MAU 34804 Lecture 24

John Stalker

Trinity College Dublin

20 March 2020

The chicken and egg economy

Imagine an economy with only two goods, eggs and chickens, and two industries, laying and hatching.
The laying industry takes chickens as an input and produces eggs as an output. More precisely, 1 chicken can produce 100 eggs. Equivalently, it takes 0.01 chickens to produce an egg.
The hatching industry takes eggs as an input and produces chickens as an output. It takes 1 egg to produce 1 chicken.
The input-output matrix is

$$A = \begin{pmatrix} 0 & 0.01 \\ 1 & 0 \end{pmatrix}.$$

As you can see, the entries of this matrix are non-negative. There is no positive power A^m for which all entries are positive, so the Perron theory doesn't apply directly.

Undeterred, let's consider the action of A on

$$\Delta_{\mathbf{w}} = \left\{\mathbf{p} \in \mathbf{R}^2 \colon p_1 \geq 0, p_2 \geq 0, w_1 p_1 + w_2 p_2 = 1\right\}$$

by $f_{\mathbf{w}}(\mathbf{p}) = \mathbf{q}$, where

$$q_1 = \frac{0.01 p_2}{0.01 w_1 p_2 + 1 w_2 p_1}, \quad q_2 = \frac{1 p_1}{0.01 w_1 p_2 + 1 w_2 p_1}.$$

The fixed point $f_{\mathbf{w}}(\mathbf{b}) = \mathbf{b}$ is

$$\mathbf{b} = \begin{pmatrix} 1/(w_1 + 10 w_2) \\ 10/(w_1 + 10 w_2) \end{pmatrix}$$

It's an eigenvector of A with eigenvalue 0.1.
This isn't a very good model of poultry farming. It first appeared in a paper of Kuhn, as a counter-example, disproving a "turnpike theorem" which Gale claimed to have proved.

Frog-centipede-snake and the minimax theorem

Consider a two person zero sum game with three pure strategies for each player, so $m = n = 3$, and payoff matrix

$$A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

This corresponds to a number of popular hand gesture games. The earliest recorded name is Shǒushì Lìng, but the game is currently best known as Rock-paper-scissors, after the three gestures which replaced the original gestures for frog, centipede and snake as the game made its way to English speaking countries.

Applying the minimax theorem

$$\Delta_P = \left\{\mathbf{p} \in \mathbf{R}^3 \colon p_1, p_2, p_3 \geq 0, p_1 + p_2 + p_3 = 1\right\},$$
$$\Delta_Q = \left\{\mathbf{q} \in \mathbf{R}^3 \colon q_1, q_2, q_3 \geq 0, q_1 + q_2 + q_3 = 1\right\},$$

$$\mu_P(\mathbf{q}) = \max(q_2 - q_3, q_3 - q_1, q_1 - q_2)$$
$$\mu_Q(\mathbf{p}) = \min(p_3 - p_2, p_1 - p_3, p_2 - p_1)$$

$$P(\mathbf{q}) = \{\mathbf{p} \in \Delta_P \colon \forall i \in \{1,2,3\} \colon q_{i+1} - q_{i+2} < \mu_P(\mathbf{q}) \Rightarrow p_i = 0\}$$
$$Q(\mathbf{p}) = \{\mathbf{q} \in \Delta_Q \colon \forall i \in \{1,2,3\} \colon p_{i+2} - p_{i+1} > \mu_Q(\mathbf{p}) \Rightarrow q_i = 0\}$$

where we've used the wrap-around convention $p_4 = p_1$, $p_5 = p_2$, $q_4 = q_1$, $q_5 = q_2$. Clearly

$$\max_{\mathbf{p} \in \Delta_P} \min_{\mathbf{q} \in \Delta_Q} f(\mathbf{p}; \mathbf{q}) = 0 = \min_{\mathbf{q} \in \Delta_Q} \max_{\mathbf{p} \in \Delta_P} f(\mathbf{p}; \mathbf{q})$$

This is attained only at the point

$$(\mathbf{p}^*; \mathbf{q}^*) = (1/3, 1/3, 1/3; 1/3, 1/3, 1/3).$$

The Ultimatum Game

Unlike the previous example, the Ultimatum Game exists only as an illustrative example or as an academic experiment, but it is illustrative of a real world phenomenon. There are various formulations, but here we'll use a zero sum version with three players. Player A chooses a number from the set $M = \{0, 1, \ldots, m\}$ and announces this choice to Players B and C. Player B then chooses to say either "accept" or "reject". Player C then has no choice but to pay $m - j$ euro to Player A and j euro to Player B if Player A chose j and Player B accepted. If Player B rejected then there are no payments.

Strategy sets

This might seem like a similar game to the two person zero sum game considered earlier, with $m+1$ pure strategies available to Player A, 2 strategies available to Player B, and only one strategy available to poor Player C. That's correct as far as Players A and C are concerned. Their strategy sets are the m-simplex

$$S_A = \Delta_P = \left\{ \mathbf{p} \in \mathbf{R}^{m+1} \colon p_i \geq 0, \sum_{i \in M} p_i = 1 \right\}$$

and the 0-simplex (single point)

$$S_C = \Delta_R = \left\{ r \in \mathbf{R} \colon r \geq 0, r = 1 \right\},$$

as expected. For Player B things are more complicated, because Player B already knows not just what mixed strategy Player A has chosen but also what pure strategy they have selected, and can adapt their own mixed strategy to this.

Player B's strategy set, utilities

What Player B needs is a probability q_j of accepting an offer of j from Player A. Their strategy set is therefore

$$S_B = [0,1]^{m+1} = \left\{\mathbf{q} \in \mathbf{R}^{m+1} \colon 0 \le q_j \le 1\right\}$$

Note that there is no reason for the q's to sum to 1. The utility functions for the the players are the expected net payments they receive:

$$u_A(\mathbf{p};\mathbf{q};r) = \sum_{j=0}^{m}(m-j)p_j q_j,$$

$$u_B(\mathbf{p};\mathbf{q};r) = \sum_{j=0}^{m} j p_j q_j,$$

$$u_C(\mathbf{p};\mathbf{q};r) = -\sum_{j=0}^{m} m p_j q_j.$$

b_i and B_i

$$b_A(\mathbf{q}; r) = \max_{0 \le j \le m} (m - j) q_j,$$

$$B_A(\mathbf{q}; r) = \left\{ \mathbf{p} \in \Delta_P \colon \forall i \in M \colon (m - i) q_i < \max_{0 \le j \le m} (m - j) q_j \Rightarrow p_i = 0 \right\}.$$

$$b_B(\mathbf{p}; r) = \sum_{j=0}^{m} j p_j = \sum_{j=1}^{m} j p_j$$

$$B_B(\mathbf{p}; r) = \left\{ \mathbf{q} \in [0, 1]^{m+1} \colon \forall i \in M - \{0\} \colon p_i > 0 \Rightarrow q_i = 1 \right\}.$$

$$b_C(\mathbf{p}; \mathbf{q}) = -\sum_{j=0}^{m} m p_j q_j, \quad B_C(\mathbf{p}; \mathbf{q}) = \{1\}.$$

Nash Equilibria

There are loads of Nash Equilibria. To be more precise, the set of Nash equilibria forms the polyhedron of an $m + 1$ dimensional simplicial complex. Most of them are fairly stupid. For example, with $m = 2$, the following are equilibria:

$$(\mathbf{p}^*; \mathbf{q}^*; r) = (1, 0, 0; 1, 1, 1; 1),$$

$$(\mathbf{p}^*; \mathbf{q}^*; r) = (0, 0, 1; 0, 0, 1; 1),$$

$$(\mathbf{p}^*; \mathbf{q}^*; r) = (0, 1, 0; 0, 1, 0; 1).$$

The Nash equilibrium conditions is *necessary* for a set of strategies to be sensible, but it's clearly not sufficient.