

Problem Set 2 Question 1

Consider a static entry game between J firms operating in T markets. Upon entry, the profit of firm j in market t depends on the number of entrants and is given by:

$$\pi_{jt}(n_{-jt}) = x_{jt}\beta - \phi_j - \delta_j \log(1 + n_{-jt}) + \epsilon_{jt}$$

where x_{jt} are market-firm specific profits shifter, ϕ_j is a firm specific fixed cost, n_{-jt} is the number of competitors in market t and $\epsilon_{jt} \sim F$ is an idiosyncratic shock. Given n_{-jt} firms, firm j enters if and only if $\pi_{jt} \geq 0$.

Part 1

In the case with $J = 2$, we can characterize the payoff matrix as follows:

	E ₂	O ₂
E ₁	$(x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t}, x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t})$	$(x_{1t}\beta - \phi_1 + \epsilon_{1t}, 0)$
O ₁	$(0, x_{2t}\beta - \phi_2 + \epsilon_{2t})$	$(0, 0)$

The pure-strategy Nash equilibrium realization depends on the parameters and the realizations of $\epsilon_{i,j}$'s. Firm i enters market t if it earns a non-negative profit, that is, $\mathbf{1}\{\pi_{it} \geq 0\}$. Hence, we can summarize the realizations of Nash equilibrium and its corresponding region of $(\epsilon_{1,j}, \epsilon_{2,j})$ as follows:

- $\mathcal{R}_3 = \{(O_1, O_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 + \epsilon_{1t} < 0 \text{ and } x_{2t}\beta - \phi_2 + \epsilon_{2t} < 0\}$
- $\mathcal{R}_1 = \{(E_1, E_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} > 0 \text{ and } x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t} > 0\}$
- $\mathcal{R}_2 + \mathcal{R}_5 = \{(E_1, O_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 + \epsilon_{1t} > 0 \text{ and } x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t} < 0\}$
- $\mathcal{R}_4 + \mathcal{R}_5 = \{(O_1, E_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} < 0 \text{ and } x_{2t}\beta - \phi_2 + \epsilon_{2t} > 0\}$

Part 2

The overlapping region of likelihood between (E_1, O_2) and (E_2, O_1) generates problems for identification. Even if we have pinned down the parameters $(\beta, \phi_1, \phi_2, \delta_1, \delta_2)$, a pair $(\epsilon_{1t}, \epsilon_{2t})$ that lies in the region \mathcal{R}_5 can either generate (E_1, O_2) or (E_2, O_1) as an equilibrium outcome and predicting which of the two occurs is impossible without further assumptions. Specifying the likelihood function given the data for all the model parameters now will involve double-counting region \mathcal{R}_5 since this will be incorporated in the likelihood terms for both (E_1, O_2) and (E_2, O_1) . The sum of the probabilities will exceed 1, and the resulting likelihood is not well-defined.

Part 3

The assumption of sequential entry simplifies the problem. Now the region \mathcal{R}_5 can be attributed to the first entrant in the market, i.e. firm 1.

- $\mathcal{R}_3 = \{(O_1, O_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 + \epsilon_{1t} < 0 \text{ and } x_{2t}\beta - \phi_2 + \epsilon_{2t} < 0\}$
- $\mathcal{R}_1 = \{(E_1, E_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} > 0 \text{ and } x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t} > 0\}$
- $\mathcal{R}_2 + \mathcal{R}_5 = \{(E_1, O_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 + \epsilon_{1t} > 0 \text{ and } x_{2t}\beta - \phi_2 - \delta_2 \log 2 + \epsilon_{2t} < 0\}$
- $\mathcal{R}_4 = \{(O_1, E_2) \text{ is observed}\} = \{(\epsilon_{1t}, \epsilon_{2t}) : x_{1t}\beta - \phi_1 - \delta_1 \log 2 + \epsilon_{1t} < 0 \text{ and } x_{2t}\beta - \phi_2 + \epsilon_{2t} > 0\}$

The issue with \mathcal{R}_5 being counted twice is now resolved with the addition of sequential entry, and therefore, the likelihood is well-defined.

Part 4

Assume we have market level data on T markets. For each market t , we only observe the number of firms n_t and some market level characteristic x_t . Assume that firms are homogeneous and play a sequential entry game in each market with the same entry profits as before:

$$\pi_t(n_t) = x_t\beta - \phi - \delta \log(n_t) + \epsilon_t$$

n_t is a Nash Equilibrium under the sequential entry assumption. All firms are equally profitable and therefore we don't need the additional assumption of entrance in order of profitability as in the Berry (1992) case.

We assume that $\epsilon_t \stackrel{\text{i.i.d.}}{\sim} N(0, 1)$ and estimate the parameters (β, ϕ, δ) from the data in `pset2_ex1.csv` using MLE.

```
In [1]: import numpy as np
import scipy.stats as stats
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import torch
import torch.optim as optim
from torch.autograd import Variable
import torch.distributions as td
```

```
In [2]: data = pd.read_csv('ps2_ex1.csv')

n, x = np.array(data['n']), np.array(data['x'])
n, x = torch.tensor(n), torch.tensor(x)

max_N, max_T = n.max(), n.size(dim=0)
```

```
In [19]: # Set parameters for the ML optimization.
beta = Variable(torch.zeros(1) + 0.04, requires_grad=True)
phi = Variable(torch.ones(1), requires_grad=True)
delta = Variable(torch.ones(1) * 2, requires_grad=True)
```

The i.i.d. normal assumption on the unobserved term allows for ordered probit estimation. This means that the likelihood of observing n_t firms is given by:

$$\begin{aligned} \mathbb{P}[\text{observing } n_t \text{ firms}] &= \mathbb{P}[\pi_t(n_t) > 0, \pi_t(n_t + 1) < 0] \\ &= \mathbb{P}[x_t\beta - \phi - \delta \log(n_t) + \epsilon_t > 0, x_t\beta - \phi - \delta \log(n_t + 1) + \epsilon_t < 0] \\ &= \mathbb{P}[-x_t\beta + \phi + \delta \log(n_t) < \epsilon_t < -x_t\beta + \phi + \delta \log(n_t + 1)] \\ &= \Phi(-x_t\beta + \phi + \delta \log(n_t + 1)) - \Phi(-x_t\beta + \phi + \delta \log(n_t)) \end{aligned}$$

The log-likelihood for a single observation (x_t, n_t) at time t is given by

$$\log \ell(\beta, \delta, \phi | x_t, n_t) = \sum_{n=1}^N \mathbf{1}\{n_t = n\} \log(\Phi(-x_t\beta + \phi + \delta \log(n_t + 1)) - \Phi(-x_t\beta + \phi + \delta \log(n_t)))$$

This implies that the log likelihood corresponding to the entire dataset is

$$\mathcal{L}(\beta, \delta, \phi | \{x_t, n_t\}) = \sum_{t=1}^T \sum_{n=1}^N \mathbf{1}\{n_t = n\} \log(\Phi(-x_t\beta + \phi + \delta \log(n_t + 1)) - \Phi(-x_t\beta + \phi + \delta \log(n_t)))$$

```
In [20]: def log_lik(b, d, p):

    term1 = td.Normal(loc=0, scale=1).cdf(-x * b + p + d * torch.log(n + 1))
    term2 = td.Normal(loc=0, scale=1).cdf(-x * b + p + d * torch.log(n))

    obj = torch.log(term1 - term2).sum()

    return obj

# Define the adaptive gradient descent optimizer used to find the estimates.
opt = optim.Adam([beta, phi, delta], lr=0.05)
```

```
In [ ]: for epoch in range(2000):

    opt.zero_grad() # Reset gradient inside the optimizer

    # Compute the objective at the current parameter values.
    loss = - log_lik(beta, delta, phi)
    loss.backward() # Gradient computed.
    opt.step()      # Update parameter values using gradient descent.
```

```
print('Step {} : Loss = {}'.format(epoch, loss.detach().clone()))
```

```
In [24]: # Store parameters as tuple
theta = (beta, delta, phi)
# Fisher Information matrix.
info_mle = -torch.tensor(torch.autograd.functional.hessian(log_lik, theta))
# Compute variance matrix.
var_mle = torch.inverse(info_mle)/max_T
# Compute standard errors.
std_mle = np.sqrt(np.diag(var_mle))
```

	Estimate	Std. Error
β	2.1063	0.01300029
δ	10.3704	0.06390927
ϕ	1.5480	0.02433986

In []: