Methods for Utility Evaluation #2

Jingchen (Monika) Hu

Vassar College

Data Confidentiality

Outline

Outline

Analysis-specific measures

- Continuous variables: e.g. mean, median, regression coefficient
- Categorical variables: e.g. contigency tables, toy example of one-way interaction

Race	Original	Synthetic
1	0.8	0.9
2	0.1	0.02
3	0.25	0.02
4	0.25	0.02
5	0.25	0.02
6	0.25	0.02

Analysis-specific measures

- Continuous variables: e.g. mean, median, regression coefficient
- Categorical variables: e.g. contigency tables, toy example of one-way interaction

Race	Original	Synthetic
1	0.8	0.9
2	0.1	0.02
3	0.25	0.02
4	0.25	0.02
5	0.25	0.02
6	0.25	0.02

$$\frac{0.8-0.9}{0.8} + \frac{0.1-0.2}{0.1} + \frac{0.25-0.02}{0.25} + \frac{0.25-0.02}{0.25} + \frac{0.25-0.02}{0.25} + \frac{0.25-0.02}{0.25}$$

Analysis-specific measures cont'd

- Geographical variables: e.g. heat map

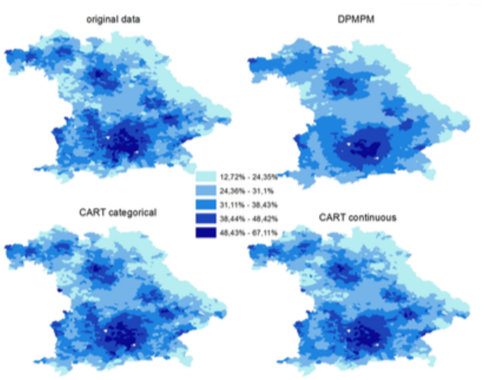

Figure 1: Share of high wage earners in Bavaria by ZIP code level.

From Drechsler, J. and Hu, J. (2019+)

Outline

Overview

- Synthetic datasets (of the same original dataset) are different from one another
    - due to uncertainty in the posterior predictive simulation
    - how can we account for the uncertainty?
    - how can we be confident the analysis results produce valid inferences?

Overview

- Synthetic datasets (of the same original dataset) are different from one another
  - due to uncertainty in the posterior predictive simulation
  - how can we account for the uncertainty?
  - how can we be confident the analysis results produce valid inferences?
- Generate multiple datasets, $m > 1$
  - variability from synthetic dataset to another
  - use appropriate combining rules
  - partially synthetic vs fully synthetic

Outline

Combining rules (partial synthesis)

- Q: a univariate estimand
  - e.g. a population mean of a univariate outcome or a univariate regression coefficient of a regression analysis
- q and v: the point estimate and variance estimate of Q from the confidential, original data
  - q and v are not available unless one has access to the original data
  - q and v are estimates from a sample, for example, when Q is a population mean, following the Central Limit Theorem
    - ★ $v = \frac{\sigma^2}{n}$ where σ is the population standard deviation if available
    - ★ $v = \frac{s}{n}$ where s is the sample standard deviation

Combining rules (partial synthesis) cont'd

- $\mathbf{Z} = (\mathbf{Z}^{(1)}, \cdots, \mathbf{Z}^{(m)})$: the set of m partially synthetic datasets
- $q^{(i)}$ and $v^{(i)}$: the values of q and v in the i-th synthetic dataset, $\mathbf{Z}^{(i)}$
- Calculate:

$$\bar{q}_m = \sum_{i=1}^{m} \frac{q^{(i)}}{m}, \quad (1)$$

$$b_m = \sum_{i=1}^{m} \frac{(q^{(i)} - \bar{q}_m)^2}{m-1}, \quad (2)$$

$$\bar{v}_m = \sum_{i=1}^{m} \frac{v^{(i)}}{m}. \quad (3)$$

Combining rules (partial synthesis) cont'd

- Use $\bar{q}_m$ as the point estimate of Q
- Use T_p as the variance estimate of $\bar{q}_m$

$$T_p = \frac{b_m}{m} + \bar{v}_m \tag{4}$$

Combining rules (partial synthesis) cont'd

- Use $\bar{q}_m$ as the point estimate of Q
- Use T_p as the variance estimate of $\bar{q}_m$

$$T_p = \frac{b_m}{m} + \bar{v}_m \tag{4}$$

- When the sample size of the synthetic data n is large, use a t distribution with degrees of freedom $v_p = (m-1)\left(1 + \frac{\bar{v}_m}{b_m/m}\right)^2$ to make inferences for estimand Q
- Obtain a 95% confidence interval for Q as:

$$\left(\bar{q}_m - t_{v_p}(0.975) \times \sqrt{\frac{b_m}{m} + \bar{v}_m},\ \bar{q}_m + t_{v_p}(0.975) \times \sqrt{\frac{b_m}{m} + \bar{v}_m}\right) \tag{5}$$

  - $t_{v_p}(0.975)$ is the t score at 0.975 with degrees of freedom v_p

Example: synthetic CE sample

- Previously, we have worked with the CE sample:
  - a Bayesian simple linear regression synthesis model
  - synthesize `log(Income)` given `log(Expenditure)`
  - one synthetic dataset saved in `synthetic_one`

Example: synthetic CE sample

- Previously, we have worked with the CE sample:
  - a Bayesian simple linear regression synthesis model
  - synthesize `log(Income)` given `log(Expenditure)`
  - one synthetic dataset saved in `synthetic_one`

- Now, synthesize $m = 20$ synthetic datasets and saved in `synthetic_m`

```
n <- dim(CEdata)[1]
m <- 20
synthetic_m <- vector("list", m)
for (i in 1:m){
  set.seed(123)
  seed <- round(runif(1, 1, 1000))
  synthetic_one <- synthesize_loginc(CEdata$LogExpenditure, 4980 + i,
                                     n, seed)
  names(synthetic_one) <- c("LogExpenditure", "LogIncome")
  synthetic_m[[i]] <- synthetic_one
}
```

Example: synthetic CE sample cont'd

- Goal: valid inferences for the unknown mean of `log(Income)` from $m = 20$ synthetic datasets

Synthetic CE sample: step 1

- Calculate the $q^{(i)}$ and $v^{(i)}$, the point estimate and the variance estimate of the mean of `log(Income)` in each of the $m = 20$ synthetic datasets, and $i = 1, \cdots, m$

```
q <- rep(NA, m)
v <- rep(NA, m)
for (i in 1:m){
  synthetic_one <- synthetic_m[[i]]
  q[i] <- mean(synthetic_one$LogIncome)
  v[i] <- var(synthetic_one$LogIncome)/n
}
```

Synthetic CE sample: step 2

- Calculate $\bar{q}_m$, b_m, and $\bar{v}_m$

```
q_bar_m <- mean(q)
b_m <- var(q)
v_bar_m <- mean(v)
```

Synthetic CE sample: step 3

- Calculate $T_p = \frac{b_m}{m} + \bar{v}_m$ as the variance estimate of $\bar{q}_m$
- Calculate $v_p = (m-1)\left(1 + \frac{\bar{v}_m}{b_m/m}\right)^2$ as the degrees of freedom of the t distribution

```
T_p <- b_m / m + v_bar_m
v_p <- (m - 1) * (1 + v_bar_m / (b_m / m))^2
```

Synthetic CE sample: step 4

- Obtain the point estimate for mean estimand Q, and the 95% confidence interval

```
q_bar_m
```

```
## [1] 10.61162
```

```
t_score_syn <- qt(p = 0.975, df = v_p)
c(q_bar_m - t_score_syn * sqrt(T_p), q_bar_m + t_score_syn * sqrt(T_p))
```

```
## [1] 10.53436 10.68889
```

Synthetic CE sample: step 4-extra

- Synthetic: [10.53, 10.69]
- Obtain the point estimate for mean estimand Q, and the 95% confidence interval from the original data

```
mean_org <- mean(CEdata$LogIncome)
sd_org <- sd(CEdata$LogIncome)
t_score_org <- qt(p = 0.975, df = n - 1)
mean_org
```

```
## [1] 10.59507
```

```
c(mean_org - t_score_org * sd_org / sqrt(n),
  mean_org + t_score_org * sd_org / sqrt(n))
```

```
## [1] 10.52328 10.66687
```

Outline

Combining rules (full synthesis)

- Q: a univariate estimand
  - e.g. a population mean of a univariate outcome or a univariate regression coefficient of a regression analysis
- q and v: the point estimate and variance estimate of Q from the confidential, original data
  - q and v are not available unless one has access to the original data
  - q and v are estimates from a sample, for example, when Q is a population mean, following the Central Limit Theorem
    - ★ $v = \frac{\sigma^2}{n}$ where σ is the population standard deviation if available
    - ★ $v = \frac{s}{n}$ where s is the sample standard deviation

Combining rules (full synthesis) cont'd

- $\mathbf{Z} = (\mathbf{Z}^{(1)}, \cdots, \mathbf{Z}^{(m)})$: the set of m fully synthetic datasets
- $q^{(i)}$ and $v^{(i)}$: the values of q and v in the i-th synthetic dataset, $\mathbf{Z}^{(i)}$
- Calculate:

$$\bar{q}_m = \sum_{i=1}^{m} \frac{q^{(i)}}{m}, \tag{6}$$

$$b_m = \sum_{i=1}^{m} \frac{(q^{(i)} - \bar{q}_m)^2}{m-1}, \tag{7}$$

$$\bar{v}_m = \sum_{i=1}^{m} \frac{v^{(i)}}{m}. \tag{8}$$

Combining rules (full synthesis) cont'd

- Use $\bar{q}_m$ as the point estimate of Q
- Use T_f as the variance estimate of $\bar{q}_m$

$$T_f = \left(1 + \frac{1}{m}\right) b_m - \bar{v}_m \tag{9}$$

- Reiter (2002) suggests an alternative, non-negative variance estimator,

$$T_f^* = \max(0, T_f) + \delta \left(\frac{n_{syn}}{n} \bar{v}_m\right) \tag{10}$$

  - $\delta = 1$ if $T_f < 0$ and $\delta = 0$ otherwise
  - n_{syn} is the number of observations in the released datasets sampled from the synthetic population

Combining rules (full synthesis) cont'd

- When the sample size of the synthetic data n is large, use a t distribution with degrees of freedom $v_f = (m-1)\left(1 - \frac{\bar{v}_m}{(1+\frac{1}{m})b_m}\right)^2$ to make inferences for estimand Q
- Obtain a 95% confidence interval for Q as:

$$
\begin{aligned}
(\bar{q}_m - t_{v_f}(0.975) \quad &\times \quad \sqrt{\left(1+\frac{1}{m}\right) b_m - \bar{v}_m}, \\
\bar{q}_m \quad + \quad t_{v_f}(0.975) \times &\sqrt{\left(1+\frac{1}{m}\right) b_m - \bar{v}_m})
\end{aligned}
\tag{11}
$$

  - $t_{v_f}(0.975)$ is the t score at 0.975 with degrees of freedom v_f

Example: synthetic CE sample

- Previously, we have worked with the CE sample:
  - a Bayesian simple linear regression synthesis model
  - synthesize `log(Income)` given `log(Expenditure)`
  - $m = 20$ synthetic datasets saved in `synthetic_one_m`

Example: synthetic CE sample

- Previously, we have worked with the CE sample:
  - a Bayesian simple linear regression synthesis model
  - synthesize `log(Income)` given `log(Expenditure)`
  - $m = 20$ synthetic datasets saved in `synthetic_one_m`
- Now, we need to synthesize both variables

Example: synthetic CE sample cont'd

- Take the sequential synthesis approach:
  1. synthesize `log(Expenditure)` from a normal synthesis model
  2. synthesize `log(Income)` from the previously developed Bayesian simple linear regression synthesis model
- Step 2 is readily available from the previous example
- To develop Step 1

Example: synthetic CE sample cont'd

- Use JAGS to estimate the normal synthesis model for `log(Expenditure)`

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

## priors
mu ~ dnorm(mu0, invtau2)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}
"
```

Example: synthetic CE sample cont'd

- Use JAGS to estimate the normal synthesis model for `log(Expenditure)`

```
y <- as.vector(CEdata$LogExpenditure)
N <- length(y)
the_data <- list("y" = y, "N" = N,
                 "mu0" = 0, "invtau2" = 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))
}
```

Example: synthetic CE sample cont'd

- Use JAGS to estimate the normal synthesis model for `log(Expenditure)`

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

Example: synthetic CE sample cont'd

- Use JAGS to estimate the normal synthesis model for `log(Expenditure)`

```
post_logexp <- as.mcmc(posterior_logexp)

synthesize_logexp <- function(index, n, seed){
  set.seed(seed)
  synthetic_Y <- rnorm(n, post_logexp[index, "mu"],
                          post_logexp[index, "sigma"])
}
```

Example: synthetic CE sample cont'd

- Synthesize two variables in a sequential manner

```
n <- dim(CEdata)[1]
m <- 20
synthetic_m <- vector("list", m)
for (i in 1:m){
  set.seed(123)
  seed_1 <- round(runif(1, 1, 1000))
  synthetic_logexp <- as.vector(synthesize_logexp(4980 + i,
                                                  n, seed_1))
  seed_2 <- round(runif(1, 1, 1000))
  synthetic_one <- synthesize_loginc(synthetic_logexp, 4980 + i,
                                     n, seed_2)
  names(synthetic_one) <- c("LogExpenditure", "LogIncome")
  synthetic_m[[i]] <- synthetic_one
}
```

Example: synthetic CE sample cont'd

- $m = 20$ synthetic datasets are saved in the list `synthetic_m`
- each dataset is generated using one of the last 20 independent MCMC iteration from the two sets of obtained 5000 MCMC samples
- any pairing of the two sets of 20 independent MCMC draws is okay
  - e.g. A_1^*, A_2^*, A_3^*, then one can do $B_1^* \mid A_2^*, B_2^* \mid A_3^*, B_3^* \mid A_1^*$

Example: synthetic CE sample cont'd

- Goal: valid inferences for the unknown regression coefficient β_1 from $m = 20$ synthetic datasets

$$LogIncome_i = \beta_0 + \beta_1 LogExpenditure_i \tag{12}$$

Synthetic CE sample: step 1

- Calculate $q^{(i)}$ and $v^{(i)}$, the point estimate and the variance estimate of the regression coefficient β_1 in each of the $m = 20$ synthetic datasets, and $i = 1, \cdots, m$
- Use the `lm()` function to perform the regression analysis
- Create `ComputeBeta1()` function to obtain $q^{(i)}$ and $v^{(i)}$

Synthetic CE sample: step 1 cont'd

```
ComputeBeta1 <- function(m, syndata){
Beta1_q <- rep(NA, m)
Beta1_v <- rep(NA, m)

for (i in 1:m){
  syndata_i <- syndata[[i]]
  syndata_i_lm <- lm(LogIncome ~ LogExpenditure, data = syndata_i)
  coef_output <- coef(summary(syndata_i_lm))
  Beta1_q[i] <- coef_output[2, 1]
  Beta1_v[i] <- coef_output[2, 2]^2
}

res <- list(Beta1_q, Beta1_v)
}
Beta1_qv <- ComputeBeta1(m, synthetic_m)
Beta1_q <- Beta1_qv[[1]]
Beta1_v <- Beta1_qv[[2]]
```

Synthetic CE sample: step 2

- Calculate $\bar{q}_m$, b_m, and $\bar{v}_m$

```
Beta1_q_bar_m <- mean(Beta1_q)
Beta1_b_m <- var(Beta1_q)
Beta1_v_bar_m <- mean(Beta1_v)
```

Synthetic CE sample: step 3

- Calculate $T_f = (1 + \frac{1}{m})b_m - \bar{v}_m$ as the variance estimate of $\bar{q}_m$

```
Beta1_T_f <- (1 + 1 / m) * Beta1_b_m - Beta1_v_bar_m
Beta1_T_f
```

```
## [1] -0.0003010283
```

Synthetic CE sample: step 3

- Calculate $T_f = (1 + \frac{1}{m})b_m - \bar{v}_m$ as the variance estimate of $\bar{q}_m$

```
Beta1_T_f <- (1 + 1 / m) * Beta1_b_m - Beta1_v_bar_m
Beta1_T_f
```

```
## [1] -0.0003010283
```

- If T_f is negative, we can use the alternative, non-negative variance estimator $T_f^* = \max(0, T_f) + \delta\left(\frac{n_{syn}}{n}\bar{v}_m\right)$
  - here $n_{syn} = n$, therefore $\frac{n_{syn}}{n} = 1$
  - $\delta = 1$ since $T_f < 0$

```
Beta1_T_f_new <- min(0, Beta1_T_f) + 1 * (1 * Beta1_v_bar_m)
Beta1_T_f_new
```

```
## [1] 0.000834984
```

Synthetic CE sample: step 3 cont'd

- Calculate $v_f = (m-1)\left(1 - \frac{\bar{v}_m}{\left(1+\frac{1}{m}\right)b_m}\right)^2$ as the degrees of freedom of the t distribution

```
Beta1_v_f <- (m - 1) * (1 - Beta1_v_bar_m / ((1 + 1 / m) * Beta1_b_m))^2
```

Synthetic CE sample: step 4

- Obtain the point estimate for regression coefficient β_1, and the 95% confidence interval

```
Beta1_q_bar_m
```

```
## [1] 0.7077417
```

```
Beta1_t_score_syn <- qt(p = 0.975, df = Beta1_v_f)
c(Beta1_q_bar_m - Beta1_t_score_syn * sqrt(Beta1_T_f_new),
  Beta1_q_bar_m + Beta1_t_score_syn * sqrt(Beta1_T_f_new))
```

```
## [1] 0.6035302 0.8119533
```

Synthetic CE sample: step 4-extra

- Synthetic: [0.60, 0.81]
- Obtain the point estimate of the unknown regression coefficient β_1, its standard error, and its t value from the output from the `lm()` on the original data

```
orgdata_lm <- lm(LogIncome ~ LogExpenditure, data = CEdata)
coef(summary(orgdata_lm))
```

```
##                 Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)    4.1112300 0.30801033 13.34770 1.697568e-37
## LogExpenditure 0.7381404 0.03489378 21.15392 2.849973e-82
```

Synthetic CE sample: step 4-extra cont'd

- Obtain the 95% confidence interval

```
Beta1_mean_org <- 0.738
Beta1_se_org <- 0.035
Beta1_t_score_org <- qt(p = 0.975, df = 21.15)
c(Beta1_mean_org - Beta1_t_score_org * Beta1_se_org,
  Beta1_mean_org + Beta1_t_score_org * Beta1_se_org)
```

```
## [1] 0.6652449 0.8107551
```

Outline

Overview

- Combining rules provide point estimate and confidence interval estimates
- We can also obtain point estimate and confidence interval estimate from the original, confidential data
- Naturally, we can evaluate how close the two confidence intervals are
  - one from the synthetic datasets
  - one from the original dataset

Interval overlap utility measure

Drechsler and Reiter (2009)

- (L_s, U_s): the 95% confidence interval for the estimand from m synthetic datasets
- (L_o, U_o): the 95% confidence interval for the estimand from the original, confidential data
- (L_i, U_i): the intersection of the two intervals
  - i.e. $(\max(L_s, L_o), \min(U_s, U_o))$

Interval overlap utility measure cont'd

- The utility measure is

$$I = \frac{U_i - L_i}{2(U_o - L_o)} + \frac{U_i - L_i}{2(U_s - L_s)} \tag{13}$$

  - nearly identical intervals indicate high utility, and result in $I \approx 1$
  - intervals with little overlap indicate low utility, and result in $I \approx 0$
- When multiple estimands are considered, we can average the values of I over all estimands to obtain a summary

Example: synthetic CE sample - partial

- mean estimate of `log(Income)`
- synthetic: [10.53, 10.69]
- origina: [10.52, 10.67]

```
L_s <- 10.53
U_s <- 10.69
L_o <- 10.52
U_o <- 10.67
L_i <- 10.53
U_i <- 10.67

I <- (U_i - L_i) / (2 * (U_o - L_o)) + (U_i - L_i) / (2 * (U_s - L_s))
I
```

```
## [1] 0.9041667
```

Example: synthetic CE sample - full

- regression coefficient estimate of β_1
- synthetic: [0.60, 0.81]
- original: [0.67, 0.81]

```
L_s <- 0.60
U_s <- 0.81
L_o <- 0.67
U_o <- 0.81
L_i <- 0.67
U_i <- 0.81

I <- (U_i - L_i) / (2 * (U_o - L_o)) + (U_i - L_i) / (2 * (U_s - L_s))
I
```

```
## [1] 0.8333333
```

Example: synthetic CE sample - conclusion

- Partial case: $I = 0.904$, close to 1, high utility
- Full case: $I = 0.833$, not as close to 1, reasonably high utility

Outline

Additional utility measures

- Point estimates and / or confidence intervals are not analytically available
  - e.g. median

Additional utility measures

- Point estimates and / or confidence intervals are not analytically available
  - e.g. median
- Solution: bootstrap

Values of m

- Effects of m on data utility

References

- Drechsler, J. and Hu, J. (2019+), Synthesizing geocodes to facilitate access to detailed geographical information in large scale administrative data. arXiv: 1803.05874.
- Reiter, J. (2002) Satisfying disclosure restrictions with synthetic data sets, Journal of Official Statistics, 531-544.
- Drechsler, J. and Reiter, J. P. (2009). Disclosure Risk and Data Utility for Partially Synthetic Data: An Empirical Study Using the German IAB. Journal of Official Statistics, pp. 589-603.