for the odds ratio under the noisy data is shifted significantly, indicating a large degree of bias. Looking at left hand plot in figure 3 shows the MAP estimate from **dapper** is similar to that in the case of the confidential data. The width of the posterior is also much larger since it properly accounts for the uncertainty due to the privacy mechanism. This illustrates the dangers of ignoring the privacy mechanism: a naive analysis not only has bias, but also severely underestimates the uncertainty associated with the odds ratio estimate.

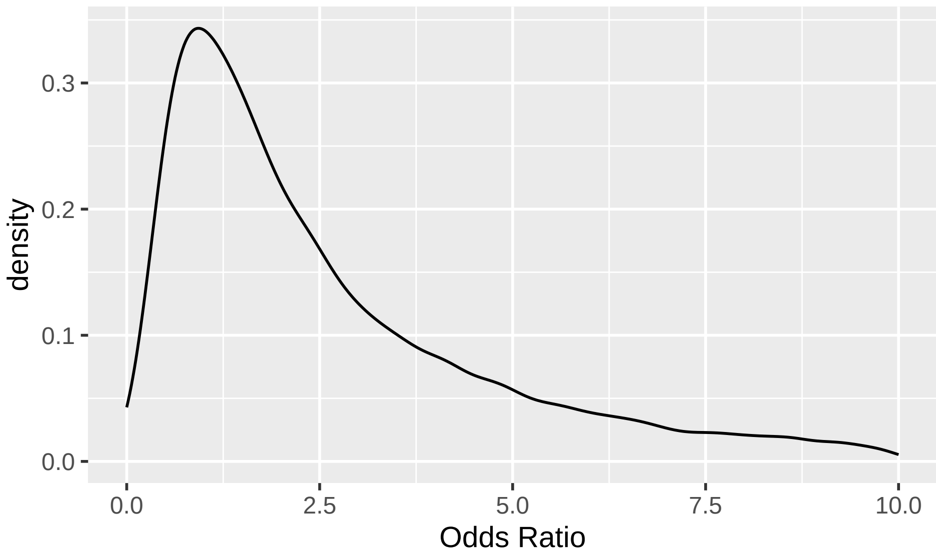

Figure 2: (Example 1) posterior density estimate for the odds ratio using 5,000 MCMC draws.

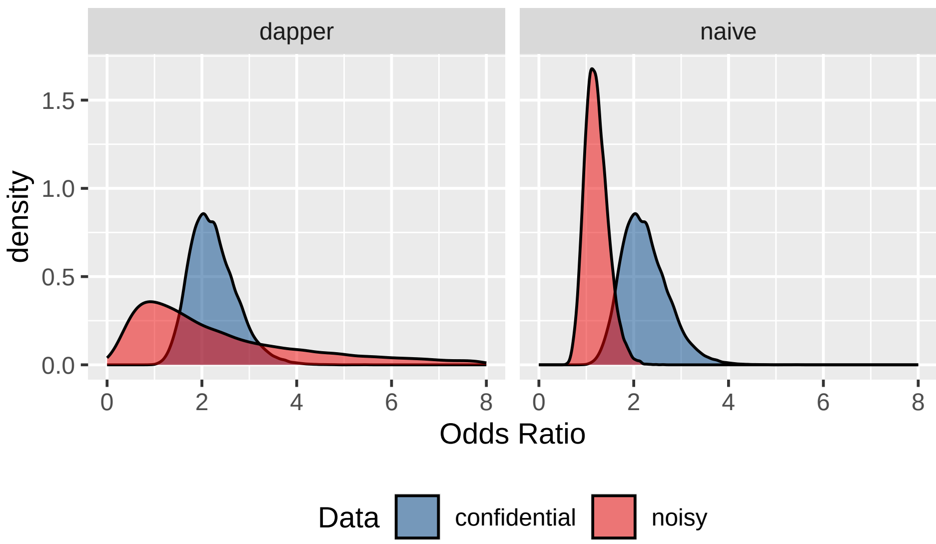

Figure 3: (Example 1) Posterior distributions (red) of the odds ratio (admission of males vs. females) using noisy (i.e. privacy-protected) data. Left panel: correct Bayesian inference using dapper which takes into account the privacy mechanism; Right panel: naive Bayesian inference treating the noisy data as noise-free. Blue distribution in both panels reflect the true posterior distribution if the analysis were to be conducted on the confidential data.

Example 2: 2x2 Contingency Table (Discrete Gaussian)

To highlight the flexibility of **dapper** we reanalyze example 1 with a different privacy mechanism. Here we take inspiration from the 2020 U.S. Decennial Census, which deployed a novel privacy protection system for count data. One of the privacy mechanisms used is the discrete Gaussian distribution. In our admissions data example, this privacy mechanism works by injecting noise into the total cell counts given in the 2x2 table. The randomized response scheme, in contrast, injects noise at the record level. The **dapper** package accommodates both the randomized response and the discrete Gaussian mechanisms, allowing us to compare the impact of the two approaches.

To begin comparing the two approaches, we need to set the privacy parameters. Since the two frameworks use different metrics, there does not exist a direct comparison. However, there is a direct relationship between zCDP and (ϵ, δ)-differential privacy. The latter framework is a relaxed version of ϵ-DP where the ratio bound only holds in probability. For our comparison, we will set $\delta = 10^{-10}$ which is the value the U.S. Census Bureau uses. For this value of δ, setting the scale parameter in the discrete Guassian distribution to $\sigma = 6.32$ will guarantee $(2\log(3), 10^{-10})$-differential privacy.[3]

	Admitted	Rejected
Female	46	118
Male	109	127

	Admitted	Rejected
Female	47	110
Male	110	131

Table 2: The table on the left shows the confidential admissions data and the right show the perturbed data as a result of applying the discrete Gaussian mechanism with $\sigma = 6.32$.

For the public, the U.S. Census Bureau only releases aggregate cell counts along with the true total count of said cells. In our example, this means the Census would have released the right hand table along with the fact that $N = 400$ in the original table. Thus, it natural to let s_{dp} be the vector of cell counts. As in example 1, we imagine the latent database as a 400×2 binary matrix. Below we describe the process for analyzing the privatized data using **dapper**. Since the latent process and posterior are the same as example 1, we only describe how to construct `st_f` and `priv_f`.

1. `st_f`: The private summary statistic s_{dp} can be written as a record additive statistic using the indicator vectors $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,0,1)$ and $(0,0,0,1)$. These four vectors correspond to the four possible cells.

```
st_f <- function(xi, sdp, i) {
  if(xi[1] & xi[2]) {
    c(1,0,0,0)
  } else if (xi[1] & !xi[2]) {
    c(0,1,0,0)
  } else if (!xi[1] & xi[2]) {
    c(0,0,1,0)
  } else {
    c(0,0,0,1)
  }
}
```

2. `priv_f`: The privacy mechanism us a discrete Gaussian distribution centered at 0.

```
priv_f <- function(sdp, sx) {
  sum(dapper::ddnorm(sdp - sx, mu = 0, sigma = 6.32, log = TRUE))
}
```

The summary table below show the results of running a chain for 2000 iterations with a burn-in of 1000 runs.

```
summary(dp_out)
```

```
#> # A tibble: 4 x 10
#>   variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
#>   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
```

[3] $\frac{1}{2}\epsilon^2$-concentrated differential privacy implies $\left(\frac{1}{2}\epsilon^2 + \epsilon\sqrt{2\log(1/\delta)}, \delta\right)$-differential privacy (Bun and Steinke 2016)

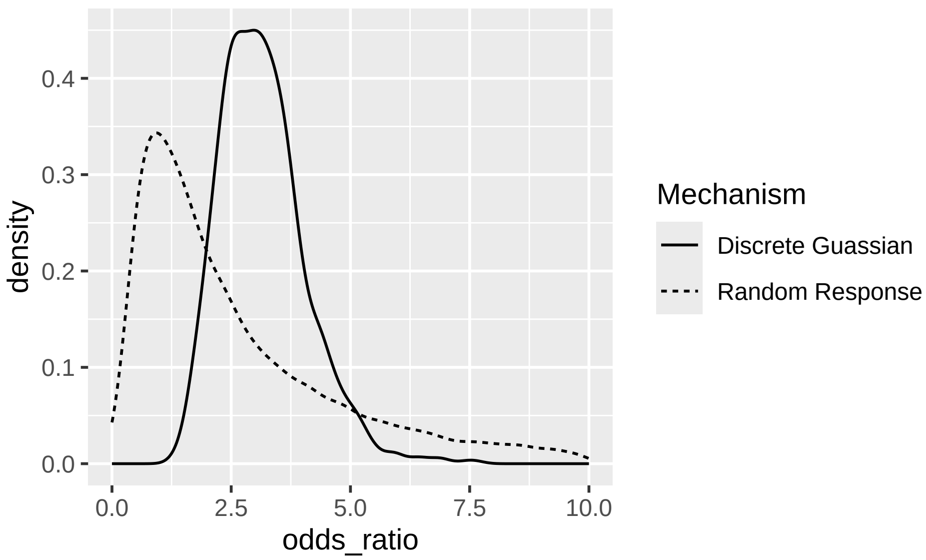

Figure 4: (Example 2) Private posterior density estimate for the odds ratio under random response (dashed) and discrete Gaussian (solid). Density plots are made using 5,000 and 1,000 MCMC draws for the random response and discrete Gaussian respectively.

```
#> 1 pi_11    0.315  0.313 0.0267 0.0271 0.273  0.360  1.00     658.     594.
#> 2 pi_01    0.286  0.285 0.0268 0.0270 0.243  0.330  1.00     611.     724.
#> 3 pi_10    0.107  0.106 0.0198 0.0197 0.0767 0.141  1.00     422.     768.
#> 4 pi_00    0.292  0.292 0.0267 0.0265 0.250  0.338  1.00     650.     813.
```

Figure 4 juxtaposes the private posterior under the randomized (dashed line) and discrete Gaussian (solid line) mechanism. Comparing the two suggest using discrete Guassian noise leads to slightly less posterior uncertainty and a mode closer to the true, confidential posterior.

Additionally, if we compare the left hand plots of figure 3 and figure 5, the discrete Gaussian induces considerably less bias when using the naive analysis.

Example 3: Linear Regression

In this section we apply **dapper** to reconstruct an example presented in Ju et al. (2022). In it, they apply a Laplace privacy mechanism to a sufficient summary statistic for a linear regression model. Let $\{(x_i, y_i)\}_{i=1}^n$ be the original, confidential data with $x_i \in \mathbb{R}^2$. They assume the true data generating process follows the model

$$\begin{aligned} y &= -1.79 - 2.89x_1 - 0.66x_2 + \epsilon \\ \epsilon &\sim N(0, 2^2) \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &\sim N_2(\mu, I_2) \\ \mu &= \begin{pmatrix} 0.9 \\ -1.17 \end{pmatrix}. \end{aligned}$$

Note, in most settings involving linear regression, the covariates are assumed to be fixed, known constants. Thus the formulation above is a departure from the norm since we are assuming a random design matrix. More details on why this framing is necessary will be provided later when describing the latent model. The paper considers the scenario where one desires to publicly release the sufficient summary statistics

$$s(x, y) = (x^T y, y^T y, x^T x).$$

This summary statistic satisfies the additive record property since $s(x, y) = \sum_{i=1}^n t(x_i, y_i)$ where

$$t(x_i, y_i) = ((x_i)^T y_i, y_i^2, (x_i)^T x_i).$$

To make the statistic compliant with the ϵ-DP criterion it is necessary to bound the value of the statistic

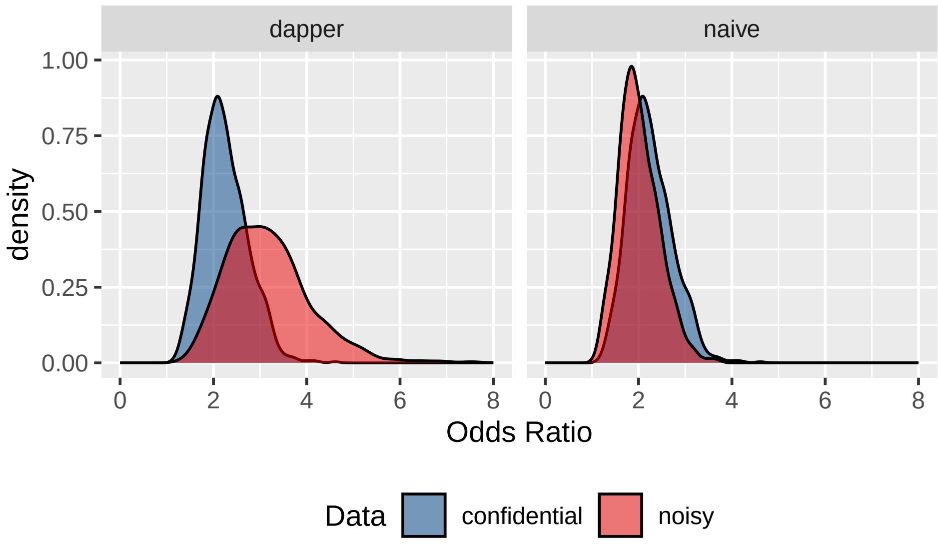

Figure 5: (Example 2) Posterior distributions (red) of the odds ratio (admission of males vs. females) using noisy (i.e. privacy-protected) data. Left panel: correct Bayesian inference using dapper which takes into account the privacy mechanism; Right panel: naive Bayesian inference treating the noisy data as noise-free. Blue distribution in both panels reflect the true posterior distribution if the analysis were to be conducted on the confidential data.

(i.e. the statistic must have finite global sensitivity). This will ensure a data point can never be too "unique." This is accomplished by clamping the data. More precisely, we define the clamping function $[z] := \min\{\max\{z, -10\}, 10\}$ which truncates a value z so that it falls into the interval $[-10, 10]$. Furthermore, we let $\tilde{z} := [z]/10$ denote the normalized clamped value of z. The clamped statistic is

$$t(x_i, y_i) = ((\tilde{x}^i)^T \tilde{y}_i, \tilde{y}_i^2, (\tilde{x}_i)^T \tilde{x}_i).$$

Ignoring duplicate entries, the statistic has ℓ_1-sensitivity $\Delta = p^2 + 4p + 3$ where p is the number of predictors in the regression model (in this example $p = 2$) .[4] Using the Laplace mechanism, ϵ-DP privacy can thus be achieved by adding i.i.d. Laplace$(0, \Delta/\epsilon)$ error to each unique entry. A tighter bound on sensitivity can be achieved using other techniques, see Awan and Slavković (2020).

1. `latent_f`: Since the privacy mechanism involves injecting noise into the design matrix, it is not possible to use the standard approach where one assumes the design matrix is a fixed, known constant. Hence to draw a sample from the latent data generating process we use the relation $f(x, y) = f(x)f(y \mid x)$. In this formulation, it is necessary to specify a distribution on the covariates x.

```
latent_f <- function(theta) {
  xmat <- MASS::mvrnorm(50 , mu = c(.9,-1.17), Sigma = diag(2))
  y <- cbind(1,xmat) %*% theta + rnorm(50, sd = sqrt(2))
  cbind(y,xmat)
}
```

2. `post_f`: Given confidential data X we can derive the posterior analytically using a normal prior on β.

$$\begin{aligned}
\beta &\sim N_{p+1}(0, \tau^2 I_{p+1}) \\
\beta \mid x, y &\sim N(\mu_n, \Sigma_n) \\
\Sigma_n &= (x^T x/\sigma^2 + I_{p+1}/\tau^2)^{-1} \\
\mu_n &= \Sigma_n (x^T y)/\sigma^2
\end{aligned}$$

In the example, we use $\sigma^2 = 2$ and $\tau^2 = 4$.

```
post_f <- function(dmat, theta) {
  x <- cbind(1,dmat[,-1])
  y <- dmat[,1]
```

[4]The original paper, Ju et al. (2022), contains a computation error and mistakenly uses $\Delta = p^2 + 3p + 3$

```
  ps_s2 <- solve((1/2) * t(x) %*% x + (1/4) * diag(3))
  ps_m <- ps_s2 %*% (t(x) %*% y) * (1/2)

  MASS::mvrnorm(1, mu = ps_m, Sigma = ps_s2)
}
```

3. `st_f`: The summary statistic contains duplicate entries. We can considerable reduce the dimension of the statistic by only considering unique entries. The `clamp_data` function is used to bound the statistic to give a finite global sensitivity.

```
clamp_data <- function(dmat) {
  pmin(pmax(dmat,-10),10) / 10
}

st_f <- function(xi, sdp, i) {
  xic <- clamp_data(xi)
  ydp <- xic[1]
  xdp <- cbind(1,t(xic[-1]))

  s1 <- t(xdp) %*% ydp
  s2 <- t(ydp) %*% ydp
  s3 <- t(xdp) %*% xdp

  ur_s1 <- c(s1)
  ur_s2 <- c(s2)
  ur_s3 <- s3[upper.tri(s3,diag = TRUE)][-1]
  c(ur_s1,ur_s2,ur_s3)
}
```

4. `priv_f`: Privacy Mechanism adds Laplace$(0, \Delta/\epsilon)$ error to each unique entry of the statistic. In this example, $\Delta = 15$ and $\epsilon = 10$.

```
priv_f <- function(sdp, sx) {
  sum(VGAM::dlaplace(sdp - sx, 0, 15/10, log = TRUE))
}
```

First we simulate fake data using the aforementioned privacy mechanism. In the example, we use $n = 50$ observations.

```
deltaa <- 15
epsilon <- 10
n <- 50

set.seed(1)
xmat <- MASS::mvrnorm(n, mu = c(.9,-1.17), Sigma = diag(2))
beta <- c(-1.79, -2.89, -0.66)
y <- cbind(1,xmat) %*% beta + rnorm(n, sd = sqrt(2))

#clamp the confidential data in xmat
dmat <- cbind(y,xmat)
sdp <-  apply(sapply(1:nrow(dmat), function(i) st_f(dmat[i,], sdp, i)), 1, sum)

#add Laplace noise
sdp <- sdp + VGAM::rlaplace(length(sdp), location = 0, scale = deltaa/epsilon)
```

We construct a privacy model using the `new_privacy` function and make 25,000 MCMC draws with a burn in of 1000 draws.

```
library(dapper)

dmod <- new_privacy(post_f   = post_f,
                    latent_f = latent_f,
                    priv_f   = priv_f,
                    st_f     = st_f,
```

```
                    npar     = 3,
                    varnames = c("beta0", "beta1", "beta2"))
```

```
dp_out <- dapper_sample(dmod,
                        sdp = sdp,
                        niter = 25000,
                        warmup = 1000,
                        chains = 1,
                        init_par = rep(0,3))
```

The output of the MCM run is reported below.

```
summary(dp_out)
```

```
#> # A tibble: 3 x 10
#>   variable   mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
#>   <chr>     <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
#> 1 beta0    -0.916 -0.864  1.49  1.49 -3.42 1.46   1.00     525.     890.
#> 2 beta1    -1.96  -2.26   1.41  1.12 -3.78 0.934  1.01     153.     318.
#> 3 beta2     0.734  0.727  1.30  1.37 -1.35 2.94   1.02     163.     484.
```

For comparison, we consider a Bayesian analysis where the design matrix is a fixed known constant and σ^2 is known. Using the diffuse prior $f(\beta) \propto 1$ leads to normal posterior.

$$
\begin{aligned}
f(\beta \mid x, y, \sigma^2) &\sim N(\hat{\beta}, \hat{\Sigma}) \\
\hat{\mu} &= (x^T x)^{-1} xy \\
\hat{\Sigma} &= \sigma^2 (x^T x)^{-1}
\end{aligned}
$$

The posterior can be written as a function of $s(x, y)$. Since we only have access to the noisy version s_{dp} we can attempt to reconstruct the posterior be extracting the relevant entries which is done below.

```
#x^Ty
s1 <- sdp[1:3]

#y^Ty
s2 <- sdp[4]

#x^Tx
s3 <- matrix(0, nrow = 3, ncol = 3)
s3[upper.tri(s3, diag = TRUE)] <- c(n, sdp[5:9])
s3[lower.tri(s3)] <- s3[upper.tri(s3)]
```

Because of the injected privacy noise, the reconstructed $(x^T x)^{-1}$ matrix is not positive definite. As a naive solution we use the algorithm proposed by Higham (1988) to find the closest positive semi-definite matrix as determined by the Forbenius norm. The pracma package contains an implementation via the `nearest_psd` function.

```
s3 <- pracma::nearest_spd(solve(s3))
bhat <- s3 %*% s1
sigma_hat <- 2^2 * s3
```

Figure 6 shows the posterior density estimates for the β coefficients based on s_{dp}. The density estimates indicates the naive method, which ignores the privacy mechanism, has bias and underestimates the variance. Likewise Figure 7 illustrates how dapper provides point estimates that are not far off from those that would have been obtained using the original confidential data. And the dramatic increase in the posterior variance indicates the privacy mechanism adds substantial uncertainty to the estimates.

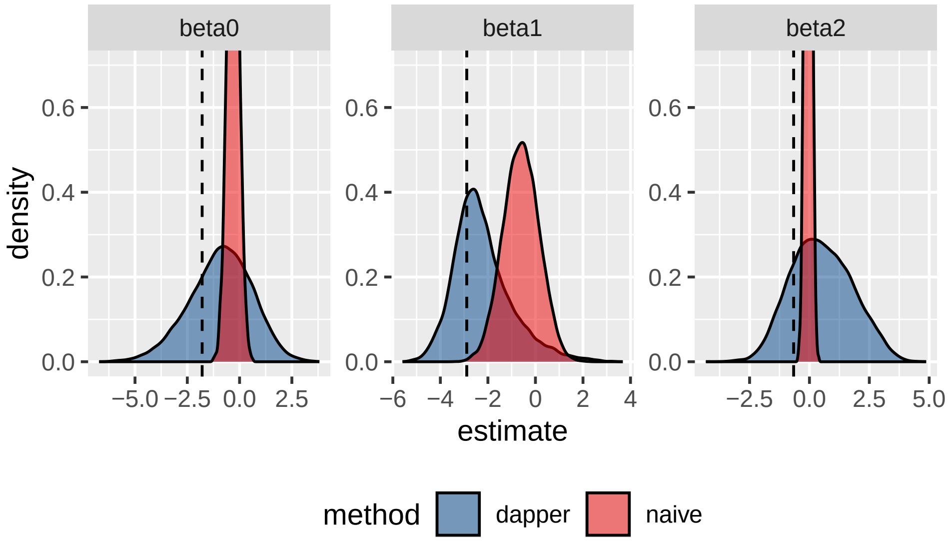

Figure 6: (Example 3) The red densities represent the posteriors of the regression coefficient that come from applying the naive analysis to the privitized data. The blue densities are the privacy aware posterior distributions. The dashed lines are the true coefficient values.

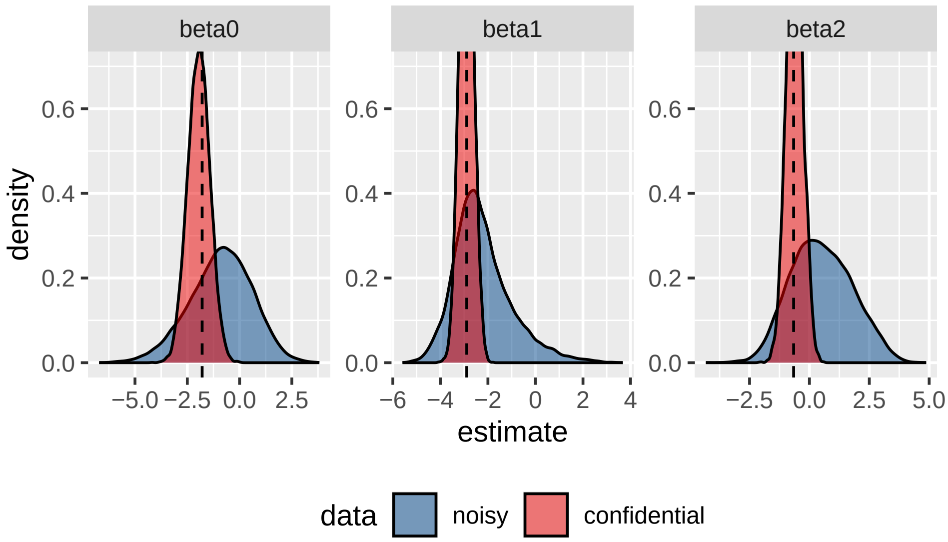

Figure 7: (Example 3) This plot compares the posteriors that would arise from applying the naive, privacy blind analysis to the confidential and privitized data. The red densities represent the posterior of the coefficient under the confidential data. The blue densities are the posterior distributions that would arise from using the privitized data. The dashed lines are the true coefficient values.

6 Discussion on Mixing and Privacy Loss Budget

Mixing can be poor when the posterior under a given privacy mechanism is much wider than the posterior that would arise using the confidential data. In other words, a small privacy budget can result in poor mixing. Intuitively, this issue arises because the step size of the chain is governed by the variance of the posterior model (step 1 of the algorithm) that assumes no privacy noise. Thus, a small privacy budget will generate a chain whose step sizes are too small to effectively explore the private posterior. The rest of this section explores a toy example that will provide insight into this phenomenon.

Suppose the confidential data consist of a single observation $x \in \mathbb{R}$, and consider the scenario where a user makes a request to view x and in return receives $s := x + \nu$, which is a noise infused version of x. For simplicity, we do not worry about constructing an ϵ-DP privacy mechanism, and take $\nu \sim N(0, \epsilon^{-2})$ for some $\epsilon > 0$. However, it will still be useful to think of ϵ as the privacy budget since smaller values of ϵ correspond to a larger amounts of noise. Using a flat prior and a normally distributed likelihood results in a normally distributed posterior described below.

$$\begin{aligned}
&f(\theta) \propto 1 \\
&s \mid x \sim N(x, \epsilon^{-2}) \\
&x \mid \theta \sim N(\theta, \sigma^2)
\end{aligned}$$

With the above model, the data augmentation process consist of the two steps

- Step 1: Sample from $x \mid \theta, s \sim N(\mu, \tau^2)$, where μ and τ are defined as:

$$\begin{aligned}
\mu &:= \frac{s/\epsilon^{-2} + \theta/\sigma^2}{1/\epsilon^{-2} + 1/\sigma^2} \\
\tau^2 &:= \frac{1}{1/\epsilon^{-2} + 1/\sigma^2}.
\end{aligned}$$

- Step 2: Sample from $\theta \mid x, s \sim N(x, \sigma^2)$.

In the setting of this example, Liu and Wu (1999) showed the Bayesian fraction of missing information gives the exact convergence rate. The Bayesian fraction of missing information, γ is defined as

$$\gamma := 1 - \frac{E[Var(\theta \mid s, x) \mid s]}{Var(\theta \mid s)} = 1 - \frac{E[Var(\theta \mid x)]}{Var(\theta \mid s)}.$$

Plugging in the appropriate quantities into the above panel gives us

$$\gamma = 1 - \frac{\sigma^2}{\sigma^2 + \epsilon^{-2}} = 1 - \frac{1}{1 + \epsilon^{-2}/\sigma^2}.$$

The chain converges faster as $\gamma \to 0$ an slower as $\gamma \to 1$. From the right hand term in the above panel, we can see γ depends only on ϵ^{-2}/σ^2 and as the privacy budget decreases (i.e. more noise is being added to x), $\gamma \to 1$.

Thus we recommend varying the privacy budget as a diagnostic for slow mixing chains. If a faster sampler is needed, and it has been determined that the privacy budget is the issue, the pseudo-likelihood scheme proposed by Andrieu and Roberts (2009) may offer significant speed ups. This scheme fits in the same data augmentation framework as `dapper` but is not implemented.

7 Summary

Currently, there is a lack of software tools privacy researchers can use to evaluate the impact of privacy mechanisms on statistical analyses. While there have been tremendous gains in the theoretical aspects of privacy, the lack of software resources to deploy and work with new privacy techniques has hampered their adoption. This gap in capability has been noted by several large industry entities who have begun building software ecosystems for working with differential privacy. However, the majority of these software tools only address privacy and not the ensuing analysis or, if it does, addresses the analysis only for specific models. Privacy researchers currently lack good tools for evaluating the impact of privacy mechanisms on a statistical analysis.

Thus **dapper** helps fill an urgent need by providing researchers a way to properly account for the noise introduced for privacy protection in their statistical analysis. A notable feature is its flexibility which allows the users to specify a custom privacy mechanism. The benefit being that **dapper** can evaluate already established privacy mechanisms as well as those that have yet to be discovered.

This package offers a significant step forward in providing general-purpose statistical inference tools for privatized data. Despite the strengths of **dapper**, it has several cumbersome requirements for good performance that limit its potential: 1) the privacy mechanism must have a closed-form density 2) a record additive statistic must be used to leverage **dapper**'s full computational potential 3) the non-private posterior sampler needs to be efficient and 4) the privacy budget cannot be too small. To improve **dapper**, future work could aim to relax some of these requirements.

References

Abowd, John M. 2018. "The U.S. Census Bureau Adopts Differential Privacy." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &Amp; Data Mining*. KDD '18. ACM. https://doi.org/10.1145/3219819.3226070.

Andrieu, Christophe, and Gareth O. Roberts. 2009. "The Pseudo-Marginal Approach for Efficient Monte Carlo Computations." *The Annals of Statistics* 37 (2). https://doi.org/10.1214/07-aos574.

Awan, Jordan, and Aleksandra Slavković. 2020. "Structure and Sensitivity in Differential Privacy: Comparing k-Norm Mechanisms." *Journal of the American Statistical Association* 116 (534): 935–54. https://doi.org/10.1080/01621459.2020.1773831.

Bickel, Peter J., Eugene A. Hammel, and John W. O'Connell. 1975. "Sex Bias in Graduate Admissions: Data from Berkeley: Measuring Bias Is Harder Than Is Usually Assumed, and the Evidence Is Sometimes Contrary to Expectation." *Science* 187 (4175): 398–404. https://doi.org/10.1126/science.187.4175.398.

Bun, Mark, and Thomas Steinke. 2016. "Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds." https://arxiv.org/abs/1605.02065.

Canonne, Clément L., Gautam Kamath, and Thomas Steinke. 2021. "The Discrete Gaussian for Differential Privacy." https://arxiv.org/abs/2004.00010.

Dalenius, Tore, and Steven P. Reiss. 1982. "Data-Swapping: A Technique for Disclosure Control." *Journal of Statistical Planning and Inference* 6 (1): 73–85. https://doi.org/10.1016/0378-3758(82)90058-1.

Ding, Bolin, Janardhan Kulkarni, and Sergey Yekhanin. 2017. "Collecting Telemetry Data Privately." https://arxiv.org/abs/1712.01524.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. "Calibrating Noise to Sensitivity in Private Data Analysis." In *Lecture Notes in Computer Science*, 265–84. Springer Berlin Heidelberg. https://doi.org/10.1007/11681878_14.

Erlingsson, Úlfar, Vasyl Pihur, and Aleksandra Korolova. 2014. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response." In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS'14. ACM. https://doi.org/10.1145/2660267.2660348.

Gong, Ruobin. 2022. "Transparent Privacy Is Principled Privacy." *Harvard Data Science Review*, no. Special Issue 2 (June). https://doi.org/10.1162/99608f92.b5d3faaa.

Higham, Nicholas J. 1988. "Computing a Nearest Symmetric Positive Semidefinite Matrix." *Linear Algebra and Its Applications* 103 (May): 103–18. https://doi.org/10.1016/0024-3795(88)90223-6.

Ju, Nianqiao, Jordan Awan, Ruobin Gong, and Vinayak Rao. 2022. "Data Augmentation MCMC for Bayesian Inference from Privatized Data." In *Advances in Neural Information Processing Systems*, edited by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. https://openreview.net/forum?id=tTWCQrgjuM.

Karwa, Vishesh, Dan Kifer, and Aleksandra B. Slavković. 2015. "Private Posterior Distributions from Variational Approximations." https://arxiv.org/abs/1511.07896.

Kenny, Christopher T., Shiro Kuriwaki, Cory McCartan, Evan T. R. Rosenman, Tyler Simko, and Kosuke Imai. 2021. "The Use of Differential Privacy for Census Data and Its Impact on Redistricting: The Case of the 2020 u.s. Census." *Science Advances* 7 (41). https://doi.org/10.1126/sciadv.abk3283.

Labs, Tumult. 2022. "SafeTab: DP Algorithms for 2020 Census Detailed DHC Race & Ethnicity." https://www2.census.gov/about/partners/cac/sac/meetings/2022-03/dhc-attachment-1-safetab-dp-algorithms.pdf.

Liu, Jun S., and Ying Nian Wu. 1999. "Parameter Expansion for Data Augmentation." *Journal of the American Statistical Association* 94 (448): 1264–74. https://doi.org/10.1080/01621459.1999.10473879.

Robert, Christian P., and George Casella. 2004. *Monte Carlo Statistical Methods*. *Springer Texts in Statistics*. Springer New York. https://doi.org/10.1007/978-1-4757-4145-2.

Santos-Lozada, Alexis R., Jeffrey T. Howard, and Ashton M. Verdery. 2020. "How Differential Privacy Will Affect Our Understanding of Health Disparities in the United States." *Proceedings of the*

National Academy of Sciences 117 (24): 13405–12. https://doi.org/10.1073/pnas.2003714117.

Tang, Jun, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. 2017. "Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12." https://arxiv.org/abs/1709.02753.

Wang, Yue, Daniel Kifer, Jaewoo Lee, and Vishesh Karwa. 2018. "Statistical Approximating Distributions Under Differential Privacy." *Journal of Privacy and Confidentiality* 8 (1). https://doi.org/10.29012/jpc.666.

Warner, Stanley L. 1965. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias." *Journal of the American Statistical Association* 60 (309): 63–69. https://doi.org/10.1080/01621459.1965.10480775.

Williams, Oliver, and Frank Mcsherry. 2010. "Probabilistic Inference and Differential Privacy." In *Advances in Neural Information Processing Systems*, edited by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. Vol. 23. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2010/file/fb60d411a5c5b72b2e7d3527cfc84fd0-Paper.pdf.

Winkler, Richelle L., Jaclyn L. Butler, Katherine J. Curtis, and David Egan-Robertson. 2021. "Differential Privacy and the Accuracy of County-Level Net Migration Estimates." *Population Research and Policy Review* 41 (2): 417–35. https://doi.org/10.1007/s11113-021-09664-5.

Kevin Eng
Rutgers University
Department of Statistics
Piscataway, NJ 08854
https://www.britannica.com/animal/quokka
ke157@stat.rutgers.edu

Jordan A. Awan
Purdue University
Department of Statistics
West Lafayette, IN 47907
https://jordan-awan.com/
jawan@purdue.edu

Nianqiao Phyllis Ju
Purdue University
Department of Statistics
West Lafayette, IN 47907
https://nianqiaoju.github.io/
nianqiao@purdue.edu

Vinayak A. Rao
Purdue University
Department of Statistics
West Lafayette, IN 47907
https://varao.github.io/
varao@purdue.edu

Ruobin Gong
Rutgers University
Department of Statistics
Piscataway, NJ 08854
https://ruobingong.github.io/
ruobin.gong@rutgers.edu