Introduction to structure of dynamic oligopoly models

- Consider a simple two-firm model, and assume that all the dynamics are deterministic.

- Let x_{1t}, x_{2t}, denote the state variables for each firm in each period. Let q_{1t}, q_{2t} denote the control variables. Example: x's are capacity levels, and q's are incremental changes to capacity in each period.

- Assume (for now) that $x_{it+1} = g(x_{it}, q_{it})$, $i = 1, 2$, so that next period's state is a *deterministic* function of this period's state and control variable. (Can allow for cross-effects with no problem.)

- Firm i (=1,2) chooses a sequence $q_{i1}, q_{i2}, q_{i3}, \ldots$ to maximize its discounted profits:

$$\sum_{t=0}^{\infty} \beta^t \Pi \left(x_{1t}, x_{2t}, q_{1t}, q_{2t}\right)$$

  where $\Pi(\cdots)$ denotes single-period profits.

- Because the two firms are duopolists, and they must make these choices recognizing that their choices can affect their rival's choices. We want to consider a dynamic equilibrium of such a model, when (roughly speaking) each firm's sequence of q's is a "best-response" to its rival's sequence.

- A firm's strategy in period t, q_{it}, can potentially depend on the whole "history" of the game ($\mathcal{H}_{t-1} \equiv \{x_{1t'}, x_{2t'}, q_{1t'}, q_{2t'}\}_{t'=0,\ldots,t-1}$), and well as on the time period t itself. This becomes quickly intractable, so we usually make some simplifying regularity conditions:

  - Firms employ *stationary* strategies: so that strategies are not explicitly a function of time t (i.e. they depend on time only indirectly, through the history $\mathcal{H}_{t-1}$). Given stationarity, we will drop the t subscript, and use primes $'$ to denote next-period values.

  - A dimension-reducing assumption is usually made: for example, we might assume that q_{it} depends only on x_{1t}, x_{2t}, which are the "payoff-relevant" state variables which directly affect firm i's profits in period i. This is usually called a "Markov" assumption. With this assumption $q_{it} = q_i(x_{1t}, x_{2t})$, for all t.
  - Furthermore, we usually make a *symmetry* assumption, that each firm employs an identical strategy assumption. This implies that $q_1(x_{1t}, x_{2t}) = q_2(x_{2t}, x_{1t})$.

- To characterize the equilibrium further, assume we have an equilibrium strategy function $q^*(\cdot, \cdot)$. For each firm i, then, and at each state vector x_1, x_2, this optimal policy must satisfy Bellman's equation, in order for the strategy to constitute subgame-perfect behavior:

$$q^*(x_1, x_2) = \text{argmax}_q \left\{ \Pi(x_1, x_2, q, q^*(x_2, x_1)) + \beta V(x_1' = g(x_1, q), x_2' = g(x_2, q^*(x_2, x_1))) \right\} \tag{1}$$

  from firm 1's perspective, and similarly for firm 2. $V(\cdot, \cdot)$ is the value function, defined recursively at all possible state vectors x_1, x_2 via the Bellman equation:

$$V(x_1, x_2) = \max_q \left\{ \Pi(x_1, x_2, q, q^*(x_2, x_1)) + \beta V(x_1' = g(x_1, q), x_2' = g(x_2, q^*(x_2, x_1))) \right\}. \tag{2}$$

- I have described the simplest case; given this structure, it is clear that the following extensions are straightforward:
  - Cross-effects: $x_i' = g(x_i, x_{-i}, q_i, q_{-i})$
  - Stochastic evolution: $x_i' | x_i, q_i$ is a random variable. In this case, replace last term of Bellman eq. by $E[V(x_1', x_2') | x_1, x_2, q, q_2 = q^*(x_2, x_1)]$.

    This expectation denotes player 1's *equilibrium beliefs* about the evolution of x_1 and x_2 (equilibrium in the sense that he assumes that player 2 plays the equilibrium strategy $q^*(x_2, x_1)$).
  - > 2 firms
  - Firms employ asymmetric strategies, so that $q_1(x_1, x_2) \neq q_2(x_2, x_1)$

  - ...

- Computing the equilibrium strategy $q^*(\cdots)$ consists in iterating over the Bellman equation (1). However, the problem is more complicated than the single-agent case for several reasons:

  - The value function itself depends on the optimal strategy function $q^*(\cdots)$, via the assumption that the rival firm is always using the optimal strategy. So value iteration procedure is more complicated:

    1. Start with initial guess $V^0(x_1, x_2)$
    2. If q's are continuous controls, then when strategies are asymmetric, we must solve for $q_1^0 \equiv q^0(x_1, x_2)$ and $q_2^0 \equiv q^0(x_2, x_1)$ to satisfy the system of first-order conditions (here subscripts denotes partial derivatives)

$$\begin{aligned} 0 &= \Pi_3\left(x_1, x_2, q_1^0, q_2^0\right) + \beta V_1^0\left(g\left(x_1, q_1^0\right), g\left(x_2, q_2^0\right)\right) \cdot g_2\left(x_1, q_1^0\right) \\ 0 &= \Pi_3\left(x_2, x_1, q_2^0, q_1^0\right) + \beta V_1^0\left(g\left(x_2, q_2^0\right), g\left(x_1, q_1^0\right)\right) \cdot g_2\left(x_2, q_2^0\right). \end{aligned} \tag{3}$$

    When strategies are symmetric, then $q^0(x_1, x_2) = q^0(x_2, x_1) \equiv q^0$, and system reduces to one FOC in one unknown:

$$0 = \Pi_3\left(x_1, x_2, q^0, q^0\right) + \beta V_1^0\left(g\left(x_1, q^0\right), g\left(x_2, q^0\right)\right) \cdot g_2\left(x_1, q^0\right). \tag{4}$$

    If q's are discrete, taking values $\in \mathcal{Q}$, then (for the asymmetric case):

$$\begin{aligned} q^0 &= \text{argmax}_{q \in \mathcal{Q}} \left\{\Pi\left(x_1, x_2, q, q_2^0\right) + \beta V^0\left(g\left(x_1, q\right), g\left(x_2, q_2^0\right)\right)\right\} \\ q_2^0 &= \text{argmax}_{q \in \mathcal{Q}} \left\{\Pi\left(x_2, x_1, q, q_1^0\right) + \beta V^0\left(g\left(x_2, q\right), g\left(x_1, q_1^0\right)\right)\right\}. \end{aligned} \tag{5}$$

    In the symmetric case:

$$q^0 = \text{argmax}_{q \in \mathcal{Q}} \left\{\Pi\left(x_1, x_2, q, q^0\right) + \beta V^0\left(g\left(x_1, q\right), g\left(x_2, q^0\right)\right)\right\} \tag{6}$$

    Thus, symmetry assumption helps a lot: computational problem is essentially the same as single-agent problem (except state space is expanded to include state variables of both firms).
    3. Update the next iteration of the value function:

$$V^1(x_1, x_2) = \left\{\Pi\left(x_1, x_2, q_1^0, q_2^0\right) + \beta V^0\left(g\left(x_1, q_1^0\right), g\left(x_2, q_2^0\right)\right)\right\}. \tag{7}$$

Note: this and the previous step must be done at all points (x_1, x_2) in the discretized grid. As usual, use interpolation or approximation to obtain $V^1(\cdots)$ at points not on the grid.

4. Stop when $\sup_{x_1,x_2} ||V^{i+1}(x_1, x_2) - V^i(x_1, x_2)|| \leq \epsilon$.

– "Curse of dimensionality": the dimensionality of the state vector (x_1, x_2) is equal to the number of firms. (For instance, if you want to discretize 1000 pts in one dimension, you have to discretize at 1,000,000 pts to maintain the same fineness in two dimensions!) Some recent papers provide computational methods to circumvent this problem (Pakes/McGuire, Imai/Jain/Ching). Both of these papers advocate only computing the value function at a (small) subset of the state points each iteration.

- Clearly, it is possible to extend the Hotz-Miller insights to facilitate estimation of dynamic oligopoly models, in the case where q is a discrete control. Advantage, as before, is that you can avoid numerically solving for the value function.

  Data directly tell you: the choice probabilities (distribution of $q|x_1, x_2$); state transitions: (joint distribution of $x'_1 x'_2 | x_1, x_2, q_2, q_2$)

- Recent papers on estimating dynamic oligopoly models without doing explicit value function iteration: Berry/Pakes/Ostrovsky, Bajari/Benkard/Levin, Aguirregabiria, Pesendorfer/Schmidt-Dengler.

References