Notebook

August 10, 2017

1 The Lemke Howson algorithm

The vertex and support enumeration algorithms are all algorithms that use an exhaustive search. For large games, this can take a long time and/or have a high computational cost. The following algorithm gives an approach to create a path through vertices in both best response polytopes to find a pair that is fully labelled.

As an example let us consider the matching pennies game:

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

First let us add 2 to all utilities:

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

The vertices for $\mathcal{P}$ are:

1. $a = (0, 0)$ has labels $\{0, 1\}$
2. $b = (1/3, 0)$ has labels $\{1, 3\}$
3. $c = (1/4, 1/4)$ has labels $\{2, 3\}$
4. $d = (0, 1/3)$ has labels $\{0, 3\}$

The vertics and labels for $\mathcal{Q}$ are:

1. $w = (0, 0)$ has labels $\{2, 3\}$
2. $x = (1/3, 0)$ has labels $\{0, 3\}$
3. $y = (1/4, 1/4)$ has labels $\{0, 1\}$
4. $z = (0, 1/3)$ has labels $\{1, 2\}$

Let us apply the algorithm:

- (a, w) have labels: $\{0, 1\}, \{2, 3\}$. Drop 0 (arbitrary decision) in $\mathcal{P}$.
- $\to (b, w)$ have labels: $\{1, 3\}, \{2, 3\}$. In $\mathcal{Q}$ drop 3.
- $\to (b, z)$ have labels: $\{1, 3\}, \{1, 2\}$. In $\mathcal{P}$ drop 1.
- $\to (c, z)$ have labels: $\{2, 3\}, \{1, 2\}$. In $\mathcal{Q}$ drop 2.
- $\to (c, y)$ have labels: $\{2, 3\}, \{0, 1\}$. Fully labeled vertex pair.

We now return the strategy pair by normalising these vertices:

$$((1/2, 1/2), (1/2, 1/2))$$

This is also implemented in `Nashpy`:

```
In [1]: import numpy as np
        import nash
        A = np.array([[1, -1], [-1, 1]])
        matching_pennies = nash.Game(A)
        matching_pennies.lemke_howson(initial_dropped_label=0)
```

```
Out[1]: (array([ 0.5,  0.5]), array([ 0.5,  0.5]))
```

You can also iterate over all possible starting labels:

```
In [2]: for eq in matching_pennies.lemke_howson_enumeration():
            print(eq)
```

```
(array([ 0.5,  0.5]), array([ 0.5,  0.5]))
(array([ 0.5,  0.5]), array([ 0.5,  0.5]))
(array([ 0.5,  0.5]), array([ 0.5,  0.5]))
(array([ 0.5,  0.5]), array([ 0.5,  0.5]))
```

In this case they will always give the same result (this game indeed has a unique equilibria), however that's not always the case and also note that not all equilibria can be obtained from a given starting vertex pair.

Carrying out the Lemke Howson algorithm in this way is straightforward when the polytope can be visualised and/or the vertices themselves can all be enumerated to begin with. This is not always the case thankfully, using tools from linear programming we can traverse our polytopes using a technique called "pivoting" or a tabular representation of the polytopes called "tableaux" (plural).

A tableau corresponds to the inequalities relating to the payoffs considered but as equalities.The inequalities for the row polytope for the matching pennies game can be rewritten as:

1. $x_1 + 3x_2 + s_1 = 1$
2. $3x_1 + x_2 + s_2 = 1$

where s_1, s_2 are "slack" variables used to convert the inequality to an equality.

This can be represented in tableau form as:

$$T_r = \begin{pmatrix} 1 & 3 & 1 & 0 & 1 \\ 3 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Note that this corresponds to:

$$T_r = \begin{pmatrix} B^T & I & 1 \end{pmatrix}$$

1.1 Basic variables and labelled vertices

A basic variable of a tableau is a column (ignoring the last column) that is linearly independent from the others. Any given vertex of a polytope corresponds to a tableau where the non basic variables are set to 0 and the basic variables are solved. Non basic variables corresponds to labels of a vertex.

For example setting the non basic variables in T_r to 0 we get:

$$x_1 = x_2 = 0$$

which corresponds to the origin of the polytope (which is the starting point for the Lemke Howson algorithm). Thus the labels of vertex corresponds to the non basic variables.

1.2 Pivoting

In the Lemke Howson algorithm we need to drop a label and move along an edge of the Polytope to arrive at another vertex where a new label will replace the dropped label. This is equivalent to making a non basic variable basic (and vice versa). We do this by carrying out row operations on the tableau.

Let us drop the 0 label (corresponding to the first column) on T_c. We need to "pivot" the first column, so one of thoe non zero variables in that column will become 0. We identify this by choosing the row with the minimum positive ratio of $(T_r)_{i5}/(T_r)_{i1}$. This ensures that the resulting vertex is valid (and will have no negative variables). In our case the ratios in the first column are:

1. First row: $1/1$
2. Second row: $1/3$

Thus we pivot on the second row. We here take multiples of rows and carry out subtractions/additions of row so as to obtain a basic variable in the first column. In our case let us multiple the first row by 3 and subtract the second row from it:

$$T_r = \begin{pmatrix} 0 & 8 & 3 & -1 & 2 \\ 3 & 1 & 0 & 1 & 1 \end{pmatrix}$$

We now (as in the example we carried out before) have non basic variables x_2 and s_2 in other words we have labels: $\{1, 3\}$. We have thus picked up the label 3 which we need to drop from the other polytope.

In tableau form the column best response polytope is given by:

$$T_c = \begin{pmatrix} 1 & 0 & 3 & 1 & 1 \\ 0 & 1 & 1 & 3 & 1 \end{pmatrix}$$

Note that this corresponds to:

$$T_c = \begin{pmatrix} I & A & 1 \end{pmatrix}$$

we need to pivot the fourth column corresponding to label 3 of T_c. First the minimum ratios: $1/1 > 1/3$, so we pivot on the second row. Let us multiply the first row by 3 and subtract the second row from it:

$$T_c = \begin{pmatrix} 3 & -1 & 8 & 0 & 2 \\ 0 & 1 & 1 & 3 & 1 \end{pmatrix}$$

This has labels: $\{1, 2\}$ with 1 being the newly picked up label that we now drop (by pivoting the second column) from T_r:

$$T_r = \begin{pmatrix} 0 & 8 & 3 & -1 & 2 \\ 24 & 0 & -3 & 9 & 6 \end{pmatrix}$$

which has labels: $\{2, 3\}$ with 2 being the newly picked up label so we now pivot the third column of T_c:

$$T_c = \begin{pmatrix} 3 & -1 & 8 & 0 & 2 \\ -3 & 9 & 0 & 24 & 6 \end{pmatrix}$$

which has labels: $\{0, 1\}$ so we have a fully labelled vertex pair. Setting the non basic variables to 0 in each tableau we obtain:

$$((2/8, 8/24), (2/8, 6/24)) = ((1/4, 1/4), (1/4, 1/4))$$

which, when normalised to have sum 1 gives:

$$((1/2, 1/2), (1/2, 1/2))$$

A summary of integer pivoting:

1. Identify the row to pivot on using the minumum ratio test.
2. Carry out row operations to ensure that the pivot row is the only row with non zero variables in that column.