Example 3: Merger simulation and partial collusion

TSE, MRes

Objective

Develop a simple model to represent the beer market and simulate the MillerCoors merger

We want to simulate (i.e. generate data on market outcomes) for the following 4 situations:

- Initial market: 4 firms compete
- Merger with competition: Firms 1 and 2 merge and compete with Firms 3 and 4
- Merger with partial competition: Firm 3 partially collude with the merged firm

Demand

Even if our focus is the supply model, we need a demand, it generates firms' market power

Consider a simple demand model (logit model, no unobserved heterogeneity):

- Utility depends on intercept $\mathbf{x}$, price $\mathbf{p}$, unobserved characteristics $\boldsymbol{\xi}$
- We must specify parameter of $\mathbf{x}$, $\beta = 1$ and parameter for the price $\alpha = -0.8$
- $\boldsymbol{\xi} \sim N(0, 1)$
- Consider each firm sells only one product so $J = 4$. There is also the outside good which utility is normalized to 0
- Market shares are such that:

$$s_j = \frac{\exp(\beta + \alpha p_j + \xi_j)}{1 + \sum_{j'} \exp(\beta + \alpha p_{j'} + \xi_{j'})}$$

They are jointly determined with the prices

Supply model: competition

Firms sell each one product, characterized by a cost c_j drawn from a U(2,3)

Under competition, each firm independently sets its price to maximize profit

For Firm 1:

$$\Pi_1 = (p_1 - c_1)s_1(\mathbf{p})$$

The FOC that defines the optimal price p_1^* is:

$$p_1^* = c_1 - \frac{s_1(\mathbf{p})}{\partial s_1(\mathbf{p})/\partial p_1}$$

Under the logit demand: $\partial s_1(\mathbf{p})/\partial p_1 = \alpha s_1(\mathbf{p})(1 - s_1(\mathbf{p}))$ So the FOC:

$$p_1^* = c_1 - \frac{1}{\alpha(1 - s_1(\mathbf{p}))}$$

This is the reaction function of Firm 1.

Equilibrium

At the equilibrium, all firms play their best responses to each other: Solve for $p_1, \ldots, p_J$ such that:

$$\begin{cases} p_1 = c_1 - \frac{1}{\alpha(1-s_1(p_1,\ldots,p_J))} \\ \vdots \\ p_J = c_J - \frac{1}{\alpha(1-s_J(p_1,\ldots,p_J))} \end{cases}$$

Using a non-linear system of equation solver “fsolve” or a recursive method (see algorithms discussed in Lecture 2)

Equilibrium after the merger

Let's express the profit of the merged entity:

$$\Pi_{12} = \Pi_1 + \Pi_2 = (p_1 - c_1)s_1(\mathbf{p}) + (p_2 - c_2)s_2(\mathbf{p})$$

FOC associated to the optimal price p_1^*:

$$s_1(\mathbf{p}) + (p_1 - c_1)\frac{\partial s_1(\mathbf{p})}{\partial p_1} + (p_2 - c_2)\frac{\partial s_2(\mathbf{p})}{\partial p_1} = 0$$

Need both FOC associated to p_1^* and p_2^* to form a system of equation, we can rewrite as:

$$\begin{pmatrix} s_1(\mathbf{p}) \\ s_2(\mathbf{p}) \end{pmatrix} + \begin{pmatrix} \frac{\partial s_1(\mathbf{p})}{\partial p_1} & \frac{\partial s_2(\mathbf{p})}{\partial p_1} \\ \frac{\partial s_1(\mathbf{p})}{\partial p_2} & \frac{\partial s_2(\mathbf{p})}{\partial p_2} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = 0$$

Firm 3 and 4's FOC are unchanged, but their prices are modified

Equilibrium after the merger with partial collusion

Collusion modeled as in Miller & Weinberg, with a parameter of profit internalization κ

When setting p_1 and p_2 the merged entity maximize:

$\Pi_{12}^c = \Pi_1 + \Pi_2 + \kappa\Pi_3 = (p_1 - c_1)s_1(\mathbf{p}) + (p_2 - c_2)s_2(\mathbf{p}) + \kappa(p_3 - c_3)s_3(\mathbf{p})$

FOC associated to p_1^*:

$$s_1(\mathbf{p}) + (p_1 - c_1)\frac{\partial s_1(\mathbf{p})}{\partial p_1} + (p_2 - c_2)\frac{\partial s_2(\mathbf{p})}{\partial p_1} + \kappa(p_3 - c_3)\frac{\partial s_3(\mathbf{p})}{\partial p_1} = 0$$

The firm 3 colluding maximizes:

$\Pi_3^c = \Pi_3 + \kappa(\Pi_1 + \Pi_2) = \kappa(p_1 - c_1)s_1(\mathbf{p}) + \kappa(p_2 - c_2)s_2(\mathbf{p}) + (p_3 - c_3)s_3(\mathbf{p})$

Which gives a FOC associated to p_3^*:

$$s_3(\mathbf{p}) + \kappa(p_1 - c_1)\frac{\partial s_1(\mathbf{p})}{\partial p_3} + \kappa(p_2 - c_2)\frac{\partial s_2(\mathbf{p})}{\partial p_3} + (p_3 - c_3)\frac{\partial s_3(\mathbf{p})}{\partial p_3} = 0$$

Equilibrium after the merger with partial collusion

These form a system of equation:

$$\begin{pmatrix} s_1(\mathbf{p}) \\ s_2(\mathbf{p}) \\ s_3(\mathbf{p}) \end{pmatrix} + \begin{pmatrix} \frac{\partial s_1(\mathbf{p})}{\partial p_1} & \frac{\partial s_2(\mathbf{p})}{\partial p_1} & \kappa\frac{\partial s_3(\mathbf{p})}{\partial p_1} \\ \frac{\partial s_1(\mathbf{p})}{\partial p_2} & \frac{\partial s_2(\mathbf{p})}{\partial p_2} & \kappa\frac{\partial s_3(\mathbf{p})}{\partial p_2} \\ \kappa\frac{\partial s_1(\mathbf{p})}{\partial p_3} & \kappa\frac{\partial s_2(\mathbf{p})}{\partial p_3} & \frac{\partial s_3(\mathbf{p})}{\partial p_3} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = 0$$

Rk: under the logit model, $\partial s_j/\partial p_k = -\alpha s_j s_k$ for $k \neq j$
$\partial s_j/\partial p_j = \alpha s_j(1 - s_j)$