Lecture 12: Generalized Method of Moments II

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last time

- Generalized Method of Moments

Today

- Generalized Method of Moments Example in R

Upcoming

- Reading for next time
  - Train textbook, Chapter 4
- Problem sets
  - Problem Set 2 is due
  - Problem Set 3 will be posted soon, due October 31

Generalized Method of Moments

We have population moment conditions

$$E[m(w_i, \theta_0)] = 0$$

The GMM estimator uses the sample analogs of these moment conditions

$$\frac{1}{n}\sum_{i=1}^{n} m(w_i, \theta)$$

The GMM estimator is the set of parameters that solves

$$\frac{1}{n}\sum_{i=1}^{n} m(w_i, \hat{\theta}) = 0$$

The GMM estimator is more flexible and less parametric (semi-parametric?) than MLE or NLS

Generalized Method of Moments Example in R

Binary Logit Model Example

We are studying how consumers make choices about expensive and highly energy-consuming appliances in their homes. We have data on 600 households who rent a studio apartment and whether or not they choose to purchase a window air conditioning unit. For each household, we observe the purchase price of the air conditioner, its annual operating cost, and some household-level data. (To simplify things, we assume there is only one "representative" air conditioner for each household.)

Alternative-specific data

- cost_system: system cost (price)
- cost_operating: annual operating cost

Household data

- elec_price: electricity price
- square_feet: apartment square footage

Load Dataset

```r
### Load and look at dataset
## Load tidyverse
library(tidyverse)
## Load dataset
data <- read_csv('ac_renters.csv')
```

Dataset

```
## Look at dataset
data

## # A tibble: 600 x 8
##    air_conditioning cost_system cost_operating elec_price square_feet
##    <lgl>                  <dbl>          <dbl>      <dbl>       <dbl>
##  1 FALSE                    620            258       13.1         828
##  2 FALSE                    685            141        9.7         553
##  3 FALSE                    570            152       10.1         410
##  4 TRUE                     497            193       11.3         397
##  5 TRUE                     541            162       10.4         395
##  6 FALSE                    663            160       10.3         584
##  7 FALSE                    579            185       11.1         515
##  8 FALSE                    502            158       10.3         333
##  9 TRUE                     562            132        9.4         349
## 10 FALSE                    495            111        8.6         227
## # ... with 590 more rows, and 3 more variables: income <dbl>,
## #   residents <dbl>, city <dbl>
```

Binary Logit Model to Estimate with GMM

The representative utility from having the window air conditioner is (we implicitly normalize the utility of no air conditioner to be zero)

$$V_n = \beta_0 + \beta_1 SC_n + \beta_2 OC_n$$

From the logit model, the probability of purchasing the air conditioner is

$$\begin{aligned} P_n &= \frac{e^{\beta_0 + \beta_1 SC_n + \beta_2 OC_n}}{1 + e^{\beta_0 + \beta_1 SC_n + \beta_2 OC_n}} \\ &= \frac{1}{1 + e^{-(\beta_0 + \beta_1 SC_n + \beta_2 OC_n)}} \end{aligned}$$

We will estimate this model using the moments we derived from the MLE first-order conditions (see previous slides or Train textbook)

$$\begin{aligned} & E[[y_n - P_n(x_n, \beta_0)]x_n] = 0 \\ \Rightarrow \quad & \frac{1}{N}\sum_{n=1}^{N}[y_n - P_n(x_n, \hat{\beta})]x_n = 0 \end{aligned}$$

Generalized Method of Moments in R

```r
### Generalized Method of Moments in R
## Load the gmm package
library(gmm)
```

```r
## Help file for the gmm function
?gmm
## Arguments for gmm function
gmm(g, x, t0, ...)
```

`gmm()` requires that you create a function, g, that

1. Takes a set of parameters and a data *matrix* as inputs
2. Calculates the empirical moments for each observation
3. Returns a $N \times L$ matrix of empirical moments

You also have to give `gmm()` arguments for

- `x`: your data *matrix*
- `t0`: parameter starting values
- Many other optional arguments
  - `gmm()` can be very finicky with the optimization arguments

Calculating Empirical Moments in a Binary Logit Model

Steps to calculate this set of empirical moments for a given set of parameters in a binary logit model

1. Calculate the representative utility of the air conditioner for each decision maker.
2. Calculate the choice probability of the air conditioner for each decision maker.
3. Calculate the econometric residual, or the difference between the outcome and the probability, for each decision maker.
4. Multiply this residual vector $(N \times 1)$ by each column of the explanatory data matrix $(N \times K)$.
5. Return the resulting $N \times K$ matrix of empirical moments.

Binary Logit Model and Moments

The representative utility of the air conditioner is

$$V_n = \beta_0 + \beta_1 SC_n + \beta_2 OC_n$$

The choice probability is

$$P_n = \frac{1}{1 + e^{-(\beta_0 + \beta_1 SC_n + \beta_2 OC_n)}}$$

The empirical moments for each decision maker are

$$[y_n - P_n(x_n, \hat{\beta})]x_n$$

This model is just-identified, so we will actually use the MM estimator

- Three parameters and three moment conditions

Function to Calculate Moments

```
### Calculate GMM estimator for binary logit AC choice using cost data
## Function to calculate binary logit moments
calculate_moments <- function(parameters, data){
  ## Extract explanatory variable data from matrix
  data_x <- data[, -1]
  ## Extract choice data from matrix
  data_y <- data[, 1]
  ## Calculate net utility of alternative given the parameters
  utility <- data_x %*% parameters
  ## Caclculate logit probability of alternative given the parameters
  probability_choice <- 1 / (1 + exp(-utility))
  ## Calculate residuals
  residuals <- data_y - probability_choice
  ## Create moment matrix
  moments <- c(residuals) * data_x
  return(moments)
}
```

Estimate MM Parameters

```r
## Create dataset for use in GMM moment function
data_gmm <- data %>%
  mutate(air_conditioning = 1 * air_conditioning,
         constant = 1) %>%
  select(air_conditioning, constant, cost_system, cost_operating) %>%
  as.matrix()
## Find the GMM estimator
model_gmm <- gmm(calculate_moments, data_gmm, rep(0, 3), vcov = 'iid',
                 control = list(reltol = 1e-25, maxit = 10000))
```

Estimation Results

```r
## Summarize GMM model results
model_gmm %>%
  summary()
```

```
##
## Call:
## gmm(g = calculate_moments, x = data_gmm, t0 = rep(0, 3), vcov = "iid",
##     control = list(reltol = 1e-25, maxit = 10000))
##
##
## Method:  twoStep
##
## Coefficients:
##           Estimate     Std. Error   t value      Pr(>|t|)
## Theta[1]   4.9913e+00   9.3515e-01   5.3374e+00   9.4292e-08
## Theta[2]  -5.4848e-03   1.4633e-03  -3.7483e+00   1.7801e-04
## Theta[3]  -1.0045e-02   2.1869e-03  -4.5934e+00   4.3609e-06
##
## J-Test: degrees of freedom is 0
##                 J-test                P-value
## Test E(g)=0:    1.57492779993196e-20  *******
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  1036
## Gradian eval. =  NA
```

Binary Logit Model with Instruments

We may be concerned that the annual operating cost is endogenous

$$E[OC_n \varepsilon_n] \neq 0$$

- Why might this be true?

We can re-write our moments with instruments instead of the explanatory variables

$$E[[y_n - P_n(x_n, \beta_0)]z_n] = 0$$
$$\Rightarrow \quad \frac{1}{N}\sum_{n=1}^{N}[y_n - P_n(x_n, \hat{\beta})]z_n = 0$$

Instruments to include in model

Exogenous variables: constant term and system cost

- These variables are their own instruments

Endogenous variable: operating cost

- Instrument with electricity price and square footage

But now we have four moment conditions (one for each instrument) but only three parameters

- This model is now overidentified
- We will use optimal (two-step) GMM to estimate it

Function to Calculate Moments with Instruments

```
### Calculate GMM estimator for binary logit AC choice using cost data
### instrumenting for operating cost with elec price and square feet
## Function to calculate binary logit moments with instruments
calculate_moments_iv <- function(parameters, data){
  ## Extract explanatory variable data from matrix
  data_x <- data[, 2:4]
  ## Extract choice data from matrix
  data_y <- data[, 1]
  ## Extract instrument data from matrix
  data_z <- data[, c(2:3, 5:6)]
  ## Calculate net utility of alternative given the parameters
  utility <- data_x %*% parameters
  ## Caclculate logit probability of alternative given the parameters
  probability_choice <- 1 / (1 + exp(-utility))
  ## Calculate residuals
  residuals <- data_y - probability_choice
  ## Create moment matrix
  moments <- c(residuals) * data_z
  return(moments)
}
```

Estimate GMM Parameters

```r
## Create dataset for use in GMM IV moment function
data_gmm_iv <- data %>%
  mutate(air_conditioning = 1 * air_conditioning,
         constant = 1) %>%
  select(air_conditioning, constant, cost_system, cost_operating,
         elec_price, square_feet) %>%
  as.matrix()
## Find the GMM IV estimator
model_gmm_iv <- gmm(calculate_moments_iv, data_gmm_iv, rep(0, 3),
                    vcov = 'iid',
                    control = list(reltol = 1e-25, maxit = 10000))
```

Estimation Results

```
## Summarize GMM IV model results
model_gmm_iv %>%
  summary()
```

```
##
## Call:
## gmm(g = calculate_moments_iv, x = data_gmm_iv, t0 = rep(0, 3),
##     vcov = "iid", control = list(reltol = 1e-25, maxit = 10000))
##
##
## Method:  twoStep
##
## Coefficients:
##           Estimate     Std. Error   t value      Pr(>|t|)
## Theta[1]   5.0037e+00   9.3613e-01   5.3451e+00   9.0385e-08
## Theta[2]  -5.4824e-03   1.4640e-03  -3.7449e+00   1.8046e-04
## Theta[3]  -1.0135e-02   2.1874e-03  -4.6334e+00   3.5967e-06
##
## J-Test: degrees of freedom is 1
##                 J-test     P-value
## Test E(g)=0:    0.0091463  0.9238094
##
## Initial values of the coefficients
##     Theta[1]     Theta[2]     Theta[3]
##  5.023667023 -0.005509947 -0.010157435
##
## #############
## Information related to the numerical optimization
## Convergence code =  0
## Function eval. =  463
## Gradian eval. =  NA
```

Test of Overidentifying Restrictions

```r
## Test overidentifying restrictions
model_gmm_iv %>%
  specTest()

##
##  ##  J-Test: degrees of freedom is 1  ##
##
##                   J-test     P-value
## Test E(g)=0:    0.0091463  0.9238094
```

Announcements

Reading for next time

- Train textbook, Chapter 4

Upcoming

- Problem Set 3 will be posted soon, due October 31