

Exercise 1.1, Structural Econometrics

Giovanni Cavalcanti

2024-10-30

Data and model

We have a panel with 50 products dispersed over 20 geographic markets. For each product, we have its market share, 4 product characteristics, prices and 6 instrumental variables for it. The consumer utility is defined as

$$U_{ijm} = -\alpha p_{jm} + \sum_{k=1}^3 \beta_k x_{jmk} + \beta_{4i} x_{jm4} + \xi_j + \xi_{jm} + \varepsilon_{ijm}$$

where i denotes individuals, j denotes choices and m markets. $\beta_{4i} = \beta_4 + \sigma_1 v_1$ and $v_i \sim N(0,1)$, ξ_j is a product fixed effect, ξ_{jm} denotes characteristics of product j at market m that is observed by consumers and firms (but not to the econometrician) and ε_{ijm} is iid EV across (i, j, m) . We further assume that there is a outside good such that $U_{im0} = \varepsilon_{im0}$ for each m .

We load the dataset and make preliminary manipulations on it

```
dataset <- read_excel(path = "dataset_ps1.xlsx") %>%
  mutate(market = as.factor(market),
         product = as.factor(product))
```

```
## # A tibble: 6 x 14
##   market product  share    x1    x2    x3    x4  price  iv1  iv2  iv3  iv4
##   <fct>  <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1      1      0.00541 0.529 1.15    0 1.89 0.00494 5.44 5.40 5.90 5.69
## 2 1      2      0.00115 0.494 1.28    0 1.94 0.00413 5.89 6.21 6.18 6.11
## 3 1      3      0.00131 0.468 1.46    0 1.72 0.00590 7.89 7.48 7.77 7.15
## 4 1      4      0.00133 0.427 1.61    0 1.69 0.00541 7.58 7.33 7.01 7.60
## 5 1      5      0.00314 0.452 1.65    0 1.50 0.00820 8.93 9.58 8.93 8.95
## 6 1      6      0.00305 0.451 1.62    0 1.73 0.00616 7.79 7.22 7.37 7.77
## # i 2 more variables: iv5 <dbl>, iv6 <dbl>
```

Problem 1: Multinomial Logit

Assume that $\sigma_1 = 0$, and $\varepsilon_{ijm} \sim TIEV(\mu, \theta)$, where $\mu > 0$ is the scale parameter and θ is the location parameter of the distribution, i.e. its distribution is $F_\varepsilon(x) = \exp(-\exp(-\mu(x - \theta)))$

Derive the share of each choice in each market.

Since we already observe the market share for each product, we only need to calculate the share for the outside option and add it to the dataset. The outside option's share represents the probability of consumers choosing none of the observed products in each market, defined as $1 - \sum_j \text{share}_j$ for each market.

The following code calculates this share and appends it to the dataset:

```
# Calculate outside good share and add outside good
outside_goods <- dataset %>%
  group_by(market) %>%
  summarize(
    product = as.factor("0"), # Label for outside good
    share = 1 - sum(share), # Calculated outside share
    x1 = 0, x2 = 0, x3 = 0, x4 = 0, # Characteristics set to 0
    price = 0, # Price of outside good is 0
    iv1 = 0, iv2 = 0, iv3 = 0, iv4 = 0, iv5 = 0, iv6 = 0 # Instrument variables set to 0
  )

# Bind the outside good rows to the original dataset and arrange by market
dataset <- bind_rows(dataset, outside_goods) %>%
  arrange(market)
```

To preview the shares for the outside option across markets:

```
## # A tibble: 6 x 3
##   market product share
##   <fct>   <fct>   <dbl>
## 1 1      0      0.433
## 2 2      0      0.455
## 3 3      0      0.436
## 4 4      0      0.463
## 5 5      0      0.437
## 6 6      0      0.427
```

1.2 Is μ identified? Why? And what about θ ?

In general, the scale parameter μ cannot be identified. Suppose that $\varepsilon_{ij} \sim T1EV(\mu_j, \theta_j)$ and $\varepsilon_{ik} \sim T1EV(\mu_k, \theta_k)$ represent the unobserved idiosyncratic utility shocks, where we assume orthogonality between these terms. When we take the difference $\varepsilon_{ik} - \varepsilon_{ij}$, it yields a logistic distribution that depends only on the difference in location parameters $\theta_j - \theta_k$ and not on the scale μ :

$$\varepsilon_{ik} - \varepsilon_{ij} \sim \text{Logistic}(\theta_j - \theta_k).$$

This difference reflects the general case of choosing product j over any other product k in the market, represented by:

$$P(a_i = k) = \text{Prob}(\varepsilon_{ik} - \varepsilon_{ij} < V_{ij} - V_{ik}, \forall k \neq j).$$

Thus, μ cannot be directly identified from this structure since it cancels out in the difference.

Regarding θ , the location parameter, it is only partially identifiable. By anchoring θ to an outside option with an observed utility level of zero, we set a baseline against which the utility levels of all other options are identified. This normalization allows us to interpret and identify the utilities of other products relative to the outside good, assuming its utility level is fixed at zero.

1.3 Suppose that $\mu = 1, \theta = 0$. Suppose that you want to estimate the model using OLS.

a) Derive the equation you will use to estimate the model.

We first start with the closed logistic formula for the share in each market

$$P(a_i = j | V_{i1}, \dots, V_{i50}) = \frac{\exp(V_j)}{\sum_{k=1}^{50} \exp(V_k)}$$

Where V_k is the vector of observed characteristics and price. If we divide each probability, which is the observed market share, the denominator cancels out, yielding

$$\frac{P(a_i = j | V_{i1}, \dots, V_{i50})}{P(a_i = 0 | V_{i1}, \dots, V_{i50})} = \frac{\exp(V_j)}{\exp(V_0)}$$

Taking the logs on both sides, we linearize it

$$\ln(P(a_i = j | \cdot)) - \ln(P(a_i = 0 | \cdot)) = V_j - V_0$$

Therefore, our estimating equation for this aggregated model is

$$\ln(\hat{P}(a_i = j | \cdot)) - \ln(\hat{P}(a_i = 0 | \cdot)) = \sum_{l=1}^L \beta_l (x_{jl} - x_{ol}) + (\xi_j - \xi_o)$$

$$\ln(\hat{P}(a_i = j | \cdot)) - \ln(\hat{P}(a_i = 0 | \cdot)) = \sum_{l=1}^L \beta_l (x_{jl} - x_{ol}) + (\xi_j^*)$$

b) Under what conditions the OLS estimator is consistent?

The model parameters β can only be consistently estimated if the vectors of observed characteristics V_j contains no unknowns or unobservables, as to have no endogeneity problem in the estimation.

1.4 Assume that $\xi_j = 0$. Estimate the parameters of the model by OLS and IV. Compare the estimates of α . Does the OLS bias have the expected sign? Explain.

The following code fits the two requested models

```
# Step 1: Create the log difference between product shares and the outside option
```

```
# Filter to get the outside share for each market
```

```
outside_share <- dataset %>%  
  filter(product == "0") %>%  
  select(market, share) %>%  
  rename(outside_share = share)
```

```
# Join with the original dataset to add the outside share to each product row
```

```
dataset_with_outside <- dataset %>%  
  left_join(outside_share, by = "market") %>%  
  filter(product != "0") %>% # Exclude the outside good itself  
  mutate(log_diff_share = log(share) - log(outside_share))
```

```
# Step 2: Estimate the model using OLS with product characteristics (x1 to x4) and price
```

```
# Fit the OLS model
```

```
ols_model <- lm(log_diff_share ~ x1 + x2 + x3 + x4 + price, data = dataset_with_outside)
```

```
# Step 3: Define IV model with instrumental variables (IV1-IV6) for endogenous variable (price)
```

```
iv_model <- ivreg(log_diff_share ~ x1 + x2 + x3 + x4 + price | . - price + iv1 + iv2 + iv3 + iv4 + iv5  
  data = dataset_with_outside)
```

```
##
## Comparison of OLS and IV Model Results
## =====
##                               Dependent variable:
##                               -----
##                               Log Difference in Share (Product vs. Outside)
##                               OLS Model          IV Model
##                               (1)              (2)
## -----
## Characteristic X1              0.350          0.778*
##                               (0.436)         (0.438)
##
## Characteristic X2              1.623***       1.664***
##                               (0.195)         (0.196)
##
## Characteristic X3              0.581***       0.797***
##                               (0.128)         (0.129)
##
## Characteristic X4              0.329***       0.326***
##                               (0.067)         (0.067)
##
## price                         -50.024***     -70.161***
##                               (8.621)         (8.702)
##
## Constant                      -7.357***     -7.424***
##                               (0.392)         (0.393)
## -----
## Observations                   1,000         1,000
## R2                             0.089         0.084
## Adjusted R2                   0.084         0.079
## Residual Std. Error (df = 994) 1.077         1.080
## F Statistic                   19.401*** (df = 5; 994)
## =====
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```

Controlling for the instruments given yields an α estimand with higher magnitude, this is expected as the endogeneity problem would lower our price coefficient results.

1.5 Based on the IV parameters estimated above compute own- and cross-price elasticities for each good in market one. Discuss the results

The following code calculates the elasticities for market 1

```
# Select data for market 1
market_data <- dataset %>% filter(market == 1, product != 0)

# Extract coefficients from the model
alpha_price <- iv_model$coefficients["price"] # Coefficient for price

# Calculate Own-Price Elasticities
market_data <- market_data %>%
  mutate(own_price_elasticity = price * alpha_price * (1 - share))
```

```

# Calculate Cross-Price Elasticities
# Create a dataframe for cross-price elasticities
cross_price_elasticities <- expand.grid(j = market_data$product, k = market_data$product)
cross_price_elasticities <- merge(cross_price_elasticities, market_data[, c("product", "price", "share")])
cross_price_elasticities <- merge(cross_price_elasticities, market_data[, c("product", "price", "share")])

# Calculate cross-price elasticities
cross_price_elasticities <- cross_price_elasticities %>%
  mutate(cross_price_elasticity = -price_k * alpha_price * share_k)

```

The first results for the elasticity matrix are

```

# Create a full elasticity matrix
# Initialize a matrix with NA values
num_products <- nrow(market_data)
elasticity_matrix <- matrix(NA, nrow = num_products, ncol = num_products)

# Fill the diagonal with own-price elasticities
rownames(elasticity_matrix) <- market_data$product
colnames(elasticity_matrix) <- market_data$product

# Fill in cross-price elasticities
for (row in 1:nrow(cross_price_elasticities)) {
  j <- as.character(cross_price_elasticities$j[row])
  k <- as.character(cross_price_elasticities$k[row])
  elasticity_matrix[j, k] <- cross_price_elasticities$cross_price_elasticity[row]
}

for (i in 1:num_products) {
  elasticity_matrix[i, i] <- market_data$own_price_elasticity[i]
}

# Print the elasticity matrix
print(elasticity_matrix[1:5,1:5])

```

```

##           1           2           3           4           5
## 1 -0.344418487  0.000333326  0.0005417495  0.0005033182  0.001806601
## 2  0.001875052 -0.289475587  0.0005417495  0.0005033182  0.001806601
## 3  0.001875052  0.000333326 -0.4132807199  0.0005033182  0.001806601
## 4  0.001875052  0.000333326  0.0005417495 -0.3787504207  0.001806601
## 5  0.001875052  0.000333326  0.0005417495  0.0005033182 -0.573603896

```