Two-player games

For any two-player game, we have the following basic notation.

<table>
<thead>
<tr>
<th>Player 1 (P₁)</th>
<th>Player 2 (P₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set of actions</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>Action</td>
<td>( i \in S_1 )</td>
</tr>
<tr>
<td>Payoff/gain</td>
<td>( A_{ij} )</td>
</tr>
</tbody>
</table>

When the two players choose actions \( i, j \) respectively, their payoffs are \( A_{ij}, B_{ij} \) respectively. These can be conveniently represented as two matrices \( A, B \) each of size \( m \times n \), where \( m = |S_1| \) and \( n = |S_2| \), as follows:

\[
\begin{bmatrix}
1 & j & n \\
\vdots & \ddots & \vdots \\
1 & (A_{11}, B_{11}) & \cdots & \cdots \\
\vdots & (A_{ij}, B_{ij}) & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
m & (A_{mn}, B_{mn}) \\
\end{bmatrix}
\]

Due to this representation, these games are also called Bi-matrix games.

Example: Matching pennies

Both the players have two actions each given by, \( S_1 = S_2 = \{Heads, Tails\} \). \( P_1 \) aims to match the outcomes, while \( P_2 \) does not. The following payoffs capture this situation:

\[
\begin{bmatrix}
H \\
T \\
\end{bmatrix}
\begin{bmatrix}
(1, -1) & (-1, 1) \\
(-1, 1) & (1, -1) \\
\end{bmatrix}
\]

In this game, no pair of actions is stable. In such a case, the players can randomize. We formalize this next.
More notation and fundamentals

The randomization between possible actions, is achieved by what is called a mixed strategy. We denote the set of mixed strategies for $P_1$ and $P_2$, by $\Delta_1$ and $\Delta_2$ respectively, given by,

$$\Delta_1 = \{ x = (x_1, x_2, \ldots, x_{|S_1|}) \mid x_i \geq 0 \forall i \in S_1, \text{and} \sum_{i \in S_1} x_i = 1 \}$$

and,

$$\Delta_2 = \{ y = (y_1, y_2, \ldots, y_{|S_2|}) \mid y_j \geq 0 \forall j \in S_2, \text{and} \sum_{j \in S_2} y_j = 1 \}$$

When the two players play strategies $x \in \Delta_1$ and $y \in \Delta_2$ respectively, the expected payoff of $P_1$ is given by $\sum_{i \in S_1} j \in S_2 A_{ij} x_i y_j = x^T A y$, and similarly, that of $P_2$ is $x^T B y$. Thus, $P_1$ tries to maximize $x^T A y$, and $P_2$ tries to maximize $x^T B y$.

**Definition (Nash equilibrium).** A strategy profile $(x', y')$ is a Nash Equilibrium (NE) iff

$$x' \in \arg\max_{x \in \Delta_1} x^T A y' \quad \text{and} \quad y' \in \arg\max_{y \in \Delta_2} x^T B y$$

Having defined the NE, one would like to answer the following questions:

- How to check if a given strategy profile is a NE?
- Does a NE exist in a given game? In every game?
- How to compute a NE?

**Theorem (Nash ’51).** Every $n$-player game has a NE ($n \in \mathbb{N}$).

**Characterization of NE**

Fix $y$ for $P_2$. Then, $P_1$ gets a payoff of $(Ay)_i$ from action $i \in S_1$. Thus, the maximum possible from any action is $\max_{i \in S_1} (Ay)_i$ (say) $v$. Hence, playing $x$ gives $P_1$ a payoff of

$$x^T A y = \sum_{i \in S_1} x_i (Ay)_i = \text{convex combination of} (Ay)_i \text{'s}$$

$$\therefore \quad x^T A y \leq v \quad \& \quad x^T A y = v \text{ iff } (\forall i \in S_1, (x_i > 0 \Rightarrow (Ay)_i = v))$$

A similar analysis works for $P_2$ as well. Fixing $P_1$’s strategy to $x$, $P_2$ gets a payoff of $(x^T B)_j$ from action $j \in S_2$. Letting $w = \max_{j \in S_2} (x^T B)_j$, we can deduce,

$$\forall y \in \Delta_2, x^T B y \leq w \quad \& \quad x^T B y = w \text{ iff } (\forall j \in S_2, (y_j > 0 \Rightarrow (x^T B)_j = w))$$

We summarize this analysis as the following theorem characterizing Nash Equilibria:
Theorem 1. \((x, y)\) is a NE iff
\[
\forall i \in S_1: \ x_i > 0 \Rightarrow (Ay)_i = v \quad \text{and,} \\
\forall j \in S_2: \ y_j > 0 \Rightarrow (x^T B)_j = w
\]
where,
\[
v = \max_{i \in S_1} (Ay)_i \quad \& \quad w = \max_{j \in S_2} (x^T B)_j
\]
This theorem allows us to easily check if a strategy profile is NE.

Zero-sum games

In these games, we have,
\[
B_{ij} = -A_{ij} \quad \forall i \in S_1, \forall j \in S_2, \text{ i.e., simply } B = -A
\]
Hence, these games are described by just one matrix \(A\). \(P_1\) tries to maximize its payoff, and thus, maximize \(x^T Ay\). Similarly, \(P_2\) tries to maximize \(x^T (-A)y\), and thus, minimize \(x^T Ay\). Hence, \(P_1\) is called the maximizer and \(P_2\) is called the minimizer.

Minimax play in zero-sum games

Suppose both the players play pessimistically. To elaborate, \(P_1\) assumes that \(P_2\) can find out its strategy \(x\), ahead of time and play \(y\) accordingly to achieve its goal of minimization of \(x^T Ay\). \(P_2\) has a similar approach in choosing its strategy. Suppose they decide \(x^*, y^*\) as their strategies respectively, by playing pessimistically as described. Then, it must mean,

\[
x^* \in \arg\max_{x \in \Delta_1} \left( \min_{y \in \Delta_2} x^T Ay \right) \quad \& \quad y^* \in \arg\min_{y \in \Delta_2} \left( \max_{x \in \Delta_1} x^T Ay \right)
\]

Now, let \(\pi_1\) denote \(P_1\)'s guaranteed payoff, that is, the minimum worst-case payoff it can ensure - precisely as demonstrated in the pessimistic approach mentioned above. That is,

\[
\pi_1 = \max_{x \in \Delta_1} \left( \min_{y \in \Delta_2} x^T Ay \right) \quad \text{(1)}
\]

\[
= \min_{y \in \Delta_2} x^*^T Ay \quad \text{(2)}
\]

Similarly, let \(\pi_2\) be \(P_2\)'s guaranteed payoff, that is,

\[
\pi_2 = \min_{y \in \Delta_2} \left( \max_{x \in \Delta_1} x^T Ay \right) \quad \text{(3)}
\]

\[
= \max_{x \in \Delta_1} x^*^T Ay \quad \text{(4)}
\]

We now show a remarkable result.
Theorem 2. For \( x^*, y^*, \pi_1, \pi_2 \) as defined above, the following hold.

1. \( \pi_1 = \pi_2 = x^* A y^* \)
2. If \((x', y')\) is a NE, then, \( x'^T A y' = x^* A y^* \)
3. \((x^*, y^*)\) is a NE.

Proof. Using the definition of \( \pi_1 \) as in (2), it follows that, \( \pi_1 \leq x^* A y^* \).
Similarly, using the definition of \( \pi_2 \) in (4), it follows that, \( \pi_2 \geq x^* A y^* \).
Combining the two, we get,
\[
\pi_1 \leq x^* A y^* \leq \pi_2 \tag{5}
\]
Further, for a NE \((x', y')\), by definition of NE, we have,
\[
x'^T A y' = \max_{x \in \Delta_1} x^T A y' \tag{6}
\]
\[
x'^T A y' = \min_{y \in \Delta_2} x^T A y \tag{7}
\]
From (7) and (1), we get, \( \pi_1 \geq x'^T A y' \).
Similarly, from (6) and (3), we get, \( \pi_2 \leq x'^T A y' \).
Combining the two, we get,
\[
\pi_2 \leq x'^T A y' \leq \pi_1 \tag{8}
\]
(5) and (8) together prove the first two parts of the theorem.

Having proven \( \pi_2 = x^* A y^* \), and again from the definition of \( \pi_2 \) in (2), it follows that \( x^* \in \arg\max_{x \in \Delta_1} x^T A y^* \). Similarly, we can get \( y^* \in \arg\min_{y \in \Delta_2} x^* A y \). Hence, \((x^*, y^*)\) is a NE by definition, proving part 3 of the theorem.

Linear Programming Formulation (in zero-sum games)

Suppose the players are playing to optimize their worst-case payoffs as in the previous section. From \( P_2 \)'s perspective, fixing its strategy to \( y \in \Delta_2 \), \( P_1 \)'s best payoff is \( \max_{i \in S_1} (Ay)_i = \text{(say)} \ v^y \).
Hence, to minimize this, \( P_2 \) wants to solve for \( \min_{y \in \Delta_2} v^y \) - equivalently, this linear program LP:

\[
\begin{align*}
\min & \quad v \\
\text{s.t.} & \quad v \geq (Ay)_i \quad \forall i \in S_1, \\
& \quad \sum_{j \in S_2} y_j = 1, \\
& \quad y_j \geq 0 \quad \forall j \in S_2
\end{align*}
\]
The constraints in (2) and (3) ensure that $y \in \Delta_2$.

Letting the dual variables corresponding to the inequalities in (1) be $x_i$'s and the dual variable corresponding to (2) be $w$, the dual DLP of the linear program above, can be written as,

\[
\begin{align*}
\text{max} & \quad w \\
\text{s.t.} & \quad w \leq (x^T A)_j \quad \forall j \in S_2, \\
& \quad \sum_{i \in S_1} x_i = 1, \\
& \quad x_i \geq 0 \quad \forall i \in S_1
\end{align*}
\]

Then, it’s easy to see that DLP is equivalent to solving for $\max x^*, \quad w^* = \min_{x^* \in \Delta_1} (x^T A)_j$, and the constraints in (5) and (6) ensure that $x \in \Delta_1$. Thus, this is precisely what $P_1$ wants to do to maximize its worst-case payoff.

Consequently, we have the following theorem:

**Theorem 3.** The solution of LP gives $y^*$, and that of DLP gives $x^*$.

Further, the following follow from the properties of the linear programming solutions:

- The set of Nash Equilibria of a zero-sum game are convex.
- Computing an equilibrium can be done in polynomial time.