Cournot Competition with Idiosyncratic Taxation

Nate Dwyer and Sandro Lera

February 6, 2019

1 Model Definition

We simulate an economy of buyers and sellers as an extension of the Cournot competition model, with additional buyer-seller specific ('idiosyncratic') transaction cost ('tax'). The goal of this tax is to avoid market dominance by one single seller based on initially only small differences in quality. Instead, the model is aimed at striking a 'fair' balance between market competition and seller protection.

The 'geography' of our model is one-dimensional with periodic boundaries, i.e. a topological circle, of length L. Due to the periodic boundary conditions, we measure physical distances with the metric $\min(|x-y|, L-|y-x|)$ for any two points ('agents') x,y on L. Without loss of generality, we set $L=1$. Our model does not rely on the implemented geography, and any other, potentially more realistic topology could be considered.

1.1 Buyers

On that circle we place some $|B|$ buyers randomly distributed. The buyers all have an identical demand curve $P+Q=a$, and each buyer $b \in B$ buys a quantity $\vec{b}$, where b_s is the amount b buys from the seller s. Each buyer tries to buy as much total quantity as they can while keeping $P+Q \leq a$ i.e. they try to maximize $\sum_{i=0}^{|S|} b_s$.

1.2 Sellers

On that circle we will also find some $|S|$ sellers, who all offer the same product, but at potentially different prices and in potentially different quantities. We denote the prices that are offered by a vector $\vec{p}$ and the quantities by the vector $\vec{q}$, where the seller s offers q_s amount of the product at a price p_s. Each seller chooses q_s and p_s simultaneously in an attempt to maximize profit.

1.3 Idiosyncratic tax

For any buyer b and seller s, we define a product-distance $D = D(b,s)$ such that the seller s that is physically closest to b has distance $D(b,s) = 1$, the next closest has distance $D = 2$, and so on.

If buyer b buys from seller s, the total price, including transaction cost (tax), is given by $p(b,s) = p_s \cdot D(b,s)^\gamma$ where γ is the scaling coefficient of the tax.

1.4 Profit Function

The profit function for seller s reads

$$\Pi_s = p_s q_s^{sold} - c q_s, \tag{1}$$

where c is the price to produce each unit of product, and q_s^{sold} is the quantity the seller s sells (which may be less but never more than the quantity q_s that was produced). Below, we then introduce the profit vector $\vec{\pi} = (\pi_1, \dots, \pi_{|S|})$ to find the Nash-equilibrium based on the profit functions of all sellers.

2 Mathematical Representation

We construct a set of buyers B and a set of sellers S, and transaction cost function $t : (B \times S) \to \mathbb{R}$, which has subfunctions $t_b : S \to \mathbb{R}$ s.t. for all $s \in S$, $t(b, s) = t_b(s)$. I don't see what are these sub-functions for. The way I see it, a specific customer-seller tax is a mapping from $B \times S$ to $\mathbb{R}$. There is no subfunction. NATE Response: t_b is used in the Algorithm part.

Also, we construct an demand curve for all $b \in B$, namely $P = a - Q$.

3 Function to put into Gambit

We construct a function $\Pi : (\mathbb{R} \times \mathbb{R})^{|S|} \to \mathbb{R}^{|S|}$ s.t. $\Pi(\vec{s}) = \vec{\pi}$, where the elements of $\vec{s}$ are the tuples (q_s, p_s) listing the quantity produced and the price offered by the seller s, and the elements of $\vec{\pi}$ are π_s, the profits of the seller s.

Given $\vec{s}$, $\vec{\pi}$ is calculated by first calculating $\vec{q}^{sold}$, the vector of total quantity sold, then using it to calculate $\vec{\pi}$ using equation (1).

Calculate $\vec{q}^{sold} = \sum_{b \in B} \vec{b}$, where $\vec{b}$ represents the amount buyer b buys from each seller. You have two matrices, Q and Q^{sold}, and the same for prices. Then you have Q_{sb} instead of Q_{ji}. NATE Response: currently, I think it might be clearer with vectors. If we do go to matrices, then we should probably go there for price as well, and we can get rid of the subfunctions that way.

3.1 Algorithm to calculate $\vec{q}^{sold}$

To determine $\vec{b}$, do the following:

For each $b_s \in \vec{b}$, have it maximize the total quantity bought by b while remaining under the individual demand curve $P = a - Q$. The resulting vector that lists the amount b buys from each seller is $\vec{b}$. Subtract $\vec{b}$ from $\vec{q}^{unsold}$ to get a new $\vec{q}^{unsold}$.

1. Set $\vec{q}^{unsold} = \vec{q}$ to start.
2. For each $b \in B$:
3. Calculate b's perceived prices: $t_b(s_s) * p_s = \vec{p}^b$.
4. Turn $\vec{p}^b$ into a $|S|$ by 2 matrix M with the first column being the price, $\vec{p}$ the second column being the remaining quantity, $\vec{q}^{unsold}$.
5. Sort M's rows by the perceived price column, and re-index the rows $0, 1, \cdots, i, \cdots, |S|$.
6. Find the maximal index i and quantity $q_i^* \leq q_i^{unsold}$ such that $p_i + q_i^* + \sum_{k=0}^{i} q_k \leq a$.
7. Now we have a vector $\vec{b}'$ of amount bought by b : $\vec{b}' =< q_0, q_1, \cdots q_{j-1}, q_i^*, 0, 0, \cdots, 0 >$.
8. Re-index this new vector using the seller order instead of the price index to get $\vec{b}$.
9. Set $\vec{q}^{unsold} = \vec{q}^{unsold} - \vec{b}$.

10. Go to step 3.

11. Calculate $\vec{q}^{sold} = \sum_{b \in B} \vec{b}$.

12. Calculate $\vec{\Pi} = \vec{p}\vec{q}^{sold} - c\vec{q}$,

4 Open Questions

- With this model, we want to demonstrate that initially small differences in quality do not lead to proportional feedback mechanisms that lead to an unequal output distribution So it might be questionable if the current Cournot model is the model of our interests, because it assumes all buyers set the same price.