Empirical IO, HW2

Shao Y.

May 21, 2022

Contents

1 Logit Model

Summary statistics

Statistic	N	Mean	St. Dev.	Min	Max
move	10,296	32.519	34.659	1	700
price	10,296	0.047	0.006	0.031	0.058
profit	10,296	14.895	8.265	−7.740	27.920
custcoun	10,296	24,668.000	3,481.407	15,881	38,200

Results Using `stargazer(style = "aer")` and `wholesale` as instrument for `price`

	Dependent variable:	
	$\ln(s_j) - \ln(s_0)$	
	OLS	*instrumental variable*
price	−103.050***	−108.623***
	(1.601)	(3.195)
Observations	10,296	10,296
R^2	0.655	0.655
Adjusted R^2	0.655	0.654
Residual Std. Error (df = 10284)	0.497	0.497
F Statistic	1,776.858*** (df = 11; 10284)	
Note:		*p<0.1; **p<0.05; ***p<0.01

2 BLP Model

Using $\ln s_{jt} - \ln s_0$ as initial value of δ_{jt}, and applying **2SLS** for linear coefficients and Nelder-Mead algorithm for nonlinear coefficients, the outcomes are as follows:

Linear coefficients		Nonlinear coefficients	
Price	−141.678	π	−0.112
		σ	1.072

Code for BLP, using R.

```
# initializing ------------------------------------------------
setwd("~/***")
require(dplyr); require(tidyr); require(data.table); require(magrittr);
require(stargazer); require(readxl); require(MASS)
rm(list = ls())
options(scipen = 10)
# -1: data & variables ----------------------------------------------
load("beer.RData") # data from "logit_main.R"
demo <- read_xlsx("DEMO.xlsx", sheet = 1) %>%
  `[`(!is.na(.$income), c(8, 16, 17)) %>%
  `names<-`(c("store", "inc", "inc_sd")) %>%
  setDT()
market <- expand(beer, store, week) %>% setDT() # 936 store * week
market <- demo[market, on = "store"]

# 0: draw consumers for each t (store * week) -------------------------
R <- 20 #   GLOBALLY        |    draw: 20 consumers (R = 20)
J <- 11 #     SET          | product: 11
nmarket <- dim(market)[1] # |  market: T = 936
D <- apply(market, MARGIN = 1, function(x) {
  rnorm(R, x[2], x[3])
}) %>% t()
V <- matrix(rnorm(nmarket * R, 0, 1), ncol = R)
draws_beer <- list("D" = D, "V" = V)

# 1: share calculation ----------------------------------------------
# R, J, nmarket are GLOBALLY SET above
share_cal <-
  function(delta, rc, chrs){
    sapply(c(1:J), function(j) {
      exp(delta[j] + as.matrix(chrs[j, ]) %*% as.matrix(rc))
                                        # 1*R, exp component
      }) %>% t() %>%                    # J*R, exp component
      apply(MARGIN = 2, function(i) {
        i / (sum(i) + 1)}) %>%          # J*R, i's simulated prob of j
      apply(MARGIN = 1, mean) %>%       # J*1, j's simulated share at t
      return()
  }

share_cal_loopT <-
  function(delta, coefs, draws, Chrs_T) {
    sapply(c(1:nmarket), function(t) {
```

```
        share_cal(
          delta = as.matrix(delta[c((J * (t - 1) + 1):(t * J)), ]),
          rc = coefs[["Pi"]] %*% draws[["D"]][t, ] + # if K,D > 1, "D"=>"D_1"...
            coefs[["Sigma"]] %*% draws[["V"]][t, ],  # i.e. all D_d, V_k
          chrs = as.matrix(Chrs_T[c((J * (t - 1) + 1):(t * J)), ])
        )
      }, simplify = T) %>% c() %>%
        return()
    }

  # 2: contract mapping ------------------------------------------------
  delta_cal <-
    function(delta0, coefs, draws, share, Chrs_T){
      epsilon <- 100
      delta_old <- delta0
      h <- 1
      while (epsilon > exp(-2) & h <= 10000) {
        delta_new <- delta_old + log(share) - log(
          share_cal_loopT(delta_old,
                          coefs  = coefs,
                          draws  = draws,
                          Chrs_T = Chrs_T)
        )
        epsilon <- max(abs(delta_new - delta_old))
        delta_old <- delta_new
        h <- h + 1
      }
      if (h > 10000) {
        stop("No convergence found")
      } else {
        return(delta_new)
      }
    }

  # 3: NLLS ------------------------------------------------
  xi_sqr <-
    function(coef_vec, delta0, draws, Chrs_T, share, X, Z, edg) {
      K <- dim(Chrs_T)[2]
      L <- length(coef_vec)
      coefs <- list("Pi" = matrix(coef_vec[1:(L - K^2)], nrow = K),
                    "Sigma" = matrix(coef_vec[-c(1:(L - K^2))], nrow = K))
      delta_hat <- delta_cal(delta0, coefs, draws, share, Chrs_T)
      X_f <- cbind(Chrs_T, X)              # names of TSLS == names of X_f
      Z_f <- cbind(Chrs_T[, -edg], Z, X)   # endogenous columns in edg
      TSLS <<- ginv(t(X_f) %*% Z_f %*% ginv(t(Z_f) %*% Z_f) %*% t(Z_f) %*% X_f) %*%
        t(X_f) %*% Z_f %*% ginv(t(Z_f) %*% Z_f) %*% t(Z_f) %*% delta_hat
      xi <- delta_hat - X_f %*% TSLS
      message(paste0("Linear coefficient: ", paste(TSLS, collapse = ", ")))
      return(sum(xi ^ 2))
    }

  # example: beer case ------------------------------------------------
  BLP <- optim(
    c(10, 10), xi_sqr,
    delta0 = as.matrix(beer$logit_y),
```

```
      draws = draws_beer,
      Chrs_T = as.matrix(beer$price),
      share = as.matrix(beer$share),
      X = as.matrix(beer[, grep("^upc([1-9]|10)$", names(beer)), with = F]),
      Z = as.matrix(beer$wholesale),
      edg = 1L
    )
```