

Notebook

August 10, 2017

1 Normal Form Games

Game theory is the study of interactive decision making. Consider the following situation:

Two friends must decide what movie to watch at the cinema. Alice would like to watch a sport movie and Bob would like to watch a comedy. Importantly they would both rather spend their evening together than apart.

To represent this mathematically we will associate **utilities** to the 4 possible outcomes:

1. Alice watches a sport movie, Bob watches a comedy: Alice receives a utility of 1 and Bob a utility of 1.
2. Alice watches a comedy, Bob watches a sport movie: Alice receives a utility of 0 and Bob a utility of 0.
3. Alice and Bob both watch a sport movie: Alice receives a utility of 3 and Bob a utility of 2.
4. Alice and Bob both watch a comedy: Alice receives a utility of 2 and Bob a utility of 3.

This is referred to as the "battle of the sexes" and we will represent it using two matrices, A will represent the utilities of Alice:

$$A = \begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$$

and matrix B will represent the utilities of Bob:

$$B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$$

We refer to **Alice as the row player** and **Bob as the column player**:

- The row player chooses which row of the matrices the players will gain their utilities;
- The column player chooses which column of the matrices the player will gain their utilities.

Thus if the row player (Alice) chooses the first row (this corresponds to a Sports movie) and the column player (Bob) chooses the second column (this corresponds to a comedy):

- The row player receives a utility of $A_{12} = 1$
- The column player receives a utility of $B_{12} = 1$

This representation of the strategic interaction between Alice and Bob is called a **Normal Form Game**.

1.1 Definition of Normal Form Game

An (N) player normal form game consists of:

- A finite set of N players
- Strategy spaces for the players: $\{S_1, S_2, S_3, \dots, S_N\}$;
- Payoff functions for the players: $u_i : S_1 \times S_2 \cdots \times S_N \rightarrow \mathbb{R}$

In this course we will only consider the case of $N = 2$.

For the battle of the sexes:

- We have $(N=2)$ players (Alice and Bob)
- The strategy spaces: $S_1 = S_2 = \{\text{Comedy, Sport movie}\}$ or equivalently $S_1 = S_2 = \{1, 2\}$
- The payoff functions mapping an element of $\tilde{s} \in S_1 \times S_2 = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ to

\mathbb{R}

:

$$u_1(\tilde{s}) = A_{\tilde{s}}$$

$$u_2(\tilde{s}) = B_{\tilde{s}}$$

We can use Python to represent these games, we will use the `nashpy` library to do so and we start by building our two matrices:

```
In [1]: import nash
        A = [[3, 1], [0, 2]]
        B = [[2, 1], [0, 3]]
```

We then create a `nash.Game` instance:

```
In [2]: battle_of_the_sexes = nash.Game(A, B)
        battle_of_the_sexes
```

```
Out[2]: Bi matrix game with payoff matrices:
```

```
Row player:
[[3 1]
 [0 2]]
```

```
Column player:
[[2 1]
 [0 3]]
```

In the next chapter we will start to see how to use that for further calculations.

2 Examples of other common games

2.1 Prisoners dilemma

Assume two thieves have been caught by the police and separated for questioning. If both thieves cooperate and don't divulge any information they will each get a short sentence. If one defects he/she is offered a deal while the other thief will get a long sentence. If they both defect they both get a medium length sentence.

This corresponds to:

$$A = \begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 5 \\ 0 & 1 \end{pmatrix}$$

```
In [3]: A = [[3, 0], [5, 1]]
        B = [[3, 5], [0, 1]]
        prisoners_dilemma = nash.Game(A, B)
        prisoners_dilemma
```

Out[3]: Bi matrix game with payoff matrices:

```
Row player:
[[3 0]
 [5 1]]
```

```
Column player:
[[3 5]
 [0 1]]
```

2.2 Hawk Dove game

Suppose two birds of prey must share a limited resource. The birds can act like a hawk or a dove. Hawks always fight over the resource to the point of exterminating a fellow hawk and/or take a majority of the resource from a dove. Two doves can share the resource.

This corresponds to:

$$A = \begin{pmatrix} 0 & 3 \\ 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ 3 & 2 \end{pmatrix}$$

```
In [4]: A = [[0, 3], [1, 2]]
        B = [[0, 1], [3, 2]]
        hawk_dove = nash.Game(A, B)
        hawk_dove
```

Out[4]: Bi matrix game with payoff matrices:

```
Row player:
[[0 3]
```

```
[1 2]]
```

```
Column player:
```

```
[[0 1]  
[3 2]]
```

2.3 Pigs

Consider two pigs. One dominant pig and one subservient pig. These pigs share a pen. There is a lever in the pen that delivers food but if either pig pushes the lever it will take them a little while to get to the food. If the dominant pig pushes the lever, the subservient pig has some time to eat most of the food before being pushed out of the way. If the subservient pig push the lever, the dominant pig will eat all the food. Finally if both pigs go to push the lever the subservient pig will be able to eat a third of the food.

This corresponds to:

$$A = \begin{pmatrix} 4 & 2 \\ 6 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 3 \\ -1 & 0 \end{pmatrix}$$

```
In [5]: A = [[4, 2], [6, 0]]  
        B = [[2, 3], [-1, 0]]  
        pigs = nash.Game(A, B)  
        pigs
```

```
Out[5]: Bi matrix game with payoff matrices:
```

```
Row player:
```

```
[[4 2]  
[6 0]]
```

```
Column player:
```

```
[[ 2  3]  
[-1  0]]
```

2.4 Matching pennies

Consider two players who can choose to display a coin either Heads facing up or Tails facing up. If both players show the same face then player 1 wins, if not then player 2 wins.

This corresponds to:

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

```
In [6]: A = [[1, -1], [-1, 1]]  
        B = [[-1, 1], [1, -1]]  
        matching_pennies = nash.Game(A, B)  
        matching_pennies
```

Out[6]: Zero sum game with payoff matrices:

Row player:

```
[[ 1 -1]
 [-1  1]]
```

Column player:

```
[[ -1  1]
 [ 1 -1]]
```

As indicated by Nashpy, this is a Zero sum game:

$$A + B = 0$$

2.5 Definition of a zero sum game

A two player normal form game with payoff matrices A, B is called **zero sum** iff:

$$A = -B$$

To define a zero sum game using nashpy we can pass a single payoff matrix (it infers what the other will be):

```
In [7]: A = [[1, -1], [-1, 1]]
        matching_pennies = nash.Game(A)
        matching_pennies
```

Out[7]: Zero sum game with payoff matrices:

Row player:

```
[[ 1 -1]
 [-1  1]]
```

Column player:

```
[[ -1  1]
 [ 1 -1]]
```