Lecture 20: Individual-Specific Parameters II

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last time

- Individual-Specific Parameters

Today

- Individual-Specific Parameters Example in R

Upcoming

- Reading for next time
  - Train textbook, Chapter 7.7
- Problem sets
  - Problem Set 4 is posted, due November 21

Individual-Specific Parameters

We can combine the unconditional (or population) distribution of coefficients, $f(\beta \mid \theta)$, and an individual's choices, y_n, and data, x_n, to generate the conditional distribution of coefficients, $h(\beta \mid y_n, x_n, \theta)$

$$h(\beta \mid y_n, x_n, \theta) = \frac{P(y_n \mid x_n, \beta) \times f(\beta \mid \theta)}{P(y_n \mid x_n, \theta)}$$

If we only want the mean of this distribution, we can simulate the conditional mean coefficients as

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid x_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid x_n, \beta^r)}$$

With enough observed choices for an individual ($T > 50$?), these means converge to the individual's coefficients

Simulation-Based Estimation Example in R

Individual-Specific Parameters Example

We are again studying how consumers make choices about expensive and highly energy-consuming systems in their homes. We have data on 250 households in California and the type of HVAC (heating, ventilation, and air conditioning) system in their home. Each household has the following choice set, and we observe the following data

Choice set

- GCC: gas central with AC
- ECC: electric central with AC
- ERC: electric room with AC
- HPC: heat pump with AC
- GC: gas central
- EC: electric central
- ER: electric room

Alternative-specific data

- ICH: installation cost for heat
- ICCA: installation cost for AC
- OCH: operating cost for heat
- OCCA: operating cost for AC

Household demographic data

- income: annual income

Load Dataset

```r
### Load and look at dataset
## Load tidyverse and mlogit
library(tidyverse)
library(mlogit)
## Load dataset from mlogit package
data('HC', package = 'mlogit')
```

Dataset

```
## Look at dataset
as_tibble(HC)
## # A tibble: 250 x 18
##    depvar ich.gcc ich.ecc ich.erc ich.hpc ich.gc ich.ec ich.er  icca
##    <fct>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <dbl>
##  1 erc       9.7     7.86    8.79   11.4    24.1   24.5   7.37  27.3
##  2 hpc       8.77    8.69    7.09    9.37   28     32.7   9.33  26.5
##  3 gcc       7.43    8.86    6.94   11.7    25.7   31.7   8.14  22.6
##  4 gcc       9.18    8.93    7.22   12.1    29.7   26.7   8.04  25.3
##  5 gcc       8.05    7.02    8.44   10.5    23.9   28.4   7.15  25.4
##  6 gcc       9.32    8.03    6.22   12.6    27.0   21.4   8.6   19.9
##  7 gc        7.11    8.78    7.36   12.4    22.9   28.6   6.41  27.0
##  8 hpc       9.38    7.48    6.72    8.93   26.2   27.9   7.3   18.1
##  9 gcc       8.08    7.39    8.79   11.2    23.0   22.6   7.85  22.6
## 10 gcc       6.24    4.88    7.46    8.28   19.8   27.5   6.88  25.8
## # ... with 240 more rows, and 9 more variables: och.gcc <dbl>,
## #   och.ecc <dbl>, och.erc <dbl>, och.hpc <dbl>, och.gc <dbl>,
## #   och.ec <dbl>, och.er <dbl>, occa <dbl>, income <dbl>
```

Format Dataset in a Long Format

```
### Format dataset
## Gather into a long dataset
hvac_long <- HC %>%
  mutate(id = 1:n()) %>%
  gather(key, value, starts_with('ich.'), starts_with('och.')) %>%
  separate(key, c('cost', 'alt')) %>%
  spread(cost, value) %>%
  mutate(choice = 1 * (depvar == alt)) %>%
  select(-depvar)
```

Dataset in a Long Format

```
## Look at long dataset
as_tibble(hvac_long)
## # A tibble: 1,750 x 8
##     icca  occa income    id alt     ich   och choice
##    <dbl> <dbl>  <dbl> <int> <chr> <dbl> <dbl>  <dbl>
##  1  17    2.79     60   133 ec    20.3   4.52      0
##  2  17    2.79     60   133 ecc    8.46  4.52      0
##  3  17    2.79     60   133 er     7.7   4.32      0
##  4  17    2.79     60   133 erc    8.16  4.32      0
##  5  17    2.79     60   133 gc    25.3   2.26      0
##  6  17    2.79     60   133 gcc    6.33  2.26      1
##  7  17    2.79     60   133 hpc   11.1   1.63      0
##  8  18.1  2.55     50    14 ec    25.6   5.21      0
##  9  18.1  2.55     50    14 ecc   11.2   5.21      0
## 10  18.1  2.55     50    14 er     9.3   3.8       0
## # ... with 1,740 more rows
```

Clean Dataset

```
## Combine heating and cooling costs into one variable
hvac_clean <- hvac_long %>%
  mutate(cooling = (nchar(alt) == 3),
         ic = if_else(cooling, ich + icca, ich),
         oc = if_else(cooling, och + occa, och)) %>%
  mutate(cooling = 1 * cooling) %>%
  select(id, alt, choice, cooling, ic, oc, income) %>%
  arrange(id, alt)
```

Cleaned Dataset

```
## Look at cleaned dataset
as_tibble(hvac_clean)
## # A tibble: 1,750 x 7
##       id alt   choice cooling    ic    oc income
##    <int> <chr>  <dbl>   <dbl> <dbl> <dbl>  <dbl>
##  1     1 ec         0       0 24.5   4.09     20
##  2     1 ecc        0       1 35.1   7.04     20
##  3     1 er         0       0  7.37  3.85     20
##  4     1 erc        1       1 36.1   6.8      20
##  5     1 gc         0       0 24.1   2.26     20
##  6     1 gcc        0       1 37.0   5.21     20
##  7     1 hpc        0       1 38.6   4.68     20
##  8     2 ec         0       0 32.7   2.69     50
##  9     2 ecc        0       1 35.2   4.32     50
## 10     2 er         0       0  9.33  3.45     50
## # ... with 1,740 more rows
```

Convert Dataset to mlogit Format

```r
## Convert cleaned dataset to mlogit format
hvac_mlogit <- mlogit.data(hvac_clean, shape = 'long',
                           choice = 'choice', alt.var = 'alt')
```

Dataset in mlogit Format

```
## Look at data in mlogit format
as_tibble(hvac_mlogit)
## # A tibble: 1,750 x 7
##          id alt   choice cooling    ic    oc income
##       <int> <fct> <lgl>    <dbl> <dbl> <dbl>  <dbl>
##  1      1 ec    FALSE          0 24.5   4.09     20
##  2      1 ecc   FALSE          1 35.1   7.04     20
##  3      1 er    FALSE          0  7.37  3.85     20
##  4      1 erc   TRUE           1 36.1   6.8      20
##  5      1 gc    FALSE          0 24.1   2.26     20
##  6      1 gcc   FALSE          1 37.0   5.21     20
##  7      1 hpc   FALSE          1 38.6   4.68     20
##  8      2 ec    FALSE          0 32.7   2.69     50
##  9      2 ecc   FALSE          1 35.2   4.32     50
## 10      2 er    FALSE          0  9.33  3.45     50
## # ... with 1,740 more rows
```

Conditional Mean Coefficients to Simulate

The representative utility of each alternative is

$$V_{nj} = \alpha AC_j + \beta_1 IC_{nj} + \beta_2 OC_{nj}$$

with

$$\ln \alpha \sim N(\mu, \sigma^2)$$

and β_1 and β_2 fixed (not random)

For each alternative, what is the mean α coefficient for the households with that HVAC system?

1. Estimate the mixed logit model
2. Simulate $\check{\alpha}_n$ for each household
3. Average $\check{\alpha}_n$ for the households with each HVAC system

Simulating Conditional Mean Coefficients

Two ways to simulate conditional mean coefficients for each household

- `mlogit` package
- Code the simulation by hand

The `mlogit` function `fitted(model, type = 'parameters')` simulates the conditional mean coefficients for every individual

- This function returns an $N \times K$ matrix of conditional mean coefficients

We can instead code the simulation by hand

- We may want to simulate additional objects that are not part of the `mlogit` functionality

Mixed Logit Model Using mlogit

```r
### Model HVAC choice as a mixed logit using mlogit
## Model choice using cooling dummy with log-normal distribution and
## fixed coefficients for installation cost and operating cost
model_1 <- hvac_mlogit %>%
  mlogit(formula = choice ~ cooling + ic + oc | 0 | 0, data = .,
         rpar = c(cooling = 'ln'), R = 100, seed = 703)
```

Model Results Using mlogit

```
## Summarize model results
model_1 %>%
  summary()
##
## Call:
## mlogit(formula = choice ~ cooling + ic + oc | 0 | 0, data = .,
##     rpar = c(cooling = "ln"), R = 100, seed = 703)
##
## Frequencies of alternatives:
##    ec   ecc    er   erc    gc   gcc   hpc
## 0.004 0.016 0.032 0.004 0.096 0.744 0.104
##
## bfgs method
## 14 iterations, 0h:0m:3s
## g'(-H)^-1g = 4.05E-08
## gradient close to zero
##
## Coefficients :
##             Estimate Std. Error z-value  Pr(>|z|)
## cooling     2.561869   0.636702  4.0237 5.730e-05 ***
## ic         -0.213818   0.035361 -6.0468 1.478e-09 ***
## oc         -1.121642   0.186787 -6.0049 1.914e-09 ***
## sd.cooling  0.766363   0.644710  1.1887    0.2346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -327.57
##
## random coefficients
##          Min.  1st Qu.   Median     Mean  3rd Qu. Max.
## cooling     0 7.728894 12.96002 17.38357 21.73173  Inf
```

Conditional Mean Coefficients Using mlogit

```
### Find the mean coefficient for each chosen alternative using mlogit
## Calculate mean coefficient for each household
coefficients_1 <- model_1 %>%
  fitted(type = 'parameters') %>%
  as_tibble() %>%
  rename(cooling_coef = cooling)
coefficients_1
## # A tibble: 250 x 1
##    cooling_coef
##           <dbl>
##  1        19.5
##  2        18.3
##  3        19.3
##  4        18.7
##  5        20.3
##  6        18.9
##  7         4.91
##  8        22.0
##  9        19.9
## 10        18.1
## # ... with 240 more rows
```

Coefficients for Each Alternative Using mlogit

```
## Average coefficient over all households with each HVAC system
hvac_clean %>%
  filter(choice == 1) %>%
  cbind(coefficients_1) %>%
  group_by(alt) %>%
  summarize(cooling_coef = mean(cooling_coef))
## # A tibble: 7 x 2
##   alt   cooling_coef
##   <chr>        <dbl>
## 1 ec            4.38
## 2 ecc          19.5
## 3 er            4.45
## 4 erc          19.5
## 5 gc            4.70
## 6 gcc          19.5
## 7 hpc          19.9
```

Steps for Simulating Conditional Mean Coefficients

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid x_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid x_n, \beta^r)}$$

1. Draw $K \times N \times R$ standard normal random variables
   - K random coefficients for each of
   - N different decision makers for each of
   - R different simulation draws
2. Find the MSLE parameters, $\hat{\theta}$
   - See slides from last week on MSLE
3. Simulate conditional mean coefficients using the MSLE parameters, $\hat{\theta}$
   1. Transform each set of K standard normals using $\hat{\theta}$ to get a set of β_n^r
   2. Calculate the choice probability of the chosen alternative for each individual and draw, $P(y_n \mid x_n, \beta_n^r)$
   3. Take a weighted average of β_n^r, with weights equal to $P(y_n \mid x_n, \beta_n^r)$, to get $\check{\beta}_n$ for each individual

Step 1: Draw Random Variables (and Organize Data)

```r
### Model HVAC choice as a mixed logit coded by hand
## Set seed for replication
set.seed(703)
## Draw standard normal random variables and split into list
draws_2_list <- 1:250 %>%
  map(., ~ tibble(cooling_coef = rnorm(100)))
## Split data into list by household
data_2_list <- hvac_clean %>%
  group_by(id) %>%
  group_split()
```

Step 2a: Simulate Choice Probabilities for One Household

```r
## Function to simulate choice probabilities for one household
simulate_probabilities <- function(parameters, draws, data){
  ## Select relevant variables and convert into a matrix [J * K]
  data_matrix <- data %>%
    select(cooling, ic, oc) %>%
    as.matrix()
  ## Transform random coefficients based on parameters [R * K]
  coefficients <- draws %>%
    mutate(cooling_coef = exp(parameters[1] + parameters[4] * cooling_coef),
           ic_coef = parameters[2],
           oc_coef = parameters[3])
  ## Calculate utility for each alternative in each draw [R * J]
  utility <- (as.matrix(coefficients) %*% t(data_matrix)) %>%
    pmin(700) %>%
    pmax(-700)
  ## Sum the exponential of utility over alternatives [R * 1]
  summed_utility <- utility %>%
    exp() %>%
    rowSums()
  ## Calculate the conditional probability for each alternative in each draw [R * J]
  conditional_probability <- exp(utility) / summed_utility
  ## Average conditional probabilities over all draws [1 * J]
  simulated_probability <- colMeans(conditional_probability)
  ## Add simulated probability to initial dataset
  data_out <- data %>%
    mutate(probability = simulated_probability)
  ## Return initial dataset with simulated probability variable
  return(data_out)
}
```

Step 2b: Calculate Simulated Log-Likelihood

```r
## Function to calculate simulated log-likelihood
simulate_log_likelihood <- function(parameters, draws_list, data_list){
  ## Simulate probabilities for each household
  data <- map2(.x = draws_list, .y = data_list,
               .f = ~ simulate_probabilities(parameters = parameters,
                                             draws = .x,
                                             data = .y))
  ## Combine individual datasets into one
  data <- data %>%
    bind_rows()
  ## Calcule the log of simulated probability for the chosen alternative
  data <- data %>%
    filter(choice == TRUE) %>%
    mutate(log_probability = log(probability))
  ## Calculate the simulated log-likelihood
  simulated_log_likelihood <- sum(data$log_probability)
  ## Return the negative of simulated log-likelihood
  return(-simulated_log_likelihood)
}
```

Step 2c: Maximize Simulated Log-Likelihood

```r
## Maximize the log-likelihood function
model_2 <- optim(par = c(log(6.53), -0.174, -1.04, 0),
                 fn = simulate_log_likelihood,
                 draws_list = draws_2_list, data_list = data_2_list,
                 method = 'BFGS', hessian = TRUE,
                 control = list(trace = 1, REPORT = 5))
## initial  value 329.883895
## iter   5 value 329.769926
## iter  10 value 327.822147
## iter  15 value 327.606084
## final  value 327.573807
## converged
```

Step 2d: Report MSLE Results

```r
## Report parameter estimates and standard errors
model_2$par
## [1]  2.5617903 -0.2137397 -1.1215855  0.7667214

model_2$hessian %>%
  solve() %>%
  diag() %>%
  sqrt()
## [1] 0.6944200 0.0311569 0.1051514 0.6748593
```

Step 3a: Simulate Coefficients for One Household

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid x_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid x_n, \beta^r)}$$

```
### Find the mean coefficient for each chosen alternative coded by hand
## Function to simulate mean coefficient for one household
simulate_coefs <- function(parameters, draws, data){
  ## Select relevant variables and convert into a matrix [J * K]
  data_matrix <- data %>%
    select(cooling, ic, oc) %>%
    as.matrix()
  ## Transform random coefficients based on parameters [R * K]
  coefficients <- draws %>%
    mutate(cooling_coef = exp(parameters[1] + parameters[4] * cooling_coef),
           ic_coef = parameters[2],
           oc_coef = parameters[3])
  ## Calculate utility for each alternative in each draw [R * J]
  utility <- (as.matrix(coefficients) %*% t(data_matrix)) %>%
    pmin(700) %>%
    pmax(-700)
  ## Sum the exponential of utility over alternatives [R * 1]
  summed_utility <- utility %>%
    exp() %>%
    rowSums()
```

Step 3a: Simulate Coefficients for One Household

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid x_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid x_n, \beta^r)}$$

```r
  ## Calculate the conditional probability for each alt in each draw [R * J]
  conditional_probability <- exp(utility) / summed_utility
  ## Extract conditional probabilities of chosen alternative for each draw [R * 1]
  probability_draw <- conditional_probability %*% data$choice
  ## Add draw probability to dataset of coefficients
  coefficients <- coefficients %>%
    mutate(probability = c(probability_draw))
  ## Calculate weighted average for each coefficient
  coefficients_weighted <- coefficients %>%
    summarize(cooling_coef = sum(cooling_coef * probability),
              probability = sum(probability)) %>%
    mutate(cooling_coef = cooling_coef / probability) %>%
    select(-probability)
  ## Add individual coefficients to initial dataset
  data_out <- data %>%
    mutate(cooling_coef = coefficients_weighted$cooling_coef)
  ## Return initial dataset with simulated probability variable
  return(data_out)
}
```

Step 3b: Simulate Coefficients for All Household

$$\check{\beta}_n = \frac{\sum_{r=1}^{R} \beta^r P(y_n \mid x_n, \beta^r)}{\sum_{r=1}^{R} P(y_n \mid x_n, \beta^r)}$$

```
## Calculate individual coefficients for each household
data_2_list <- map2(.x = draws_2_list, .y = data_2_list,
                    .f = ~ simulate_coefs(parameters = model_2$par,
                                          draws = .x,
                                          data = .y))
## Combine list of data into one tibble
data_2 <- data_2_list %>%
  bind_rows()
```

Conditional Mean Coefficients for Each HVAC System

```r
## Calculate average coefficients for Google phone consumers
data_2 %>%
  filter(choice == 1) %>%
  group_by(alt) %>%
  summarize(cooling_coef = mean(cooling_coef))
## # A tibble: 7 x 2
##   alt   cooling_coef
##   <chr>        <dbl>
## 1 ec            4.37
## 2 ecc          19.5
## 3 er            4.45
## 4 erc          19.5
## 5 gc            4.69
## 6 gcc          19.5
## 7 hpc          19.9
```

Announcements

Reading for next time

- Train textbook, Chapter 7.7

Upcoming

- Problem Set 4 is posted, due November 21