

```
In [15]: using Random, Statistics, Parameters, StatsBase, Distributions, Optim, ForwardDiff, Calculus, LinearAlgebra, DataFrames
```

Throught the questions, I report mostly results. All other codes are included in the code file (ps1.jl).

Question 1

- $M = 250$ markets
- Market characteristics: $X_m \sim N(3, 1), m = 1, 2, \dots, 250$.
- Firm-specific characteristics: $Z_{fm} \sim N(0, 1), f = 1, \dots, F_m, \quad m = 1, \dots, 250$.

Data generating

```
In [64]: @with_kw mutable struct parameters
    M::Int64 = 250
    α::Float64 = 1.0
    β::Float64 = 1.0
end

param = parameters()
θ = 1.0;
μ = 2.0;
σ = 1.0;
tru_param = [μ, σ, θ];
```

Market characteristics and draw numbers of potential entrants in each market

```
In [65]: X = rand(Normal(3,1), param.M)
F = [2,3,4]
entrant = sample(MersenneTwister(342), F, param.M; replace = true, ordered = false);
```

Draw firm-specific and unobservable fixed cost

```
In [67]: uf_num = rand(MersenneTwister(123), Normal(tru_param[1], tru_param[2]), sum(entrant, dims = 1)[1])
z_firm = rand(Normal(0,1), sum(entrant, dims = 1)[1])
u_firm_new = Vector{Float64}[]
z_firm_new = Vector{Float64}[]
k = 1
j = 0
for i in eachindex(entrant)
    j += entrant[i]
    temp_1 = uf_num[k:j]
    temp_2 = z_firm[k:j]
    u_firm_new = push!(u_firm_new, temp_1)
    z_firm_new = push!(z_firm_new, temp_2)
    k = j + 1
end
Z = copy(z_firm_new);
```

Get equilibrium entered firm numbers and firm's entry decisions.

To compute equilibrium entered firm numbers and each firm's decisions. I follow the specification in Berry (1992). The number of firms that enter the market m is computed as

$$N_m = \max_{0 \leq n \leq F_m} (n : \#\{f : \Pi_{mf}(n, u_{fm}) \geq 0\} \geq n)$$

Firm decisions can be computed by checking if each firm's the cost-rank number is less than equal to the number of entrants in the market m (N_m).

```
In [68]: entered_firm, decision = eq_firm_calc(tru_param, param, X, entrant, Z, u_firm_new);
```

For the expositional purpose, I here create dataframe (In the actual estimation, this dataframe is not used)

```
In [47]: first(data_1, 10)
```

Out[47]: 10 rows × 5 columns

	market_index	observed_profit	potential_firm_number	entry_decision	eq_firm_number
	Int64	Float64	Int64	Int64	Int64
1	1	0.914035	4	0	0
2	1	0.417636	4	0	0
3	1	-0.479137	4	0	0
4	1	-2.34252	4	0	0
5	2	2.63331	2	1	1
6	2	0.210277	2	0	1
7	3	2.17458	4	0	0
8	3	1.49642	4	0	0
9	3	1.04001	4	0	0
10	3	0.992963	4	0	0

- Data show market index, observed fixed costs (market + firm specific), potential firm numbers for each market, each firm's entry decision, and equilibrium entered firm numbers for each market.

Question 2 : Probit estimator

Following one of the special cases explained in Berry (1992), I focus on the probabilities of the number of firms in each market with three cases.

- $Pr(N = 0)$
- $Pr(N = 1)$
- $Pr(N = 2)$

Compute probit estimates and standard errors for (μ, σ, δ)

Here I use BFGS for the probit estimator and compute standard error using the information matrix.

- Do you need to make any equilibrium selection assumptions?

I made an equilibrium selection assumption that profitable firms enter first sequentially.

```
In [69]: opt_probit = Optim.optimize(vars -> entry_probit(vars, param, X, Z, entrant, entered_firm), ones(3), BFGS(), Optim.Options(show_trace = false, g_tol = 1e-7));
estimates_probit = opt_probit.minimizer
```

```
Out[69]: 3-element Vector{Float64}:
 2.16704931717102835
 0.9586206048170339
 0.6026102979703265
```

```
In [58]: hessian_probit = hessian( vars -> entry_probit(vars, param, X, Z, entrant, entered_firm) )
se_probit = diag(inv(hessian_probit(estimates_probit)))
```

```
Out[58]: 3-element Vector{Float64}:
 0.019013639206240877
 0.019354139457049585
 0.03598819093003883
```

Question 3 and 4 : Method of Simulated Moments

MSM estimator (including incorrect specification cases)

(a-1) The correctly specified model: Using identities of firms and numbers of entered firms

```
In [91]: opt_identity = Optim.optimize(vars -> simulated_mm(vars, param, X, Z, entered_firm, decision, entrant, 200, "identity"), ones(3), Optim.Options(show_trace = false, g_tol = 1e-7));
estimates_identity = opt_identity.minimizer
```

```
Out[91]: 3-element Vector{Float64}:
 1.9816565196623905
 0.851003693563963
 1.1334036592648404
```

(a-2) The correctly specified model: Using just the numbers of entered firm.

```
In [103]: opt_number = Optim.optimize(vars -> simulated_mm(vars, param, X, Z, entered_firm, decision, entrant, 200, "number"), ones(3), Optim.Options(show_trace = false, g_tol = 1e-7));
estimates_msm = opt_number.minimizer
```

```
Out[103]: 3-element Vector{Float64}:
 1.933239484005392
 0.9789171418973519
 1.1882196697374585
```

(b-1) The incorrectly specifcied model: Using identities of firms and numbers of entered firms.

```
In [92]: opt_identity_rev = Optim.optimize(vars -> simulated_mm(vars, param, X, Z, entered_firm, decision, entrant, 200, "identityrev"), ones(3), Optim.Options(show_trace = false, g_tol = 1e-7));
estimates_identity_rev = opt_identity_rev.minimizer
```

```
Out[92]: 3-element Vector{Float64}:
 1.2231182031066525
 2.7818914310742255
 2.2687948757076404
```

(b-2) The incorrectly specifcied model: Using just the numbers of entered firm.

```
In [108]: opt_number_rev = Optim.optimize(vars -> simulated_mm(vars, param, X, Z, entered_firm, decision, entrant, 200, "numberrev"), ones(3), Optim.Options(show_trace = false, g_tol = 1e-7));
estimates_msm_rev = opt_number_rev.minimizer
```

```
Out[108]: 3-element Vector{Float64}:
 1.9550171263455598
 1.917147396384817
 1.25705981799055
```

Standard error: Bootstrap (Bootstrap simulation : 100 times)

- Bootstrapping procedures

- Random sample with replacement for Z_{fm} for each market.
- Solve equilibrium entered firm numbers and entry decisions.
- Get estimates for μ, σ , and δ .
- Repeat step 1-3 for S times (S is bootstrapping number).

After bootstrapping, get standard errors of bootstrapped estimates.

```
In [83]: ident, num, ident_rev, num_rev = msm_bootstrap(param, X, Z, u_firm_new, entrant, 100);
```

Results: (Standard errors are reported in the brackets)

I omit codes for the estimations here (There are provided in my julia code file)

	Probit estimator	Specification 1 (Identity & Number)	Specification 2 (Number)	Specification 3 (Identity & Number, Incorrect)	Specification 4 (Number, Incorrect)
	Estimate (Standard error)	Estimate (Standard error)	Estimate (Standard error)	Estimate (Standard error)	Estimate (Standard error)
μ	2.0133 (0.016)	1.9817 (0.1176)	1.9332 (0.1176)	1.2231 (0.3405)	1.9550 (0.1491)
σ	0.9053 (0.0078)	0.8510 (0.1054)	0.9789 (0.1054)	2.7818 (0.2340)	1.9171 (0.1320)
δ	0.7073 (0.0236)	1.1334 (0.1452)	1.1882 (0.1452)	2.2688 (0.4972)	1.2570 (0.2315)

Discussion: Choice of moments

- First specification: using identities and numbers of firms entered

For the first specification (using identities and numbers of firms entered), population moments are

$$m(\theta) = E[(D - D(\theta), N - N(\theta))] = 0, \text{ where } D \text{ is entry decisions and } N \text{ is entered firm numbers.}$$

Corresponding sample moments are

$$\hat{m}(\theta) = \frac{1}{M} \sum_{m=1}^M \left((D_m - \hat{D}(\theta)_m, N_m - \hat{N}(\theta)_m) \right)$$

and Simulated Method of Moments replace $\hat{D}(\theta), \hat{N}(\theta)$ with $\tilde{D}(\theta) = \frac{1}{S} \sum_s D^s(\theta), \tilde{N}(\theta) = \frac{1}{S} \sum_s N^s(\theta)$

More specifically, the sample moment condition $\hat{m}(\theta)$ is $K \times 1$ vector, where $K = \sum_{m=1}^M F_m + M$ where F_m is potential entrant numbers in market m .

- Second specification: using only numbers of firms entered

$$m(\theta) = E[(N - N(\theta))] = 0, \text{ where } D \text{ is entry decisions and } N \text{ is entered firm numbers.}$$

Corresponding sample moments are

$$\hat{m}(\theta) = \frac{1}{M} \sum_{m=1}^M \left(N_m - \hat{N}(\theta)_m \right)$$

and Simulated Method of Moments replace $\hat{N}(\theta)$ with $\tilde{N}(\theta) = \frac{1}{S} \sum_s N^s(\theta)$.

Sample moment condition $\hat{m}(\theta)$ is $M \times 1$ vector.

Then the GMM estimator $\hat{\theta}_{gmm}$ is

$$\hat{\theta}_{gmm} = \operatorname{argmin}_{\theta} \hat{m}^{\top}(\theta) W \hat{m}(\theta), \text{ where } W, \text{ weighting matrix, is an identity matrix here.}$$

- Due to the discontinuity, I use a non-derivative method (Nelder-Mead)

Discussion: Estimates

- Case 1: Using both identities and numbers

Comparing specifications 1 and 3 (Using both identities and numbers of firms entered), incorrectly specified estimation gives bigger estimates for σ and δ . This is reasonable because, in the incorrectly specified model, firms with the highest fixed costs (lowest profitability) enter first so the competitive effect, δ , is bigger than that of the correctly specified case. Since the because of entry is the opposite in the incorrect model, the variance of unobservable fixed costs increases while the mean level of unobservable fixed costs decreases. The standard errors also show that the incorrect model estimates are noisier than those of the correctly specified model.

- Case 2: Using only firm numbers

Comparison of specifications 2 and 4 show the difference between estimates and standard errors when the entry order is correct and incorrect. While specification 4 indicates the variance of unobservable fixed costs and the competitive effect increase when the entry sequence is incorrectly specified, the magnitude of the difference is smaller than that of case 1. This is because in case 1 (using both identities and entered firm numbers), misspecification causes biased predictions for identities mainly. The misspecification also causes wrong predictions on the numbers of firms entered in each market but it is not as severe as misprediction of firms' entry decisions.

Question 5: Moment inequality

For the moment inequality estimation, the objective function is

$$Q(\theta) = \int \|P(x) - H_1(x, \theta)_- \| + \|P(x) - H_2(x, \theta)_+ \| dF_x$$

and the sample conditions are

$$Q_n(\theta) = \frac{1}{N} \sum_{i=1}^n \|P_n(x_i) - \hat{H}_1(x_i, \theta)_- \| + \|P_n(x_i) - \hat{H}_2(x_i, \theta)_+ \|\|$$

where H_1 and H_2 are replaced with $\tilde{H}_1 \equiv \frac{1}{R} \tilde{H}_1(X, \theta)$ and $\tilde{H}_1 \equiv \frac{1}{R} \tilde{H}_2(X, \theta)$, respectively.

First stage. $Pr(y|X) \sim P_n(x_i)$

First stage is to estimate $Pr(y|X)$ using a nonparametric frequency estimator. This procedure is done by counting market specific entry decisions. (I include the estimation code in the code file).

1. $P_n(x)$ when the market has 2 potential entrants

```
In [114]: println(p1)
```

[0.3979591836734694, 0.23469387755102042, 0.22448979591836735, 0.14285714285714285]

2. $P_n(x)$ when the market has 3 potential entrants

```
In [115]: println(p2)
```

[0.18666666666666668, 0.14666666666666667, 0.13333333333333333, 0.17333333333333334, 0.12, 0.08, 0.10666666666666667, 0.08]

3. $P_n(x)$ when the market has 4 potential entrants

```
In [116]: println(p3)
```

[0.11688311688311688, 0.012987012987012988, 0.03896103896103896, 0.09090909090909091, 0.025974025974025976, 0.07792207792207792, 0.1038961038961039, 0.07792207792207792, 0.06493506493506493, 0.07792207792207792, 0.06493506493506493, 0.07792207792207792, 0.025974025974025976, 0.025974025974025976, 0.06493506493506493, 0.05194805194805195]

Second stage. $Q(\theta)$

To implement the second stage, I implement H_1 and H_2 following the procedures explained in Ciliberto and Tamer (2009) supplementary material.

θ here is $\mu, \sigma, \delta_j^i, \quad \forall j \neq i, \quad \forall i \neq j$

Estimates

$$\mu = 2.9334329179686596 \quad \sigma = 0.5872881231290252 \quad \delta_1^2 = 0.8139350178194646 \quad \delta_2^1 = 0.7921335593224738 \quad \delta_1^3 = 1.2476807781835633 \quad \delta_2^3 = 1.4537333669587873$$

$$\delta_1^1 = 0.790593791266027 \quad \dots \delta_3^1 = 1.3832306774113636 \quad \delta_2^2 = 0.6882627719479003 \quad \delta_3^2 = 1.4116227470387794 \quad \delta_4^1 = 0.8604243251831124 \quad \delta_4^2 = 0.8701771450292943$$

$$\delta_4^3 = 0.8930463362880245$$

Constrcut confidence interval

I wasn't able to finish this part due to time limit. I will finish this part and resubmit the file.

Estimates from the moment inequality are different from the ones in Q2 and Q3.4. Competitive effects are much more flexible. The previous estimations assume $\delta = \delta_j^i, \quad \forall i, j$, the moment inequality estimation can estimate fully flexible δ_j^i . Also, I did not impose any selection rule, such as lowest fixed cost firms entering first, for the estimation. This is a lot more flexible than the MSM estimator and probit estimator. Since I have not finished the confidence interval part, I am not 100% sure what the disadvantages are but this estimation takes longer time. (I will fix the incomplete part...).