Data Augmentation for Privacy Aware Analysis

DP Group

September 2023

Abstract

This paper serves as a reference and introduction on using the **mcDP** R package. The goal of this package is to provide tools to explore the impact of different privacy regimes on a Bayesian analysis. A key strength of this framework is the ability to target the exact posterior in settings where the likelihood is too complex to analytically express. The main purpose of the package is to ingest four model components and return a Markov chain targeting the posterior given the privatized data. These model components are the (1) naive posterior sampler (2) likelihood sampler (3) privacy mechanism and (4) summary statistic.

Introduction

Introduce some basic notation

1. $\pi(\theta \mid x)$.
2. $f(x \mid \theta)$.
3. $\eta(s_{dp} \mid x)$.
4. $T(x)$.

Simple Binomial Proportion

Reference the transportation example. Discrete Gaussian distribution

Definition 0.1. Let $\mu, \sigma \in \mathbb{R}$ and $\sigma > 0$. The discrete Gaussian distribution has probability distribution

$$P(X = x) = \frac{\exp(-(x-\mu)^2/2\sigma^2)}{\sum_{y \in \mathbb{Z}} \exp(-(y-\mu)^2/2\sigma^2}$$

Simple Example

Assume we add noise to sample mean estimate.

Linear Regression with Clamping

(dp paper) considers a simple linear regression model with

```
set.seed(1)
deltaa <- 13
n <- 100
epsilon <- 100
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))
z <- tstat(cbind(y,xmat))

## Error in tstat(cbind(y, xmat)):  could not find function "tstat"

sdp <- z + VGAM::rlaplace(length(z), location = 0, scale = deltaa/epsilon)

## Error in eval(expr, envir, enclos):  object 'z' not found

sdp

## Error in eval(expr, envir, enclos):  object 'sdp' not found
```

```
#define model
post_smpl <- function(dmat, theta) {
  x <- cbind(1,dmat[,-1])
  y <- dmat[,1]

  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)
}

lik_smpl <- function(theta) {
  xmat <- c(rnorm(1, mean = .9), rnorm(1, mean = -1.17))
  y <- c(1,xmat) %*% theta + rnorm(1, sd = sqrt(2))
  c(y,xmat)
}

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

tstat <- function(dmat) {
  sdp_mat <- clamp_data(dmat)
```

```
  ydp <- sdp_mat[,1, drop = FALSE]
  xdp <- cbind(1,sdp_mat[,-1, drop = FALSE])

  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)
}

st_update <- function(st, xs, xo) {
  st - tstat(t(xo)) + tstat(t(xs))
}

st_init <- function(dmat) {
  tstat(dmat)
}


gen_priv_zt <- function(epsilon) {
  function(sdp, zt) {
    sum(VGAM::dlaplace(sdp - zt, 0, deltaa/epsilon, TRUE))
  }
}
```

```
dmod <- new_privacy(post_smpl = post_smpl,
                    lik_smpl = lik_smpl,
                    ll_priv_mech = gen_priv_zt(epsilon),
                    st_update = st_update,
                    st_calc = st_init,
                    npar = 3)

## Error in new_privacy(post_smpl = post_smpl, lik_smpl = lik_smpl, ll_priv_mech
= gen_priv_zt(epsilon), :  could not find function "new_privacy"

tmp <- mcmc_privacy(dmod,
                    sdp = sdp,
                    nobs = n,
                    init_par = beta,
                    niter = 1000,
                    chains = 1,
                    varnames = c("beta0", "beta1", "beta2"))
```

```
## Error in mcmc_privacy(dmod, sdp = sdp, nobs = n, init_par = beta,
niter = 1000, :  could not find function "mcmc_privacy"
```

```
posterior::summarize_draws(tmp$chain)
```

```
## Error in eval(expr, envir, enclos):  object 'tmp' not found
```