ECON 512: Empirical Methods **Spring 2019**

Lecture 14: Ericson-Pakes Games

Instructor: Paul Grieco *Date: Feb 4/11*

These lecture notes are for my personal use. If you are reading them, I decided to distribute them as an experiment. There are typos and probably outright errors, if you find them please accept my apologies and report them to me.

In these notes I will introduce the Ericson-Pakes Framework, which is a workhorse model for the empirical study of dynamic games. It views the data generating process as the outcome of a Markov Perfect Equilibrium (MPE). Background material for these notes comes from,

- Ericson & Pakes (1995), "Markov-Perfect Industry Dynamics: A Framework for Empirical Work", *Review of Economic Studies.*
- Pakes & McGuire (1994), "Computing Markov Perfect Equilibria: Numerical Implications of a Dynamic Differentiated Product Model, *RAND Journal of Economics.*
- Doraszelski & Pakes (2007), "A Framework for Applied Dynamic Analysis in IO," Handbook of Industrial Organization, Chapter 30.

14.1 Introducing Dynamic Games

Why do we need these?

- Similar firms, even in the same industry, often have different "fates" (Dunne, Roberts, Samuleson 1988)
  - Gross entry and exit are both much larger than the net change in the number of firms.
  - We see observably similar firms where one is expanding and the other contracting.
- EP model give us hope to account for such heterogeneity while still making statements about market structure.
- Markov assumption is empirically convenient:
  - Our data has a finite history.
  - Allows for the estimation of agent policies by conditioning on the set of state variables.
  - Counterfactual equilibrium can be computed and simulated.

Key Assumptions (some relaxed by future work):

- State is publicly observed by all players.
- Heterogeneity arises from privately observed shocks that are i.i.d. over agents and time.
- Solution concept is Markov Perfect Equilibrium: Strategies depend on current payoff-relevant state, not histories. (Collusion?).

- Game is stationary.

What are we adding from DP?

- Stochastic endogenous transition matrix.
- Continous choice with discrete state.
- Multiple agents.

Challenges for this literature:

- Curse of dimensionality.
- States unobserved by the econometrician.
- States unobserved by the players (asymmetric information).
- Multiple Equilibria.
- Dependence on functional forms.

14.2 Basic Idea: Markov Perfect Equilibrium

- Time is discrete, and horizon is infinite.
- N firms are described by a state ω_i (cost, product quality, etc).
- The total state is described by $\omega = (\omega_1, \ldots, \omega_N, \tilde{\omega})$ where $\tilde{\omega}$ is some aggregate market state determined by nature.
- Each period, firms choose an action from a feasible set $x_i \in X_i(\omega)$, the action profile is $(x_1, \ldots, x_N)$. Market may be hit by an aggregate shock $\eta \in Y$.
- State-to-State transitions are determined by a family of distributions:

$$\mathcal{P}_{\omega_i'} \equiv \{\rho(\cdot|\omega_i, x_i, \eta) : \omega_i \in \Omega_i, x_i \in X_i, \eta \in Y\}$$

For now, let's ignore the industry aggregate state and shock. They are easy to put back in because they are exogenous. Then the state is just $(\omega_1, \ldots, \omega_N)$ and we can write the value function of firm i as,

$$V(\omega_i, \omega_{-i}) = \max_{x_i \in X_i(\omega)} E[\pi_i(\omega_i, \omega_{-i}, x_i, x_{-i}) + \beta V(\omega_i', \omega_{-i}')|\omega_i, \omega_{-i}, x_i]$$

Where,

- π_i represents profits when firm of firm i *given* opponents are following equilibrium strategy x_{-i}^*.
- Likewise, the expectation is itself an equilibrium object that assumes (a) opponents are following x_{-i}^* (b) in future firm i will follow x_i^* (principle of optimality).
- The key is that holding opponent strategies fixed, this is a single agent optimization problem, that we are able to solve. I.e., we know how to compute i's best response to opponent strategy $\tilde{x}_{-i}(\cdot)$.

Definition 14.1 *A Markov perfect equilibrium (MPE) is a strategy profile* $(x_1(\cdot), x_2(\cdot), \ldots, x_N(\cdot))$ *such that*

- *Every agent's strategy is optimal given its perceptions about future industry structure (the state), and*
- *Those perceptions are consistent with each player's equilibrium behavior (strategies).*

These conditions map into two sets of necessary conditions for equilibrium: (a) the first order conditions of the Bellman equation and (b) the Bellman equation itself substituting in optimal strategies.

Computing an MPE amounts to solving this set of equations. We will focus on doing this by iterating best responses in hopes they converge. However, the equilibrium problem is NOT a contraction map, so we have no guarantee.

14.3 Aside: Estimation without Computing Equilibrium

An interesting feature of these models is they can be estimated *without* having to be solved. Bajari, Benkard, and Levin (2007) and others provide methodologies. They are covered in the IO sequence. Here's a rough idea:

1. The key is that with the Markov assumption, policy functions (conditional choice probabilities, or CCPs) can be (nonparameterically?) estimated from the data. They are just actions conditional on states, $P(x_i|\omega)$. State transitions can also be estimated, they are $P(\omega'|\omega, x)$.

2. Once policies and transitions are estimated, we can forward simulate firm actions to determine expected paths thorough the state space. Given parameters this means we can compute the expected value of following the estimated policy.

$$V_i(\omega, \sigma_i, \sigma_{-i}) = E\left[\sum_{t=0}^{\infty} \beta^t \pi_i(\omega_t, x_t) | \omega_0 = \omega\right]$$

3. Agent optimization means the estimated policy maximizes the agents long-run utility, that is for any alternative policy $\tilde{\sigma}_i$, it must be true that, $\forall i, \omega$:

$$V_i(\omega, \sigma_i, \sigma_{-i}) \geq V_i(\omega, \tilde{\sigma}_i, \sigma_{-i})$$

   So I estimate profit parameters by minimizing the objective:

$$\frac{1}{|\Omega| \times |\tilde{\sigma}_i|} \sum_{s, \tilde{\sigma}} \min\left\{0, V(\omega, \tilde{\sigma}_i, \sigma_{-i}) - V(\omega, \sigma_i, \sigma_{-i})\right\}^2$$

While this might look complicated, if we let π be linear in parameters it can actually boil down to a very simple problem. Why? The key is you only need to do the forward simulation a single time, and so the optimization becomes a linear moment inequality problem.

Concluding thoughts:

- Usually there are many "alternative policies" and you can't include them all. Researcher must choose what to include.

- Very large games will have very large state spaces, and you might not observe every state. Making it difficult to non-parametrically estimate the CCPs.
- That said, this makes estimating several firm dynamic games feasible. Even simple dynamic games models can take quite a while to compute. Repeatedly re-calculating equilibria to compute a likelihood or moments is a non-starter for now.
- On the other hand, recovering structural parameters may be of limited interest if you can't compute counterfactuals.

14.4 Computation of Equilibrium Example: Quality Ladder Model

Consider a differentiated products industry where firms can invest in improving product quality. Each firms state is the quality of its product, there are L quality levels:

$$\Omega_i = \{1, \dots, L\}$$

The timing of the game is as follows:

1. Firms start period with state $\omega = (\omega_1, \dots, \omega_N)$.
2. Firms simultaneously name prices, consumers purchase, profits realized.
3. Firms simultaneously invest in improving their quality.
4. State variable updated based on firm investments, proceed to next period.

Note, the paper gives a version of this model that includes a discrete entry/exit decision of firms. We won't cover that extension in class, as it mostly just adds notation and one additional state ("not operating").

14.4.1 Pricing Game

Utility of firm i's product for consumer c is,

$$u_{ic} = g(\omega_i) - p_i + \varepsilon_{ic}$$

where $g(\cdot)$ governs the extent of vertical differentiation in the market place and $\varepsilon_{ic} \sim T1EV$ represents horizontal differentiation.

In the code,

$$g(\omega_i) = \begin{cases} 3\omega_i - 4 & 1 \leq \omega_i \leq 5 \\ 12 + \log(2 - \exp(16 - 3 * \omega_i)) & 6 \leq \omega_i \leq 18 \end{cases}$$

It is increasing, eventually concave, and very flat at high values of ω_i this flatness reduces the benefit of investment to make sure that the stationary distribution stays in the interior or the state space.

Consumers may choose to buy from any firm or an outside option with utility $u_{i0} = \varepsilon_{0c}$. So demand is:

$$D_i(p; \omega) = \frac{\exp(g(\omega_i) - p_i)}{1 + \sum_{j=1}^{N} \exp(g(\omega_j) - p_j)}$$

Firms choose the price of their good, p_i to maximize flow profits:

$$\pi_i(\omega) = \max_{p_i} D_i(p_i, p_{-i}; \omega)(p_i - c)$$

Note that price is fully flexible from period to period, so it has no dynamic implications (like labor in the capital growth model). Using some logit-related algebra, we can show the first order condition for firm i is:

$$1 - (1 - D_i(p_i, p_{-i}; \omega))(p_i - c) = 0$$

This gives us N equations to solve for the N unknown prices to find the pricing equilibrium for state ω. Doing so for every state gives us the flow profits for the game, $\pi_i(\omega)$ for every player, which we will henceforth treat as a known parameter.

FYI, to derive the FOC, notice:

$$\begin{aligned} \frac{\partial D_i}{\partial p_i}(p_i - c) + D_i &= 0 \\ -D_i(1 - D_i)(p_i - c) + D_i &= 0 \\ -(1 - D_i)(p_i - c) + 1 &= 0 \end{aligned}$$

14.4.2 Investment

Firms' quality levels evolve as follows:

- In each period, the firm chooses a level of investment x_i.
  - with probability $\frac{\alpha x}{1+\alpha x}$, investment is successful and quality moves up one state.
  - otherwise, quality is unchanged by investment.
- With probability δ firms suffer a "quality depreciation shock" which reduces their quality by one state.

We need the depreciation shock to keep the state in the interior of the state space, we can think of it as an improvement of the outside good.

$$\rho(\omega_i' | \omega_i, x_i) = \begin{cases} \frac{(1-\delta)\alpha x}{1+\alpha x} & \omega_i' = \omega_i + 1 \\ \frac{(1-\delta)+\delta\alpha x}{1+\alpha x} & \omega_i' = \omega_i \\ \frac{\delta}{1+\alpha x} & \omega_i' = \omega_i - 1 \\ 0 & \text{otherwise} \end{cases}$$

Notice that the transition matrix is a banded diagonal matrix. This sparsity will be very useful computationally. Also notice that transition probabilities are smooth in x, so we will be able to use first order conditions.

Boundary Conditions: In state L, you can't go up, so if you invest then and are successful you simply stay at state L, same with state 1 and going down.

14.4.3 Investment: Monopoly

To keep things simple, let's start with the monopoly problem, and then extend to a game. The Bellman equation is:

$$V(\omega_i) = \max_{x_i} \pi_i(\omega) - x_i + \beta \sum_{\omega'=1}^{L} V(\omega')\rho(\omega_i'|\omega_i, x_i)$$

The principle of optimality gives us the first order condition,

$$-1 + \beta \sum_{\omega'=1}^{L} V(\omega') \frac{\partial \rho(\omega_i'|\omega_i, x_i)}{\partial x_i} = 0$$

Or simply that the marginal cost of investment should equal its (dynamic) benefits. Using the definition of ρ we have,

$$\begin{aligned} 1 = \beta \Bigg(& V(\omega_i + 1) \frac{(1-\delta)\alpha(1+\alpha x) - (1-\delta)\alpha^2 x}{(1+\alpha x)^2} \\ & + V(\omega_i) \frac{\delta\alpha(1+\alpha x) - (1-\delta)\alpha - \delta\alpha^2 x}{(1+\alpha x)^2} \\ & + V(\omega_i - 1) \frac{\delta\alpha}{(1+\alpha x)^2} \Bigg) \end{aligned}$$

This simplifies to the quadratic condition:

$$\begin{aligned} (1+\alpha x)^2 &= \alpha\beta \left[(1-\delta)V(\omega_i+1) + (\delta - (1-\delta))V(\omega_i) - \delta V(\omega_i - 1)\right] \\ &= \alpha\beta \left[(1-\delta)(V(\omega_i+1) - V(\omega)) + \delta(V(\omega_i) - V(\omega_i - 1))\right] \end{aligned}$$

So the optimal investment level is the positive solution to this equation is a closed form expression:

$$x^*(\omega) = \max \left\{ 0, \frac{-1 + \sqrt{\alpha\beta \left[(1-\delta)(V(\omega_i+1) - V(\omega)) + \delta(V(\omega_i) - V(\omega_i - 1))\right]}}{\alpha} \right\}$$

We also know the Bellman equation must hold at the optimum so,

$$V(\omega_i) = \pi(\omega_i) - x^*(\omega_i) + \beta \sum_{\omega'=\omega_i-1}^{\omega_i+1} V(\omega')p(\omega'|\omega_i, x^*(\omega_i))$$

This gives us $2L$ equations that are needed to solve for $2L$ unknowns, $(V(\cdot), x^*(\cdot))$. Even better, we know this problem to be a contraction. In practice we have several options for how to solve this problem.

14.4.3.1 Value Function Iteration (an old favorite)

Since the solution to the Bellman given $V(\cdot)$ has a closed form, this is particularly straightforward:

1. Given V^{r-1}, use FOC to calculate $x^r(\cdot)$.
2. Substitute $x^j(\cdot)$ into Bellman to find V^r.
3. If $||V^j - V^{j-1}|| < \varepsilon_v$ and $||x^j - x^{j-1}|| < \varepsilon_x$, stop, otherwise go to 1.

14.4.3.2 Policy Function Iteration

- VFI uses optimal policy for one period, and otherwise "re-uses" the old policy.
- PFI uses the optimal policy over the entire horizon, because bellman is linear in the policy function, this is easy to do.
- Pro: Converges in many fewer iterations.
- Con: Each iteration takes longer, because it involves a matrix inverse.

Note we can write the Bellman equation in matrix notation:

$$V = \pi - x + \beta P(x)V$$

. Where V, π, x^* are all $K \times 1$ and $P(x)$ is $K \times K$ (banded diagonal). Now we just solve for V:

$$V = (I - \beta P(x))^{-1}(\pi - x)$$

This is the value of following policy function x over the infinite horizon.

Algorithm:

1. Given V^{j-1}, use FOC to calculate $x^j(\cdot)$.
2. $V^j = (I - \beta P(x))^{-1}(\pi - x^j)$
3. If $||V^j - V^{j-1}|| < \varepsilon_v$ and $||x^j - x^{j-1}|| < \varepsilon_x$, stop, otherwise go to 1.

14.4.3.3 Newton's Method

One can attempt to solve this problem directly using a non-linear solver. E.g., trying to solve this system of equations using fmincon, KNITRO, or another constrained optimization pakage. We discussed this idea earlier. In practice, you will need to program up derivative information (not too hard) and also have a very good start point (harder).

14.4.3.4 Homotopy

Homotopy methods can be combined with Newton's method to relieve the need for a good start point. It can be particularly useful when you want to solve the game with many different parameterizations or search for multiple equilibria. Well talk about it more (if time) next week.

14.4.4 Investment: Duopoly

What changes when we go to multiple players to compute equilibrium of a game?

- Many more states, in in principle L^N (but these are reduced via exchangeability).
- Transition matrix depends on opponent strategies, expectations need to integrate over opponent strategies.
- Equilibrium is NOT a contraction due to updating opponent strategy.
- Existence can be proven under common conditions, but uniqueness is much harder and often doesn't hold.

14.4.4.1 Exchangability

Typically we assume that firms play a symmetric MPE, this means that a firms strategy depends only on its current state, but not its identity. In other words, we look for equilibrium in exchangeable strategies. For two players, exchangeability means:

$$x_1(\omega^i, \omega^j) = x_2(\omega^j, \omega^i)$$

Or more generally, using game theoretic notation:

$$f(\omega_i, \omega_{-i}) = f(\omega_i, \omega_{\pi(-i)})$$

Where $\pi(\cdot)$ represents any permutation of opponents.

Using symmetry/exchangeability:

- Reduces the number of variables we need to solve for, since all players have the same strategy.
- Means we can simply permute player i's strategy to find player j's strategy, simplifying computation.

In practice, exchangeability can pose some challenges? What if we want to allow for something like firm fixed effects or covariates? Then exchangeability is not really all that useful.

14.4.4.2 Bellman Equation and Optimal Policy Given Opponent's Strategy

Given Exchangability, the Bellman equation for the duopoly problem is:

$$V(\omega_1, \omega_2) = \max_{x \geq 0} \left\{ \pi(\omega) - x + \beta \sum_{\omega_1'} \sum_{\omega_2'} V(\omega_1', \omega_2') P(\omega_1' | \omega_1, \underbrace{x}_{\text{Choice}}) P(\omega_2' | \omega_2, \underbrace{x_2(\omega)}_{\text{Opp Strategy}}) \right\}$$

However, if we integrate over opponent strategy, we can construct player 1's continuation value conditional on firm 2's strategy as:

$$W(\omega_1'; \omega) = \sum_{\omega_2'} V(\omega_1', \omega_2') P(\omega_2' | \omega_2, x_2(\omega))$$

We can replace this in the Bellman to get:

$$V(\omega_1, \omega_2) = \max_{x \geq 0} \left\{ \pi(\omega) - x + \beta \sum_{\omega_1'} W(\omega_1'; \omega) P(\omega_1' | \omega_1, x) \right\}$$

Since $W(\cdot)$ does not depend on x, we are essentially back to the monopoly problem. This makes since, the solution to this problem is player 1's best response to player 2 playing $x_2(\omega)$. The solution to this problem is:

$$x_1(\omega) = \max \left\{ 0, \frac{-1 + \sqrt{\alpha\beta \left[(1-\delta)(W(\omega_1 + 1; \omega) - W(\omega_1; \omega)) + \delta(W(\omega_1) - W(\omega_1 - 1; \omega)) \right]}}{\alpha} \right\}$$

14.4.4.3 Value Function Iteration

So value function iteration proceeds as follows, given we have an initial guess of V^{j-1} and x^{j-1}, which are firm 1's value function and policy function, respectively:

1. Using exchangeability assumption, $x_2(\omega) = x^{j-1}(\omega_2, \omega_1)$.
2. Calculate firm 1 continuation value given firm 2's strategy: $W(\omega_1'; \omega)$.
3. Use $W(\cdot)$ to calculate firm 1's best response, $x^j(\omega)$.
4. Use x^j and W in Bellman equation to calculate firm 1's value function V^j.
5. Check for convergence of Value and Policy functions.

However, there is no guarantee of convergence. For this problem. The updating of the policy function (in addition to the value function) is what breaks the contraction property. In particular it is possible for the iterations to fall into "cycles" where two (non-symmetric) strategies are best responses to each-other. To avoid this, Doraszelski and Pakes recommend "dampening" which uses a convex combination of previous strategies instead of full updating:

$$x^j = (1-\lambda)x^{j-1} + \lambda x^j$$
$$V^j = (1-\lambda)V^{j-1} + \lambda V^j$$

Where $\lambda \in (0,1)$, typically close to 1.