Crossfitting-SRE

Lei Shi

2024-11-26

Setting 3: SRE setting

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
set.seed(2024)

Nk = c(200, 400, 200)
Nk1 = c(100, 200, 100)
Nk0 = Nk - Nk1
N = sum(Nk)

# generate data
X1 = rnorm(N)
X2 = rnorm(N)

full_data = data.frame(
  Y1 = c(rpois(Nk[1], lambda = exp(0.2 * X1 + 0.2 * X2)),
         rpois(Nk[2], lambda = exp(0.4 * X1 + 0.4 * X2)),
         rpois(Nk[3], lambda = exp(0.6 * X1 + 0.6 * X2))),
  Y0 = c(rpois(Nk[1], lambda = exp(0.1 * X1 + 0.1 * X2)),
         rpois(Nk[2], lambda = exp(0.2 * X1 + 0.2 * X2)),
         rpois(Nk[3], lambda = exp(0.3 * X1 + 0.3 * X2))),
  X1 = X1,
  X2 = X2,
```

```
  strata = rep(c(1,2,3), times = Nk)
)

tau = mean(full_data$Y1 - full_data$Y0)

MC = 1e3
result_correct_model = data.frame(
  tau = rep(tau, MC),
  tau_hat_cf = rep(0, MC),
  var_hat_cf = rep(0, MC),
  tau_hat = rep(0, MC),
  var_hat = rep(0, MC)
)

result_wrong_model = data.frame(
  tau = rep(tau, MC),
  tau_hat_cf = rep(0, MC),
  var_hat_cf = rep(0, MC),
  tau_hat = rep(0, MC),
  var_hat = rep(0, MC)
)
```

correct model setting

```
# treatment assignment
set.seed(2024)
pb = txtProgressBar(min = 0, max = MC, initial = 0, style = 3)
```

```
##   |                                                                      |
```

```
for (iter in 1:MC){
  setTxtProgressBar(pb, iter)
  Z = c(sample(rep(c(1,0), times = c(Nk1[1], Nk0[1]))),
        sample(rep(c(1,0), times = c(Nk1[2], Nk0[2]))),
        sample(rep(c(1,0), times = c(Nk1[3], Nk0[3]))))

  full_data$Y = full_data$Y1 * Z + full_data$Y0 * (1-Z)
  full_data$Z = Z

  # naive estimator
  tau_hat = (full_data %>% group_by(strata, Z) %>%
               summarize(Yhat = mean(Y), num = n()) %>%
               mutate(Yhat = ifelse(Z == 1, Yhat, -Yhat)) %>%
               group_by(strata) %>%
               summarize(strata_est = sum(Yhat), strata_num = sum(num)) %>%
               ungroup() %>%
               summarize(tau_hat = sum(strata_est * strata_num)/sum(strata_num)))$tau_hat[1]

  var_hat = (full_data %>%
               group_by(strata, Z) %>%
               summarize(vhat = var(Y), num = n()) %>%
               mutate(vhat_over_n = vhat/num) %>%
               group_by(strata) %>%
               summarize(strata_var_hat = sum(vhat_over_n), strata_num = sum(num)) %>%
```

```
            ungroup() %>%
            summarize(var_hat = sum(strata_var_hat * strata_num^2)/sum(strata_num)^2))$var_hat[1]


# split by treatment
Nka = c(50, 100, 50)
Nkb = Nk - Nka
Nk1a = c(25, 50, 25)
Nk1b = Nk1 - Nk1a
Nk0a = Nka - Nk1a
Nk0b = Nkb - Nk1b
Na = sum(Nka)
Nb = sum(Nkb)

full_data$S = rep(NA, N)

S1 = c(sample(rep(c("a", "b"), times = c(Nk1a[1], Nk1b[1]))),
       sample(rep(c("a", "b"), times = c(Nk1a[2], Nk1b[2]))),
       sample(rep(c("a", "b"), times = c(Nk1a[3], Nk1b[3]))))

S0 = c(sample(rep(c("a", "b"), times = c(Nk0a[1], Nk0b[1]))),
       sample(rep(c("a", "b"), times = c(Nk0a[2], Nk0b[2]))),
       sample(rep(c("a", "b"), times = c(Nk0a[3], Nk0b[3]))))

full_data$S = rep(NA, N)
full_data$S[Z == 1] = S1
full_data$S[Z == 0] = S0

# fitting GLM models
glm_fit_1a = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1+X2",
                          data = full_data %>% filter(strata == x & S == "a" & Z == 1),
                          family = "poisson")
                    })

glm_fit_1b = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1+X2",
                          data = full_data %>% filter(strata == x & S == "b" & Z == 1),
                          family = "poisson")
                    })

glm_fit_0a = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1+X2",
                          data = full_data %>% filter(strata == x & S == "a" & Z == 0),
                          family = "poisson")
                    })

glm_fit_0b = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1+X2",
```

```
                        data = full_data %>% filter(strata == x & S == "b" & Z == 0),
                        family = "poisson")
                    })

# get predictions
pred1_on_a = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_1b[[x]],
                              newdata = full_data %>% filter(strata == x & S == "a"),
                              type = "response")
                    })
pred1_on_b = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_1a[[x]],
                              newdata = full_data %>% filter(strata == x & S == "b"),
                              type = "response")
                    })
pred0_on_a = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_0b[[x]],
                              newdata = full_data %>% filter(strata == x & S == "a"),
                              type = "response")
                    })
pred0_on_b = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_0a[[x]],
                              newdata = full_data %>% filter(strata == x & S == "b"),
                              type = "response")
                    })

pred1 = rep(0, N)
pred1[full_data$S == "a"] = do.call(c, pred1_on_a)
pred1[full_data$S == "b"] = do.call(c, pred1_on_b)

pred0 = rep(0, N)
pred0[full_data$S == "a"] = do.call(c, pred0_on_a)
pred0[full_data$S == "b"] = do.call(c, pred0_on_b)

full_data$pred1 = pred1
full_data$pred0 = pred0

full_data$epsilon = 0
full_data$epsilon[Z == 1] = full_data$Y[Z == 1] - full_data$pred1[Z == 1]
full_data$epsilon[Z == 0] = full_data$Y[Z == 0] - full_data$pred0[Z == 0]

residual_est_by_strata = full_data %>%
  group_by(S, strata, Z) %>%
  summarize(avg_epsilon = mean(epsilon)) %>%
  mutate(avg_epsilon = ifelse(Z == 1, avg_epsilon, -avg_epsilon)) %>%
  group_by(S, strata) %>%
  summarize(est_by_strata = sum(avg_epsilon))

mean_model_diff_by_strata = full_data %>%
```

```
    group_by(S, strata) %>%
    summarize(model_diff = mean(pred1 - pred0))

  tau_hat_cf = sum((residual_est_by_strata$est_by_strata + mean_model_diff_by_strata$model_diff) * c(Nka

  var_by_strata = full_data %>%
    group_by(S, strata, Z) %>%
    summarize(var_eps = var(epsilon), num = n()) %>%
    mutate(var_eps_over_num = var_eps/num) %>%
    group_by(S, strata) %>%
    summarize(var_sum = sum(var_eps_over_num))

  var_hat_cf = sum(var_by_strata$var_sum * c(Nka^2, Nkb^2)/N^2)

  result_correct_model$tau_hat_cf[iter] = tau_hat_cf
  result_correct_model$var_hat_cf[iter] = var_hat_cf
  result_correct_model$tau_hat[iter] = tau_hat
  result_correct_model$var_hat[iter] = var_hat
}
```

```
##   |                                                                              |
```

```
print(tau)
```

```
## [1] 0.19875
```

```
print(mean(result_correct_model$tau_hat))
```

```
## [1] 0.1964325
```

```
print(var(result_correct_model$tau_hat))
```

```
## [1] 0.004804384
```

```
print(mean(result_correct_model$var_hat))
```

```
## [1] 0.007564406
```

```
print(mean(result_correct_model$tau_hat_cf))
```

```
## [1] 0.1971591
```

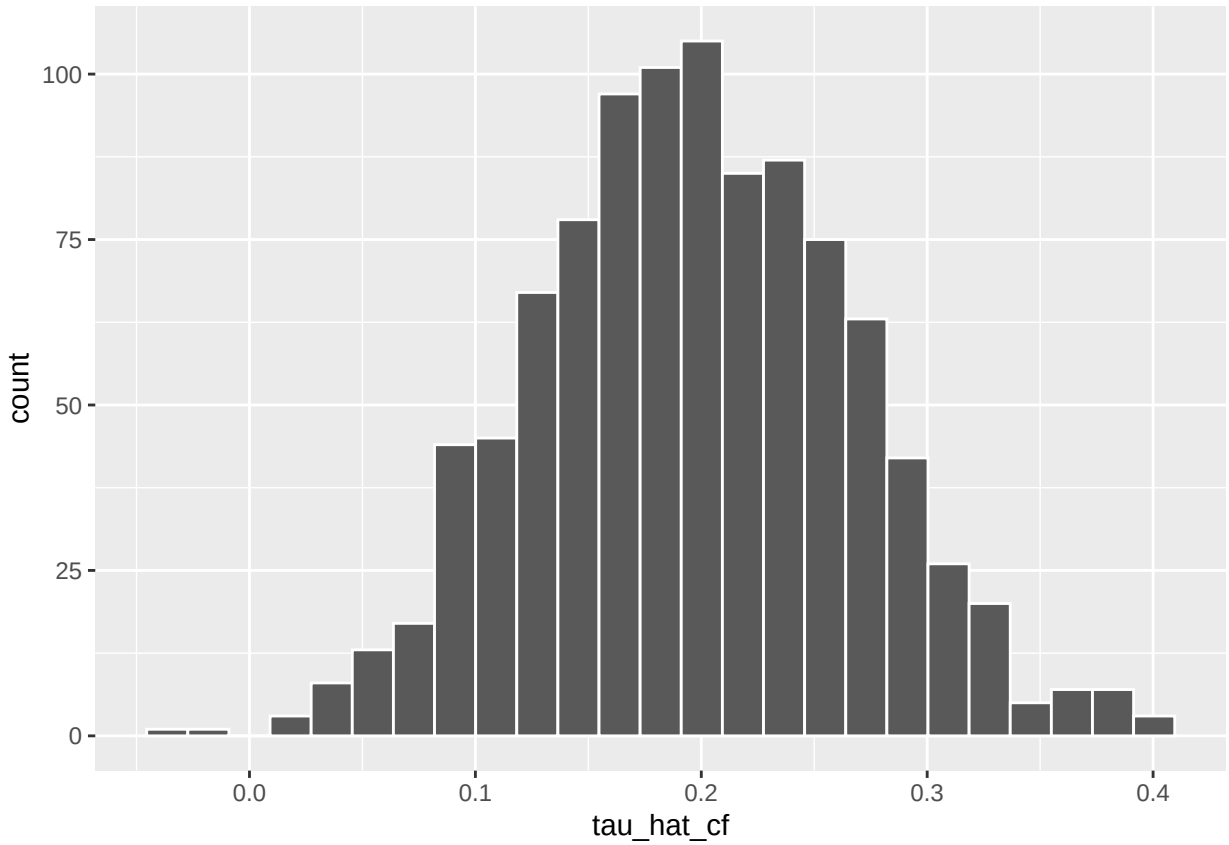
```
print(var(result_correct_model$tau_hat_cf))
```

```
## [1] 0.004915314
```

```
print(mean(result_correct_model$var_hat_cf))
```

```
## [1] 0.008063326
```

```
result_correct_model %>% ggplot() +
  geom_histogram(aes(x = tau_hat_cf), col = "white", bins = 25)
```



wrong model setting

```
# treatment assignment
set.seed(2024)
pb = txtProgressBar(min = 0, max = MC, initial = 0, style = 3)
```

```
##   |                                                                      |
```

```
for (iter in 1:MC){
  setTxtProgressBar(pb, iter)
  Z = c(sample(rep(c(1,0), times = c(Nk1[1], Nk0[1]))),
        sample(rep(c(1,0), times = c(Nk1[2], Nk0[2]))),
        sample(rep(c(1,0), times = c(Nk1[3], Nk0[3]))))

  full_data$Y = full_data$Y1 * Z + full_data$Y0 * (1-Z)
  full_data$Z = Z

  # naive estimator
  tau_hat = (full_data %>% group_by(strata, Z) %>%
               summarize(Yhat = mean(Y), num = n()) %>%
               mutate(Yhat = ifelse(Z == 1, Yhat, -Yhat)) %>%
               group_by(strata) %>%
               summarize(strata_est = sum(Yhat), strata_num = sum(num)) %>%
               ungroup() %>%
               summarize(tau_hat = sum(strata_est * strata_num)/sum(strata_num)))$tau_hat[1]

  var_hat = (full_data %>%
```

```
              group_by(strata, Z) %>%
              summarize(vhat = var(Y), num = n()) %>%
              mutate(vhat_over_n = vhat/num) %>%
              group_by(strata) %>%
              summarize(strata_var_hat = sum(vhat_over_n), strata_num = sum(num)) %>%
              ungroup() %>%
              summarize(var_hat = sum(strata_var_hat * strata_num^2)/sum(strata_num)^2))$var_hat[1]

# split by treatment
Nka = c(50, 100, 50)
Nkb = Nk - Nka
Nk1a = c(25, 50, 25)
Nk1b = Nk1 - Nk1a
Nk0a = Nka - Nk1a
Nk0b = Nkb - Nk1b
Na = sum(Nka)
Nb = sum(Nkb)

full_data$S = rep(NA, N)

S1 = c(sample(rep(c("a", "b"), times = c(Nk1a[1], Nk1b[1]))),
       sample(rep(c("a", "b"), times = c(Nk1a[2], Nk1b[2]))),
       sample(rep(c("a", "b"), times = c(Nk1a[3], Nk1b[3]))))

S0 = c(sample(rep(c("a", "b"), times = c(Nk0a[1], Nk0b[1]))),
       sample(rep(c("a", "b"), times = c(Nk0a[2], Nk0b[2]))),
       sample(rep(c("a", "b"), times = c(Nk0a[3], Nk0b[3]))))

full_data$S = rep(NA, N)
full_data$S[Z == 1] = S1
full_data$S[Z == 0] = S0

# fitting GLM models
glm_fit_1a = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1",
                          data = full_data %>% filter(strata == x & S == "a" & Z == 1),
                          family = "poisson")
                    })

glm_fit_1b = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1",
                          data = full_data %>% filter(strata == x & S == "b" & Z == 1),
                          family = "poisson")
                    })

glm_fit_0a = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1",
                          data = full_data %>% filter(strata == x & S == "a" & Z == 0),
                          family = "poisson")
                    })
```

```r
glm_fit_0b = lapply(c(1,2,3),
                    function(x){
                      glm("Y~X1",
                          data = full_data %>% filter(strata == x & S == "b" & Z == 0),
                          family = "poisson")
                    })

# get predictions
pred1_on_a = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_1b[[x]],
                              newdata = full_data %>% filter(strata == x & S == "a"),
                              type = "response")
                    })
pred1_on_b = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_1a[[x]],
                              newdata = full_data %>% filter(strata == x & S == "b"),
                              type = "response")
                    })
pred0_on_a = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_0b[[x]],
                              newdata = full_data %>% filter(strata == x & S == "a"),
                              type = "response")
                    })
pred0_on_b = lapply(c(1,2,3),
                    function(x){
                      predict(glm_fit_0a[[x]],
                              newdata = full_data %>% filter(strata == x & S == "b"),
                              type = "response")
                    })

pred1 = rep(0, N)
pred1[full_data$S == "a"] = do.call(c, pred1_on_a)
pred1[full_data$S == "b"] = do.call(c, pred1_on_b)

pred0 = rep(0, N)
pred0[full_data$S == "a"] = do.call(c, pred0_on_a)
pred0[full_data$S == "b"] = do.call(c, pred0_on_b)

full_data$pred1 = pred1
full_data$pred0 = pred0

full_data$epsilon = 0
full_data$epsilon[Z == 1] = full_data$Y[Z == 1] - full_data$pred1[Z == 1]
full_data$epsilon[Z == 0] = full_data$Y[Z == 0] - full_data$pred0[Z == 0]

residual_est_by_strata = full_data %>%
  group_by(S, strata, Z) %>%
  summarize(avg_epsilon = mean(epsilon)) %>%
  mutate(avg_epsilon = ifelse(Z == 1, avg_epsilon, -avg_epsilon)) %>%
  group_by(S, strata) %>%
```

```
    summarize(est_by_strata = sum(avg_epsilon))

  mean_model_diff_by_strata = full_data %>%
    group_by(S, strata) %>%
    summarize(model_diff = mean(pred1 - pred0))

  tau_hat_cf = sum((residual_est_by_strata$est_by_strata + mean_model_diff_by_strata$model_diff) * c(Nka

  var_by_strata = full_data %>%
    group_by(S, strata, Z) %>%
    summarize(var_eps = var(epsilon), num = n()) %>%
    mutate(var_eps_over_num = var_eps/num) %>%
    group_by(S, strata) %>%
    summarize(var_sum = sum(var_eps_over_num))

  var_hat_cf = sum(var_by_strata$var_sum * c(Nka^2, Nkb^2)/N^2)

  result_wrong_model$tau_hat_cf[iter] = tau_hat_cf
  result_wrong_model$var_hat_cf[iter] = var_hat_cf
  result_wrong_model$tau_hat[iter] = tau_hat
  result_wrong_model$var_hat[iter] = var_hat
}
```

```
##   |                                                                      |
```

```
print(tau)
```

```
## [1] 0.19875
```

```
print(mean(result_wrong_model$tau_hat))
```

```
## [1] 0.1964325
```

```
print(var(result_wrong_model$tau_hat))
```

```
## [1] 0.004804384
```

```
print(mean(result_wrong_model$var_hat))
```

```
## [1] 0.007564406
```

```
print(mean(result_wrong_model$tau_hat_cf))
```

```
## [1] 0.1973323
```

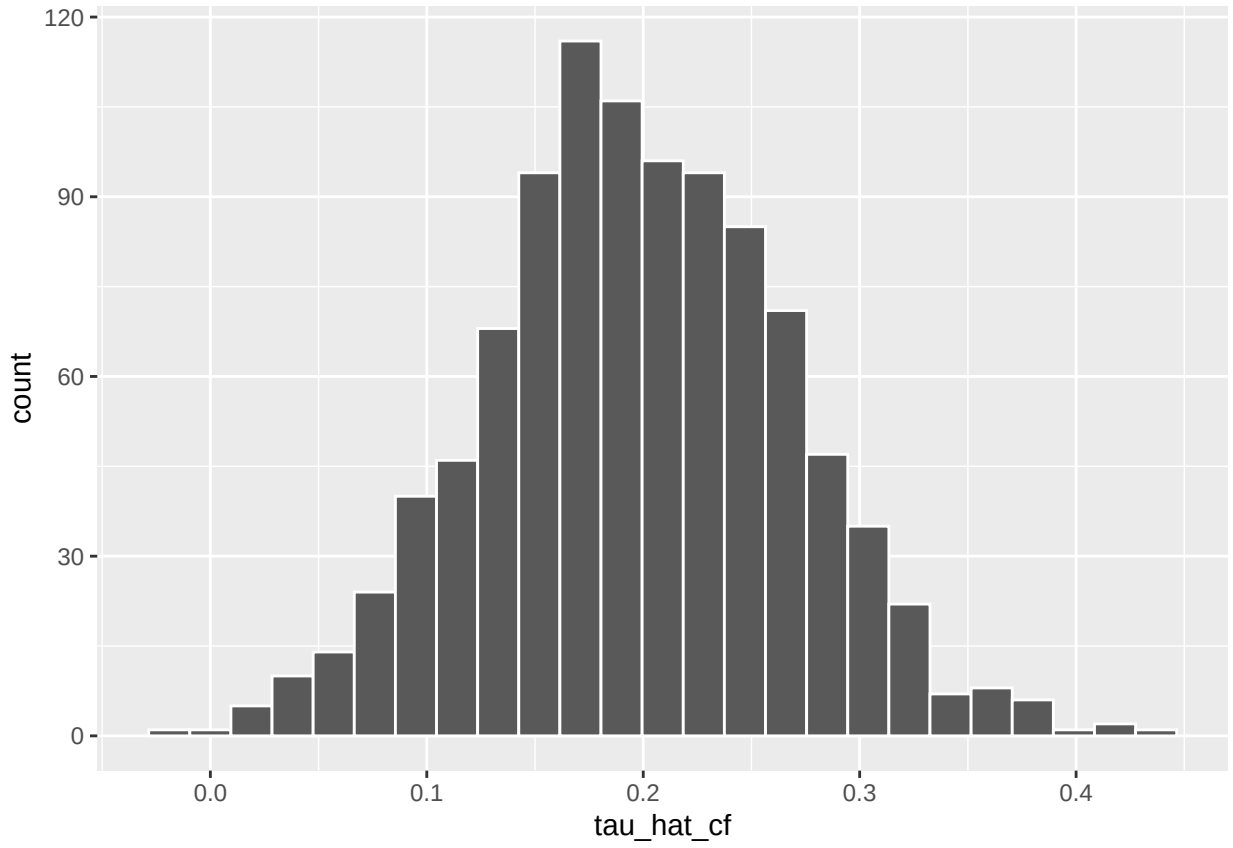
```
print(var(result_wrong_model$tau_hat_cf))
```

```
## [1] 0.004912635
```

```
print(mean(result_wrong_model$var_hat_cf))
```

```
## [1] 0.007813498
```

```
result_wrong_model %>% ggplot() +
  geom_histogram(aes(x = tau_hat_cf), col = "white", bins = 25)
```


120
90
60
30
0
count
0.0
0.1
0.2
0.3
0.4
tau_hat_cf