Notebook

August 10, 2017

1 Support enumeration

The definition implies that a Nash equilibrium is a pair of best responses.

We can use this and the best response condition of the previous chapter to find Nash equilibrium.

1.1 Definition of support

For a given strategy σ, the support of σ: $\mathcal{S}(\sigma)$ is the set of strategies for which $\sigma_i > 0$:

$$i \in \mathcal{S}(\sigma) \Leftrightarrow \sigma_i > 0$$

For example:

- If $\sigma = (1/3, 1/2, 0, 0, 1/6)$: $\mathcal{S}(\sigma) = \{1, 2, 5\}$
- If $\sigma = (0, 0, 1, 0)$: $\mathcal{S}(\sigma) = \{3\}$

```
In [4]: import numpy as np
        sigma = np.array([1/3, 1/2, 0, 0, 1/6])
        np.where(sigma > 0)  # Recall Python indexing starts at 0

Out[4]: (array([0, 1, 4]),)

In [5]: sigma = np.array([0, 0, 1, 0])
        np.where(sigma > 0)  # Recall Python indexing starts at 0

Out[5]: (array([2]),)
```

1.2 Definition of nondegenerate games

A two player game is called nondegenerate if no mixed strategy of support size k has more than k pure best responses.

For example, the following game is degenerate:

$$A = \begin{pmatrix}1 & 1 & 0\\ 2 & 3 & 0\end{pmatrix} \qquad B = \begin{pmatrix}1/2 & -1 & -1/2\\ -1 & -1 & 2\end{pmatrix}$$

Indeed, consider $\sigma_c = (0, 0, 1)$, we have $|\mathcal{S}(\sigma_c)| = 1$ and:

$$A\sigma_c^T = \begin{pmatrix}0\\ 0\end{pmatrix}$$

So the number of pure best responses to σ_c is 2.
Thus the game considered is indeed degenerate.

```
In [6]: A = np.array([[1, 1, 0], [2, 3, 0]])
        sigma_c = np.array([0, 0, 1])
        (np.dot(A, sigma_c))
```

```
Out[6]: array([0, 0])
```

This leads to the following algorithm for identifying Nash equilibria:

1.3 Support enumeration algorithm

For a nondegenerate 2 player game $(A, B) \in \mathbb{R}^{m\times n^2}$ the following algorithm returns all nash equilibria:

1. For all $1 \leq k \leq \min(m, n)$;
2. For all pairs of support (I, J) with $|I| = |J| = k$
3. Solve the following equations (this ensures we have best responses):

$$\sum_{i\in I}\sigma_{ri}B_{ij} = v \text{ for all } j \in J$$

$$\sum_{j\in J}A_{ij}\sigma_{cj} = u \text{ for all } i \in I$$

4. Solve

- $\sum_{i=1}^m \sigma_{ri} = 1$ and $\sigma_{ri} \geq 0$ for all i
- $\sum_{j=1}^n \sigma_{ci} = 1$ and $\sigma_{cj} \geq 0$ for all j

5. Check the best response condition.

Repeat steps 3,4 and 5 for all potential support pairs.

1.4 2 by 2 example of support enumeration

As an example consider the matching pennies game.

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

1. Consider $k = 1$: so here we are just considering supports of size 1, in other words pairs of pure best responses. The easiest way to identify these is by looking at the best responses:

$$A = \begin{pmatrix} \underline{1} & -1 \\ -1 & \underline{1} \end{pmatrix} \qquad B = \begin{pmatrix} -1 & \underline{1} \\ \underline{1} & -1 \end{pmatrix}$$

So there are no pairs.

1. Thus we start again with $k = 2$.
2. There is only one pair of best responses to be considered: $I = J = \{1, 2\}$.
3. The equations we need to solve are:

$$-\sigma_{r1} + \sigma_{r2} = v$$

$$\sigma_{r1} - \sigma_{r2} = v$$

and

$$\sigma_{c1} - \sigma_{c2} = u$$

$$-\sigma_{c1} + \sigma_{c2} = u$$

We don't actually care (or know!) the values of u, v so we in fact solve:

$$-\sigma_{r1} + \sigma_{r2} = \sigma_{r1} - \sigma_{r2}$$

$$\sigma_{c1} - \sigma_{c2} = -\sigma_{c1} + \sigma_{c2}$$

which gives:

$$\sigma_{r1} = \sigma_{r2}$$

$$\sigma_{c1} = \sigma_{c2}$$

4. This gives:

$$\sigma_r = (1/2, 1/2)$$

$$\sigma_c = (1/2, 1/2)$$

5. Finally we check the best response condition: (we already did this in the previous chapter).

Note that for 2 player games with $m = n = 2$ step 5 is trivial so in fact to find best mix strategy Nash equilibrium for games of this size simply reduces to finding a solution to 2 linear equations (step 3).

Let us consider a large game:

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 2 & -1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 1/2 & -1 & -1/2 \\ -1 & 3 & 2 \end{pmatrix}$$

1. It is immediate to note that there are no pairs of pure best responses.
2. All possible support pairs are:

- $I = (1, 2)$ and $J = (1, 2)$
- $I = (1, 2)$ and $J = (1, 3)$
- $I = (1, 2)$ and $J = (2, 3)$

3. Let us solve the corresponding linear equations:

- $I = (1, 2)$ and $J = (1, 2)$:

$$1/2\sigma_{r1} - \sigma_{r2} = -\sigma_{r1} + 3\sigma_{r2}$$
$$\sigma_{r1} = 8/3\sigma_{r2}$$

$$\sigma_{c1} + \sigma_{c2} = 2\sigma_{c1} - \sigma_{c2}$$
$$\sigma_{c1} = 2\sigma_{c2}$$

- $I = (1, 2)$ and $J = (1, 3)$:

$$1/2\sigma_{r1} - \sigma_{r2} = -1/2\sigma_{r1} + 2\sigma_{r2}$$
$$\sigma_{r1} = 3\sigma_{r2}$$

$$\sigma_{c1} - \sigma_{c3} = 2\sigma_{c1} + 0\sigma_{c3}$$
$$\sigma_{c1} = -\sigma_{c3}$$

- $I = (1, 2)$ and $J = (2, 3)$:

$$-\sigma_{r1} + 3\sigma_{r2} = -1/2\sigma_{r1} + 2\sigma_{r2}$$
$$\sigma_{r1} = 2\sigma_{r2}$$

$$\sigma_{c2} - \sigma_{c3} = -\sigma_{c2} + 0\sigma_{c3}$$
$$2\sigma_{c2} = \sigma_{c3}$$

4. We check which supports give valid mixed strategies:

- $I = (1, 2)$ and $J = (1, 2)$:

$$\sigma_r = (8/11, 3/11)$$
$$\sigma_c = (2/3, 1/3, 0)$$

- $I = (1, 2)$ and $J = (1, 3)$:

$$\sigma_r = (3/4, 1/4)$$
$$\sigma_c = (k, 0, -k)$$

which is not a valied mixed strategy.

- $I = (1, 2)$ and $J = (2, 3)$:

$$\sigma_r = (2/3, 1/3)$$
$$\sigma_c = (0, 1/3, 2/3)$$

5. Let us verify the best response condition:

- $I = (1, 2)$ and $J = (1, 2)$:

$$\sigma_c = (2/3, 1/3, 0)$$

$$A\sigma_c^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Thus σ_r is a best response to σ_c

$$\sigma_r = (8/11, 3/11)$$
$$\sigma_r B = (1/11, 1/11, 2/11)$$

Thus σ_c is not a best response to σ_r (because there is a better response outside of the support of σ_c).

- $I = (1, 2)$ and $J = (2, 3)$:

$$\sigma_c = (0, 1/3, 2/3)$$

$$A\sigma_c^T = \begin{pmatrix} -1/3 \\ -1/3 \end{pmatrix}$$

Thus σ_r is a best response to σ_c

$$\sigma_r = (2/3, 1/3)$$
$$\sigma_r B = (0, 1/3, 1/3)$$

Thus σ_c is a best response to σ_r.

Thus the (unique) Nash equilibrium for this game is:

$$((2/3, 1/3), (0, 1/3, 2/3))$$

Note that we can confirm all of this using `nashpy` (which by default uses support enumeration):Note that we can confirm all of this using `nashpy` (which by default uses support enumeration):

```
In [7]: import nash
        A = np.array([[1,-1], [-1, 1]])
        game = nash.Game(A)
        list(game.support_enumeration())

Out[7]: [(array([ 0.5,  0.5]), array([ 0.5,  0.5]))]
```

```
In [8]: A = np.array([[1, 1, -1], [2, -1, 0]])
        B = np.array([[1/2, -1, -1/2], [-1, 3, 2]])
        game = nash.Game(A, B)
        list(game.support_enumeration())

Out[8]: [(array([ 0.66666667,  0.33333333]),
          array([-0.        ,  0.33333333,  0.66666667]))]
```

If you recall the degenerate game mentioned previously:

```
In [9]: A = np.array([[1, 1, 0], [2, -1, 0]])
        B = np.array([[1/2, -1, -1/2], [-1, 3, 2]])
        game = nash.Game(A, B)
        list(game.support_enumeration())

Out[9]: []
```

This result is given without proof: