Methods for Risk Evaluation #3

Jingchen (Monika) Hu

Vassar College

Data Confidentiality

Outline

Outline

Overview

- Attribute disclosure refers to the intruder correctly inferring the true value(s) of synthesized variable(s) in the released synthetic datasets
- AR potentially exist in fully synthetic data and partially synthetic data

Overview

- Attribute disclosure refers to the intruder correctly inferring the true value(s) of synthesized variable(s) in the released synthetic datasets
- AR potentially exist in fully synthetic data and partially synthetic data
- Roadmap
  1. notations and setup
  2. key estimating steps (importance sampling)
  3. illustrative example: synthetic CE sample

Outline

Notations and setup

- $\mathbf{y}_i = (y_{i1}, \cdots, y_{ip})$: the vector response of observation i in the original confidential dataset, where direct identifiers (such as name or SSN) are removed
- When needed, we use j as the variable index, and $j = 1, \cdots, p$. Among the p variables
  - $\mathbf{y}_i^s$: synthesized variables
  - $\mathbf{y}_i^{us}$: un-synthesized variables
- $\mathbf{y}_i = (\mathbf{y}_i^s, \mathbf{y}_i^{us})$: the i-th observation
- $\mathbf{y} = (\mathbf{y}^s, \mathbf{y}^{us})$: the entire dataset containing n observations
  - for fully synthetic data, $\mathbf{y}^{us} = \emptyset$, therefore $\mathbf{y} = \mathbf{y}^s$
  - without loss of generality, we use $\mathbf{y} = (\mathbf{y}^s, \mathbf{y}^{us})$
- $\mathbf{Z} = (\mathbf{Z}^{(1)}, \cdots, \mathbf{Z}^{(m)})$: $m > 1$ synthetic datasets

Notations and setup cont'd

- Assumptions about intruder's knowledge and behavior
  1. the intruder intends to learn the value of $\mathbf{y}_i^s$ for some record i in $\mathbf{y}$
  2. available information to the intruder:
     - $\mathbf{y}^{us} = \{\mathbf{y}_i^{us} : i = 1, \cdots, n\}$: the un-synthesized values of all n observations
     - A: any auxiliary information known by the intruder about records in $\mathbf{y}$
     - S: any information known by the intruder about the process of generating $\mathbf{Z}$

Notations and setup cont'd

- $\mathbf{Y}_i^s$: the random variable representing the intruder's uncertain knowledge of $\mathbf{y}_i^s$
- The intruder seeks the distribution:

$$p(\mathbf{Y}_i^s \quad | \quad \mathbf{Z}, \mathbf{y}^{us}, A, S) \tag{1}$$

$$p(\mathbf{Y}_i^s = \mathbf{y}^* \quad | \quad \mathbf{Z}, \mathbf{Y}^{us}, A, S) \tag{2}$$

  - if $\mathbf{Y}_i^s$ is a vector of categorical variables, consider $\mathbf{y}^*$ as one plausible combination of categorical responses of those variables in the neighborhood of $\mathbf{y}_i$
  - if $\mathbf{Y}_i^s$ is a vector of continuous variables, consider $\mathbf{y}^*$ as one plausible combination of continuous responses of those variables in the neighborhood of $\mathbf{y}_i$ <u>within certain distance</u>

Notations and setup cont'd

- For the confidential data holder
  1. assumptions on the level of intruder's knowledge of $\mathbf{y}^{us}$, A, and S
  2. how to approximate $p(\mathbf{Y}_i^s = \mathbf{y}^* \mid \mathbf{Z}, \mathbf{Y}^{us}, A, S)$ (Bayesian thinking)

Outline

First step: Bayes' rule

$$
\begin{aligned}
p(\mathbf{Y}_i^s = \mathbf{y}^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S) \propto p(\mathbf{Z} \mid \mathbf{Y}_i^s &= \mathbf{y}^*, \mathbf{y}^{us}, A, S) \\
p(\mathbf{Y}_i^s &= \mathbf{y}^* \mid \mathbf{y}^{us}, A, S) \qquad (3)
\end{aligned}
$$

- $\mathbf{y}^*$: one possible guess of $\mathbf{Y}_i^s$ by the intruder
- $\mathbf{y}^{us}$, A, and S: available to the intruder
- $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$: the synthetic data distribution given what the intruder knows
- $p(\mathbf{Y}_i^s = \mathbf{y}^* \mid \mathbf{y}^{us}, A, S)$: the intruder's prior on $\mathbf{Y}_i^s = \mathbf{y}^*$ given $\mathbf{y}^{us}$, A, and S

Knowledge of $\mathbf{y}^{us}$

- $\mathbf{y}^{us} = \{\mathbf{y}_i^{us} : i = 1, \cdots, n\}$: the set of un-synthesized values of all n observations
- Partial synthesis: intruder has access to $\mathbf{Z}$, therefore $\mathbf{y}^{us}$ can be determined and thus available
- Full synthesis: $\mathbf{y}^{us} = \emptyset$

Knowledge of $\mathbf{y}^{us}$

- $\mathbf{y}^{us} = \{\mathbf{y}_i^{us} : i = 1, \cdots, n\}$: the set of un-synthesized values of all n observations
- Partial synthesis: intruder has access to $\mathbf{Z}$, therefore $\mathbf{y}^{us}$ can be determined and thus available
- Full synthesis: $\mathbf{y}^{us} = \emptyset$
- Without loss of generality, we keep $\mathbf{y}^{us}$

Assumptions about A

- A: auxiliary information known by the intruder about records in $\mathbf{y}$
- Numerous possible scenarios

Assumptions about A

- A: auxiliary information known by the intruder about records in $\mathbf{y}$
- Numerous possible scenarios
- "Worst case": $A = \mathbf{y}^s_{-i}$
    - the intruder knows the original values of the synthesized variables of all records except for record i
    - strong intruder knowledge and conservative
    - if AR under such conservative assumption are acceptable, AR should be acceptable for weaker assumptions
    - realistic for computing purposes (more in detail later)

Assumptions about S

- S: any information known by the intruder about the process of generating $\mathbf{Z}$
- Examples:
  1. code for the synthesizer
  2. descriptions of the synthesis model

Assumptions about S

- S: any information known by the intruder about the process of generating $\mathbf{Z}$
- Examples:
  1. code for the synthesizer
  2. descriptions of the synthesis model
- Such information sometimes can be public available with great details
  - recall the SynLBD synthesis paper

Choosing the prior $p(\mathbf{Y}_i^s = \mathbf{y}^* \mid \mathbf{y}^{us}, A, S)$

- Common practice: a uniform prior for all possible guesses $\mathbf{y}^*$
- Using a uniform prior cancels out the terms when comparing different guesses

$$\begin{aligned} p(\mathbf{Y}_i^s = \mathbf{y}^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S) \propto p(\mathbf{Z} \mid \mathbf{Y}_i^s &= \mathbf{y}^*, \mathbf{y}^{us}, A, S) \\ p(\mathbf{Y}_i^s &= \mathbf{y}^* \mid \mathbf{y}^{us}, A, S) \end{aligned}$$

- Do you think a uniform prior is reasonable? In what situation using it makes sense? When you might overestimate or underestimate the AR using uniform prior?

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$

- Independence between $\mathbf{Z}^{(l)}$'s:

$$p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S) = \prod_{l=1}^{m} p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S) \tag{4}$$

- Work with each $\mathbf{Z}^{(l)}$

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$ cont'd

- Under the "worst case" scenarior of $A = \mathbf{y}_{-i}^s$:

$$p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S), \tag{5}$$

which is very close to the distribution from which the synthetic data $\mathbf{Z}^{(l)}$ is generated, as in

$$p(\mathbf{Z}^{(l)} \mid \mathbf{y}, S) = p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}_i, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) \tag{6}$$

- $\mathbf{y}_i$ is the true record in the original confidential dataset $\mathbf{y}$
- The difference between Equations (5) and (6)?

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$ cont'd

- Under the "worst case" scenario of $A = \mathbf{y}_{-i}^s$:

$$p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S), \tag{5}$$

which is very close to the distribution from which the synthetic data $\mathbf{Z}^{(l)}$ is generated, as in

$$p(\mathbf{Z}^{(l)} \mid \mathbf{y}, S) = p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}_i, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) \tag{6}$$

- $\mathbf{y}_i$ is the true record in the original confidential dataset $\mathbf{y}$
- The difference between Equations (5) and (6)?
- The only difference in the conditioned quantities is difference between $\mathbf{y}^*$ (the random guess) and $\mathbf{y}_i$ (the true record)

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$ cont'd

- Monte Carlo approximation
- If we use Θ to denote the parameters in the synthesis model M, we could incorporate Θ draws in our estimation of $p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S)$

$$p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) = \int p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S, \Theta) \\ p(\Theta \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) d\Theta$$

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$ cont'd

$$
\begin{aligned}
p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) = \int p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S, \Theta) \\
p(\Theta \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) d\Theta
\end{aligned}
$$

- The Monte Carlo step requires re-estimation of the synthesis model M for each $\mathbf{Y}_i^s = \mathbf{y}^*$
- Could be computationally prohibitive if many possible guesses of $\mathbf{Y}_i^s$ need to be evaluated

The estimation of $p(\mathbf{Z} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A, S)$ cont'd

$$p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) = \int p(\mathbf{Z}^{(l)} \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S, \Theta)$$
$$p(\Theta \mid \mathbf{Y}_i^s = \mathbf{y}^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) d\Theta$$

- The Monte Carlo step requires re-estimation of the synthesis model M for each $\mathbf{Y}_i^s = \mathbf{y}^*$
- Could be computationally prohibitive if many possible guesses of $\mathbf{Y}_i^s$ need to be evaluated
- To avoid the re-estimation of M to draw Θ samples, we can use the importance sampling strategy
  - available draws of Θ from $p(\Theta \mid \mathbf{y})$ (the model used for generating the synthetic dataset $\mathbf{Z}^{(l)}$)
  - use them as proposals for the importance sampling algorithm

The importance sampling strategy

- Suppose we seek to estimate the expectation of some function $g(\Theta)$, where Θ has density $f(\Theta)$
- Further suppose that we have a sample $(\Theta^{(1)}, \cdots, \Theta^{(H)})$ available from a convenient distribution $f^*(\Theta)$ that slightly differs from $f(\Theta)$
- We can estimate $E_f(g(\Theta))$ using

$$E_f(g(\Theta)) \approx \frac{1}{H} \sum_{h=1}^{H} g(\Theta^{(h)}) \frac{f(\Theta^h)/f^*(\Theta^h)}{\sum_{h=1}^{H} f(\Theta^h)/f^*(\Theta^h)} \tag{7}$$

- We only require that $f(\Theta)$ and $f^*(\Theta)$ be known up to constants.

The importance sampling strategy

- Suppose we seek to estimate the expectation of some function $g(\Theta)$, where Θ has density $f(\Theta)$
- Further suppose that we have a sample $(\Theta^{(1)}, \cdots, \Theta^{(H)})$ available from a convenient distribution $f^*(\Theta)$ that slightly differs from $f(\Theta)$
- We can estimate $E_f(g(\Theta))$ using

$$E_f(g(\Theta)) \approx \frac{1}{H} \sum_{h=1}^{H} g(\Theta^{(h)}) \frac{f(\Theta^h)/f^*(\Theta^h)}{\sum_{h=1}^{H} f(\Theta^h)/f^*(\Theta^h)} \tag{7}$$

- We only require that $f(\Theta)$ and $f^*(\Theta)$ be known up to constants.
- What are our $f^*(\Theta)$ and $f(\Theta)$?

Outline

CE sample synthesis

```r
CEdata <- read.csv(file = "CEdata.csv")
CEdata$LogIncome <- log(CEdata$Income)
CEdata$LogExpenditure <- log(CEdata$Expenditure)
```

```r
n <- dim(CEdata)[1]
synthetic_one <- synthesize_loginc(CEdata$LogExpenditure,
                                   1, n, seed = 123)
names(synthetic_one) <- c("LogExpenditure", "LogIncome")
```

AR calculation for CE sample

- `m = 1` for illustration
- Intruder knows each records' `UrbanRural`, `Race`, `Expenditure` (all un-synthesized variables)
- Intruder trys to use this information to infer the true values of the synthesized variable, `Income`, based on the synthetic CE data in `CEdata_syn`

```
CEdata_org <- CEdata[, 1:4]
CEdata_syn <- as.data.frame(cbind(CEdata_org[, "UrbanRural"],
                                  exp(synthetic_one
                                      [, "LogIncome"]),
                                  cbind(CEdata_org
                                        [, c("Race",
                                             "Expenditure")])))
names(CEdata_syn) <- c("UrbanRural", "Income",
                       "Race", "Expenditure")
```

Estimating steps and assumptions

$$p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S) \propto p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A, S) \, p(Y_i^s = y^* \mid \mathbf{y}^{us}, A, S) \quad (8)$$

- Y_i^s: the univariate random variable representing the intruder's guess of the income of CU i
- y^*: one possible guess
- $\mathbf{Z}$: the synthetic CE sample (as in `CEdata_syn`)
- $\mathbf{y}^{us}$: the set of un-synthesized values of all n observations, which corresponds to the three un-synthesized variables `UrbanRural`, `Race`, `Expenditure` in the CE sample

Estimating steps and assumptions cont'd

- $A = \mathbf{y}^s_{-i}$ ("worst case" scenario)
- S: the intruder knows that the synthesis model is a Bayesian linear regression
- $p(Y_i^s = y^* \mid \mathbf{y}^{us}, A, S)$: assume a uniform prior, that is, all possible guesses of y^* are equally likely

$$p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S) \tag{9}$$

Estimating steps and assumptions cont'd

- Monte Carlo approximation
- Θ: the parameters in the synthesis model M

$$p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S) = \int p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S, \Theta) \\ p(\Theta \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S) d\Theta \quad (10)$$

- What are Θ in the CE example?

Estimating steps and assumptions cont'd

- Monte Carlo approximation
- Θ: the parameters in the synthesis model M

$$p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) = \int p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S, \Theta) \\ p(\Theta \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) d\Theta \quad (10)$$

- What are Θ in the CE example?
- $\Theta = \{\beta_0, \beta_1, \sigma\}$ in the Bayesian simple linear regression synthesis model M

Estimating steps and assumptions cont'd

- The importance sampling strategy

$$E_f(g(\Theta)) \approx \frac{1}{H}\sum_{h=1}^{H} g(\Theta^{(h)})\frac{f(\Theta^h)/f^*(\Theta^h)}{\sum_{h=1}^{H} f(\Theta^h)/f^*(\Theta^h)}$$

- Define $g(\Theta)$:

$$g(\Theta) = p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) \tag{11}$$

- We approximate the expectation of each $g(\Theta)$ with respect to

$$f(\Theta) = p(\Theta \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}_{-i}^s, S) \tag{12}$$

- While trying to utilize samples $(\Theta^{(1)}, \cdots, \Theta^{(H)})$ from a convenient distribution

$$f^*(\Theta) = p(\Theta \mid \mathbf{y}, S) \tag{13}$$

Estimating steps and assumptions cont'd

- The importance sampling strategy

$$\begin{aligned} g(\Theta^{(h)}) &= p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S, \Theta^{(h)}) \\ &= \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(- \frac{(\tilde{y}_i - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right), \quad (14) \end{aligned}$$

- $\tilde{y}_i$: the synthetic `log(Income)`
- X_i: the un-synthesized `log(Expenditure)` of CU i in the synthetic dataset $\mathbf{Z}$ (as in `CEdata_syn`)

Estimating steps and assumptions cont'd

- The importance sampling strategy
- Obtain $p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A = \mathbf{y}^s_{-i}, S) = \frac{1}{H}\sum_{h=1}^{H} p_h q_h$ where

$$
\begin{aligned}
p_h &= \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(\tilde{y}_i - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right) \\
q_h &= \frac{\left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(y^* - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right) / \left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(y_i - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right)}{\sum_{h=1}^{H} \left(\left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(y^* - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right) / \left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(y_i - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right) \right)}
\end{aligned}
$$

- y^* is the guessed value
- y_i is the true value for CU i's `log(Income)`

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example

- We need to work with the logarithm of `Income` and `Expenditure` in `CEdata_org` and `CEdata_syn`
  - the model M is fitted with logged continuous variables
- For ease of computation later, we round the logged values to 1 decimal point

```
CEdata_org$LogIncome <- round(log(CEdata_org$Income),
                              digits = 1)
CEdata_org$LogExpenditure <- round(log(CEdata_org$Expenditure),
                                   digits = 1)
CEdata_syn$LogIncome <- round(log(CEdata_syn$Income),
                              digits = 1)
CEdata_syn$LogExpenditure <- round(log(CEdata_syn$Expenditure),
                                   digits = 1)
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

- For illustration purpose, we demonstrate with CU 8: $y_i = 11.6, \tilde{y}_1 = 10.1, X_1 = 9.8$

```
i <- 8
y_i <- CEdata_org$LogIncome[i]
y_i_guesses <- seq((y_i - 2.5), (y_i + 2.5), 0.5)
X_i <- CEdata_syn$LogExpenditure[i]
G <- length(y_i_guesses)
```

- Assume a collection of 11 possible guesses: $\{9.1, 9.6, 10.1, 10.6, 11.1, 11.6, 12.1, 12.6, 13.1, 13.6, 14.1\}$
- Use a uniform prior, $p(Y_i^s = y^* \mid \mathbf{y}^{us}, A, S) = \frac{1}{11}$

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

- Use the importance strategy with $H = 50$ parameter draws of $\Theta = \{\beta_0, \beta_1, \sigma\}$ from the Bayesian simple linear regression synthesis model
- The parameter draws are saved in `post`

```
H <- 50
beta0_draws <- post[1:H, "beta0"]
beta1_draws <- post[1:H, "beta1"]
sigma_draws <- post[1:H, "sigma"]
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

- For computational stability, we use the `compute_logsumexp()` function below in calculating $\log(p_h q_h)$
- $p_h = \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma^{(h)}} \exp\left(-\frac{(\tilde{y}_i - \beta_0^{(h)} - \beta_1^{(h)} X_i)^2}{2(\sigma^{(h)})^2} \right) \right)$: take product of many normal pdfs

$$\log\left(\sum_{i=1}^{n} \exp(x_i)\right) = a + \log\left(\sum_{i=1}^{n} \exp(x_i - a)\right), \tag{15}$$

where $a = \max_i x_i$.

```
compute_logsumexp <- function(log_vector){
  log_vector_max <- max(log_vector)
  exp_vector <- exp(log_vector - log_vector_max)
  sum_exp <- sum(exp_vector)
  log_sum_exp <- log(sum_exp) + log_vector_max
  return(log_sum_exp)
}
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

```
CU_i_logZ_all <- rep(NA, G)
for (g in 1:G){
  q_sum_H <- sum((dnorm(y_i_guesses[g],
                        mean = (beta0_draws + beta1_draws * X_i),
                        sd = sigma_draws)) /
            (dnorm(y_i, mean = (beta0_draws + beta1_draws * X_i),
                   sd = sigma_draws)))
  log_pq_h_all <- rep(NA, H)
  for (h in 1:H){
    log_p_h <- sum(log(dnorm(CEdata_syn$LogIncome,
                             mean = (beta0_draws[h] + beta1_draws[h] *
                                       CEdata_syn$LogExpenditure),
                             sd = sigma_draws[h])))
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

```
    log_q_h <- log(((dnorm(y_i_guesses[g],
                       mean = (beta0_draws[h] + beta1_draws[h] * X_i),
                       sd = sigma_draws[h])) /
            (dnorm(y_i, mean = (beta0_draws[h] + beta1_draws[h] * X_i),
                   sd = sigma_draws[h]))) / q_sum_H)
    log_pq_h_all[h] <- log_p_h + log_q_h
  }
  CU_i_logZ_all[g] <- compute_logsumexp(log_pq_h_all)
}
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

- With uniform prior, output `CU_i_logZ_all` is $\log(p(\mathbf{Z} \mid Y_i^s = y^*, \mathbf{y}^{us}, A, S)) \propto \log(p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S))$
- To re-normalize and obtain probabilities of each of $\log(p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S))$, we can apply the log-sum-exp trick again

```r
prob <- exp(CU_i_logZ_all - max(CU_i_logZ_all)) /
  sum(exp(CU_i_logZ_all - max(CU_i_logZ_all)))
outcome <- as.data.frame(cbind(y_i_guesses, prob))
names(outcome) <- c("guess", "probability")
outcome[order(outcome$probability, decreasing = TRUE), ]
```

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

```
##    guess probability
## 8   12.6  0.09231563
## 7   12.1  0.09228750
## 9   13.1  0.09204320
## 6   11.6  0.09203442
## 5   11.1  0.09160126
## 10  13.6  0.09136939
## 4   10.6  0.09099926
## 3   10.1  0.09020571
## 11  14.1  0.09017674
## 2    9.6  0.08916632
## 1    9.1  0.08780057
```

- The true value for CU 8, $y_i = 11.6$ (with $\tilde{y}_i = 10.1, X_i = 9.8$), has a probability of 0.0916 out of 1 to be guessed correctly, when compared to 10 other simular values in the neighborhood of 11.6
- It is ranked 4 among the 11 possible guesses

Calculating $p(Y_i^s = y^* \mid \mathbf{Z}, \mathbf{y}^{us}, A, S)$ for the CE example cont'd

As a comparison, CU 10 $(y_i = 11.6, \tilde{y}_i = 10.7, X_i = 9.5)$

```
##    guess probability
## 8   12.6  0.09247509
## 7   12.1  0.09236756
## 9   13.1  0.09225174
## 6   11.6  0.09201971
## 10  13.6  0.09158484
## 5   11.1  0.09149332
## 4   10.6  0.09081751
## 11  14.1  0.09034931
## 3   10.1  0.08998757
## 2    9.6  0.08896616
## 1    9.1  0.08768719
```

Final comments

- We can repeat this calculation process for all $i \in 1, \cdots, n = 994$ observations in the CE sample (write a function)
- Report the normalized probability of the true value being guessed correctly, as well as its ranking among the 11 possible guesses within the neighborhood
- Summarize / visualize the distributions of probability and rank