Bayesian Synthesis Models

Jingchen (Monika) Hu

Vassar College

Data Confidentiality

Outline

1. Introduction

2. Preserving relationships and Bayesian models

3. Bayesian synthesis models estimation

4. Generating synthetic values for sensitive variables

5. Miscellany

Outline

1. Introduction
2. Preserving relationships and Bayesian models
3. Bayesian synthesis models estimation
4. Generating synthetic values for sensitive variables
5. Miscellany

Adding random noise

- Add noise to
  - provide privacy protection: noise.
  - preserve important relationships: signal.
- For a sensitive continuous variable, e.g. family income Y_i, we can add noise Y_i^* from a normal centered at 0 (why 0?):

$$Y_i^* \sim \mathrm{Normal}(0, \sigma)$$

Adding random noise

- Add noise to
  - provide privacy protection: noise.
  - preserve important relationships: signal.
- For a sensitive continuous variable, e.g. family income Y_i, we can add noise Y_i^* from a normal centered at 0 (why 0?):

$$Y_i^* \sim \mathrm{Normal}(0, \sigma)$$

- This approach preserves distributional characteristics of family income, but not its relationships with other variable(s).

The CE sample

- The Consumer Expenditure Surveys (CE)
  - conducted by the U.S. Census Bureau for the U.S. Bureau of Labor Statistics.
  - contains data on expenditures, income, and tax statistics about consumer units (CU) across the country.
  - provides information on the buying habits of U.S. consumers.

The CE sample

- The Consumer Expenditure Surveys (CE)
  - conducted by the U.S. Census Bureau for the U.S. Bureau of Labor Statistics.
  - contains data on expenditures, income, and tax statistics about consumer units (CU) across the country.
  - provides information on the buying habits of U.S. consumers.

```r
CEdata <- read.csv(file = "CEdata.csv")
head(CEdata)
```

```
##   UrbanRural Income Race Expenditure
## 1          1  98600    1    5972.167
## 2          1  24360    1    5854.500
## 3          1  80200    1    5506.667
## 4          1 150500    1    8968.891
## 5          1 130000    1   10092.833
## 6          1  32836    1    5520.267
```

The CE sample cont'd

Variable	Information
UrbanRural	Binary; the urban / rural status of CU: 1 = Urban, 2 = Rural.
Income	Continuous; the amount of CU income bfore taxes in past 12 months.
Race	Categorical; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race.
Expenditure	Continuous; CU's total expenditures in last quarter.

The synthesis goals and questions

- Suppose `Income` is deemed the most sensitive among the four variables, and we want to add noise to it for privacy protection.
- To complete this, we can consider the following questions:
    1. What kind of relationships do you think are important to preserve in the confidential data?

The synthesis goals and questions

- Suppose `Income` is deemed the most sensitive among the four variables, and we want to add noise to it for privacy protection.
- To complete this, we can consider the following questions:
  1. What kind of relationships do you think are important to preserve in the confidential data?
  2. Given your synthesis goals (i.e. providing sufficient privacy protection while preserving important relationships), what kind of Bayesian models do you think are appropriate?

The synthesis goals and questions

- Suppose `Income` is deemed the most sensitive among the four variables, and we want to add noise to it for privacy protection.
- To complete this, we can consider the following questions:
  1. What kind of relationships do you think are important to preserve in the confidential data?
  2. Given your synthesis goals (i.e. providing sufficient privacy protection while preserving important relationships), what kind of Bayesian models do you think are appropriate?
  3. How to estimate the chosen Bayesian models, and how can you generate synthetic data from them?

The synthesis goals and questions

- Suppose `Income` is deemed the most sensitive among the four variables, and we want to add noise to it for privacy protection.
- To complete this, we can consider the following questions:
  1. What kind of relationships do you think are important to preserve in the confidential data?
  2. Given your synthesis goals (i.e. providing sufficient privacy protection while preserving important relationships), what kind of Bayesian models do you think are appropriate?
  3. How to estimate the chosen Bayesian models, and how can you generate synthetic data from them?
  4. And finally, how do you evaluate whether your methods have achieved your goals?

The synthesis goals and questions

- Suppose Income is deemed the most sensitive among the four variables, and we want to add noise to it for privacy protection.
- To complete this, we can consider the following questions:
  1. What kind of relationships do you think are important to preserve in the confidential data?
  2. Given your synthesis goals (i.e. providing sufficient privacy protection while preserving important relationships), what kind of Bayesian models do you think are appropriate?
  3. How to estimate the chosen Bayesian models, and how can you generate synthetic data from them?
  4. And finally, how do you evaluate whether your methods have achieved your goals?
- This lecture focuses on Questions 1-3.

Outline

1. Introduction
2. Preserving relationships and Bayesian models
3. Bayesian synthesis models estimation
4. Generating synthetic values for sensitive variables
5. Miscellany

Important relationships?

- What kind of relationships between `Income` and the other three (`UrbanRural`, `Race`, `Expenditure`) do you think are important to preserve? Why?
- When we say "preserve", what do we mean?

Important relationships?

- What kind of relationships between `Income` and the other three (`UrbanRural`, `Race`, `Expenditure`) do you think are important to preserve? Why?
- When we say "preserve", what do we mean?
  - Inferences done on synthetic data are "close" to those done on confidential data.

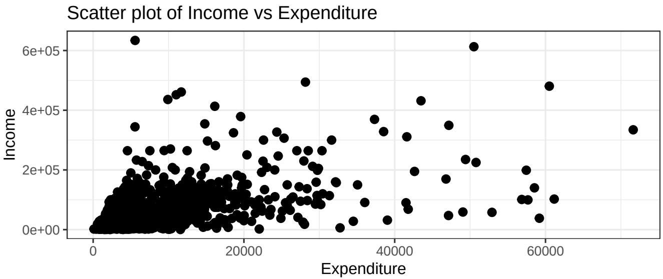
Important relationships?

- What kind of relationships between `Income` and the other three (`UrbanRural`, `Race`, `Expenditure`) do you think are important to preserve? Why?
- When we say "preserve", what do we mean?
    - Inferences done on synthetic data are "close" to those done on confidential data.
    - However, too "close" could mean less privacy protection.
    - We need to strike the balance between utility and disclosure risks.

Example: Relationship between Income and Expenditure

- Suppose we want to preserve the relationship between `Income` and `Expenditure`.
- Visualizing this relationship on the raw scale of these two variables shows fanning trend.

```
ggplot(CEdata, aes(x = Expenditure, y = Income)) +
  geom_point(size = 1) +
  labs(title = "Scatter plot of Income vs Expenditure") +
  theme_bw(base_size = 6, base_family = "")
```



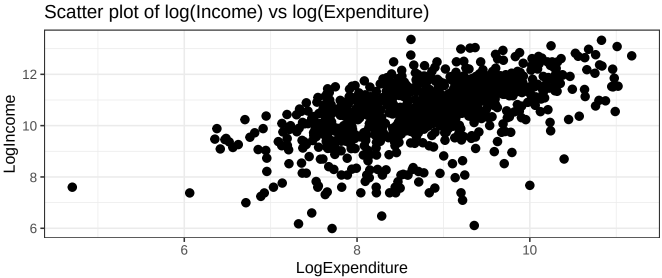
Income and Expenditure: log scale

- On the log scale, the relationship between log(Income) and log(Expenditure) appears to be linear.

```
CEdata$LogIncome <- log(CEdata$Income)
CEdata$LogExpenditure <- log(CEdata$Expenditure)
```

```
ggplot(CEdata, aes(x = LogExpenditure, y = LogIncome)) +
  geom_point(size = 1) +
  labs(title = "Scatter plot of log(Income) vs log(Expenditure)") +
  theme_bw(base_size = 6, base_family = "")
```

Income and Expenditure: log scale cont'd



- To capture and preserve this linear relationship, a Bayesian linear regression model of `log(Expenditure)` on `log(Income)` seems a good choice.

A Bayesian simple linear regression model

- Let Y_i be the `log(Income)` and X_i be the `log(Expenditure)` for CU i, a Bayesian simple linear regression model can be expressed as:

$$\begin{aligned} Y_i \mid \mu_i, \sigma &\overset{ind}{\sim} \mathrm{Normal}(\mu, \sigma) && (1) \\ \mu_i &= \beta_0 + \beta_1 X_i && (2) \end{aligned}$$

A Bayesian simple linear regression model

- Let Y_i be the `log(Income)` and X_i be the `log(Expenditure)` for CU i, a Bayesian simple linear regression model can be expressed as:

$$
\begin{aligned}
Y_i \mid \mu_i, \sigma &\overset{ind}{\sim} \mathrm{Normal}(\mu, \sigma) & (1) \\
\mu_i &= \beta_0 + \beta_1 X_i & (2)
\end{aligned}
$$

- The expected `log(Income)` is μ_i, which is a linear function of `log(Expenditure)` X_i through the intercept parameter β_0 and the slope parameter β_1.
- The intercept β_0: the expected `log(Income)` μ_i for CU i that has zero `log(Expenditures)` (i.e. $X_i = 0$).
- The slope β_1: the chance in the expected `log(Income)` μ_i when the `log(Expenditures)` of CU i increases by 1 unit (i.e. X_i increases by \$log(1)).

Outline

1 Introduction

2 Preserving relationships and Bayesian models

3 Bayesian synthesis models estimation

4 Generating synthetic values for sensitive variables

5 Miscellany

A weakly informative prior

- To estimate the proposed Bayesian linear regression model, we need to assign appropriate prior distributions for all parameters in the model: $\{\beta_0, \beta_1, \sigma\}$.

A weakly informative prior

- To estimate the proposed Bayesian linear regression model, we need to assign appropriate prior distributions for all parameters in the model: $\{\beta_0, \beta_1, \sigma\}$.
- If we have limited prior information about these parameters, we could use a weakly informative prior distribution. Assuming independence of the three parameters:

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma). \tag{3}$$

A weakly informative prior cont'd

- We can then give individual weakly informative prior for each parameter:

$$\begin{aligned} \beta_0 &\sim \text{Normal}(\mu_0, s_0) && (4) \\ \beta_1 &\sim \text{Normal}(\mu_1, s_1) && (5) \\ 1/\sigma^2 &\sim \text{Gamma}(a, b), && (6) \end{aligned}$$

where we can use $\mu_0 = \mu_1 = 0$, $s_0 = s_1 = 100$, and $a = b = 1$.

MCMC simulation by JAGS

- Let's use JAGS (Just Another Gibbs Sampler) to estimate our chosen Bayesian simple linear regression model.

MCMC simulation by JAGS

- Let's use JAGS (Just Another Gibbs Sampler) to estimate our chosen Bayesian simple linear regression model.

- We will obtain pre-specified number of posterior parameter draws from the JAGS output, which will be used for synthetic data generation through the posterior predictive distribution.

MCMC simulation by JAGS

- Let's use JAGS (Just Another Gibbs Sampler) to estimate our chosen Bayesian simple linear regression model.
- We will obtain pre-specified number of posterior parameter draws from the JAGS output, which will be used for synthetic data generation through the posterior predictive distribution.
- Make sure that we require the `runjags` and `coda` libraries.

```r
require(runjags)
require(coda)
```

Using JAGS: part 1

- Describe the model by a script.

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

Using JAGS: part 2

- Define the data and prior parameters.

```r
y <- as.vector(CEdata$LogIncome)
x <- as.vector(CEdata$LogExpenditure)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0, "g1" = 0.0001,
                 "a" = 1, "b" = 1)

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}
```

Using JAGS: part 3

- Generate samples from the posterior distribution.

```r
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("beta0", "beta1", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 50,
                      inits = initsfunction)
```

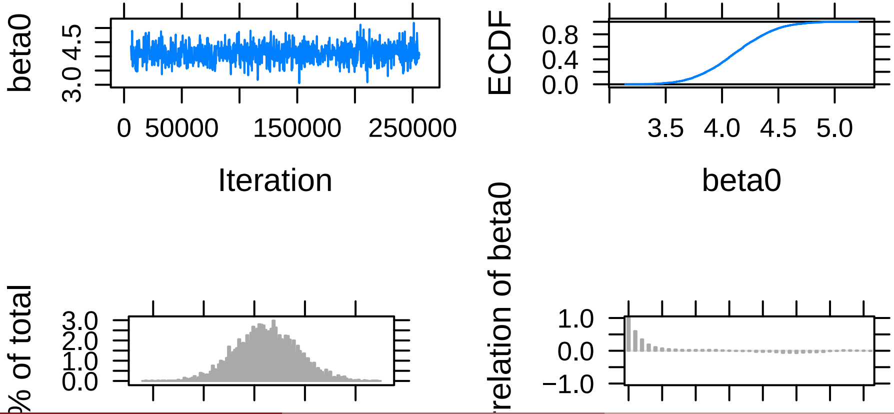
- The value of `thin` is set given MCMC diagnostics.

Using JAGS: part 4

- MCMC diagnostics (check β_1 and σ as well).

```r
plot(posterior, vars = "beta0")
```

```
## Generating plots...
```



Using JAGS: part 5

- Saving posterior parameter draws.

```
post <- as.mcmc(posterior)
```

Using JAGS: part 5

- Saving posterior parameter draws.

```
post <- as.mcmc(posterior)
```

- The key to using Bayesian synthesis models is to save posterior parameter draws of estimated parameters. These draws will be soon used to generate synthetic data given the posterior predictive distribution of the data values.

Using JAGS: part 5

- Saving posterior parameter draws.

```
post <- as.mcmc(posterior)
```

- The key to using Bayesian synthesis models is to save posterior parameter draws of estimated parameters. These draws will be soon used to generate synthetic data given the posterior predictive distribution of the data values.

- `post` contains 5000 rows and 3 columns:
  - each column corresponds to a parameter: `beta0`, `beta1`, and `sigma`.
  - each row corresponds to one of the 5000 MCMC iterations.

Using JAGS: part 5

- Saving posterior parameter draws.

```
post <- as.mcmc(posterior)
```

- The key to using Bayesian synthesis models is to save posterior parameter draws of estimated parameters. These draws will be soon used to generate synthetic data given the posterior predictive distribution of the data values.

- `post` contains 5000 rows and 3 columns:
  - each column corresponds to a parameter: `beta0`, `beta1`, and `sigma`.
  - each row corresponds to one of the 5000 MCMC iterations.

- Next, we will use the posterior parameter draws saved in `post` to generate synthetic values for `log(Income)`.

Outline

1 Introduction

2 Preserving relationships and Bayesian models

3 Bayesian synthesis models estimation

4 Generating synthetic values for sensitive variables

5 Miscellany

Generating one set of synthetic data

- The synthetic data generation process is no different from making prediction of future values. To predict future `log(Income)`, $\tilde{Y}_i$ for a CU given its `log(Expenditure)`, X_i:

$$\tilde{Y}_i \mid \beta_0, \beta_1, \sigma \overset{ind}{\sim} \mathrm{Normal}(\beta_0 + \beta_1 X_i, \sigma). \tag{7}$$

Generating one set of synthetic data cont'd

- Given X_i and one set of posterior draws of the parameters $\{\beta_0, \beta_1, \sigma\}$, we could simulate $\tilde{Y}_i$ for each all n CUs:

$$\begin{array}{rcl}
\text{simulate } E[Y_1] = \beta_0 + \beta_1 X_1 & \rightarrow & \text{sample } \tilde{Y}_1 \sim \mathrm{Normal}(E[Y_1], \sigma) \\
\text{simulate } E[Y_2] = \beta_0 + \beta_1 X_2 & \rightarrow & \text{sample } \tilde{Y}_2 \sim \mathrm{Normal}(E[Y_2], \sigma) \\
 & \vdots & \\
\text{simulate } E[Y_n] = \beta_0 + \beta_1 X_n & \rightarrow & \text{sample } \tilde{Y}_n \sim \mathrm{Normal}(E[Y_n], \sigma)
\end{array}$$

Generating one set of synthetic data cont'd

- Suppose we use one set of posterior draws, the function below returns a synthetic dataset with synthesized `log(Income)` and un-synthesized `log(Expenditure)`.

```
synthesize <- function(X, index, n){
  mean_Y <- post[index, "beta0"] +  X * post[index, "beta1"]
  synthetic_Y <- rnorm(n, mean_Y, post[, "sigma"])
  data.frame(X, synthetic_Y)
}
```

- The input `X` is a vector of the un-synthesized variable, i.e. `log(Expenditure)`.
- `index` indicates which set of posterior draws to be used.

Generating one set of synthetic data cont'd

- For example, `index = 1` of we use the first of 5000 sets. `n` is the number of observations.

```r
synthesize <- function(X, index, n){
  mean_Y <- post[index, "beta0"] +  X * post[index, "beta1"]
  synthetic_Y <- rnorm(n, mean_Y, post[, "sigma"])
  data.frame(X, synthetic_Y)
}
```

```r
n <- dim(CEdata)[1]
synthetic_one <- synthesize(CEdata$LogExpenditure, 1, n)
names(synthetic_one) <- c("logExpenditure", "logIncome_syn")
```

Generating multiple sets of synthetic data

- Typical practice generates multiple sets of synthetic data, for example, $m = 20$.
- Later, we will explore why multiple synthetic datasets are needed for data utility evaluation.

Generating multiple sets of synthetic data cont'd

- To use the last $m = 20$ sets of the obtained 5000 MCMC iterations for generating $m = 20$ synthetic datasets:

```r
n <- dim(CEdata)[1]
m <- 20
synthetic_m <- vector("list", m)
for (l in 1:m){
  synthetic_one <- synthesize(CEdata$LogExpenditure, 4980+l, n)
  names(synthetic_one) <- c("logExpenditure", "logIncome_syn")
  synthetic_m[[l]] <- synthetic_one
}
```

Generating multiple sets of synthetic data cont'd

- To use the last $m = 20$ sets of the obtained 5000 MCMC iterations for generating $m = 20$ synthetic datasets:

```r
n <- dim(CEdata)[1]
m <- 20
synthetic_m <- vector("list", m)
for (l in 1:m){
  synthetic_one <- synthesize(CEdata$LogExpenditure, 4980+l, n)
  names(synthetic_one) <- c("logExpenditure", "logIncome_syn")
  synthetic_m[[l]] <- synthetic_one
}
```

- Use a list `synthetic_m` to save all $m = 20$ synthetic datasets.
- Each synthetic dataset contains the un-synthesized `logExpenditure`, and the synthesized `logIncome_syn` from the estimated Bayesian simple linear regression model.

Outline

1 Introduction

2 Preserving relationships and Bayesian models

3 Bayesian synthesis models estimation

4 Generating synthetic values for sensitive variables

5 Miscellany

Preserve important relationships

- If `Expenditure` is deemed sensitive, what relationships do you want to preserve, why, and how?
- If `UrbanRural` is deemed sensitive, what relationships do you want to preserve, why, and how?
- If `Race` is deemed sensitive, what relationships do you want to preserve, why, and how?

Preserve important relationships

- If `Expenditure` is deemed sensitive, what relationships do you want to preserve, why, and how?
- If `UrbanRural` is deemed sensitive, what relationships do you want to preserve, why, and how?
- If `Race` is deemed sensitive, what relationships do you want to preserve, why, and how?
- If `Income` and `UrbanRural` are deemed sensitive, what relationships do you want to preserve, why, and how?

Partial synthesis vs full synthesis

- Partial synthesis: only a subset of variables / attributes are deemed sensitive and to be synthesized.
- Full synthesis: all variables / attributes are deemed sensitive and to be synthesized.

Partial synthesis vs full synthesis

- Partial synthesis: only a subset of variables / attributes are deemed sensitive and to be synthesized.
- Full synthesis: all variables / attributes are deemed sensitive and to be synthesized.

- Implications for Bayesian synthesis models?
- Implications for disclosure risks evaluation?