Lecture 1: Introduction to Games

Algorithmic and Mathematical Foundations of Game Theory and Economics
Yun Kuen Cheung, School of Computing, The Australian National University

1 What is a Game?

A game has two or more players. Each player chooses an action from her action space. The joint actions of all players determine the outcome, which is often represented numerically by the payoffs to the players.

When there are two players, we sometimes follow the convention in computer science to call the players Alice and Bob. The payoffs are specified by two matrices $\mathbf{A}, \mathbf{B}$ of the same dimensions. The rows of the each matrix correspond to Alice's actions, while the columns correspond to Bob's actions. If Alice chooses action j and Bob chooses action k, the payoffs to Alice and Bob are A_{jk} and B_{jk} respectively. Since such games are specified by two matrices, we often call them bimatrix games.

Example 1.1. One of the most well-known games is Rock-Paper-Scissors. Each player can choose one of the three actions: Rock (R), Paper (P) or Scissors (S). When there are two players, the payoffs are determined by: if a player wins, she gets \$1 from her opponent; both players get \$0 at a draw. The bimatrix game is:

		Bob's action		
		R	P	S
Alice's action	R	(0,0)	(-1,1)	(1,-1)
	P	(1,-1)	(0,0)	(-1,1)
	S	(-1,1)	(1,-1)	(0,0)

The bimatrix can be represented in the form of $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$ specifies the payoffs to Alice, and $\mathbf{B} = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$ specifies the payoffs to Bob.

When Alice chooses P and Bob chooses R, Alice wins, so she gets \$1 from Bob. In other words, Alice's payoff is \$1 and Bob's payoff is -\$1. To specify these two payoffs, the entry at the second row and the first column is (1,-1). When Alice and Bob both choose S, it is a draw and hence both players receive payoff \$0. This explains why the bottom-right corner entry is (0,0).

Example 1.2. Another well-known game is the *Prisoner's Dilemma*. Two members of a criminal gang are arrested. If both remain silent, they will be sentenced to one year on a minor charge. If one of them testifies and the other remains silent, the testifier will go free while the other member will be sentenced to five years on a major charge. If both testify, both will be sentenced to three years. The bimatrix game is:

		Bob's action	
		silent	testify
Alice's action	silent	(-1,-1)	(-5,0)
	testify	(0,-5)	(-3,-3)

Example 1.3. Consider the following game with three players. There are two piles of coins, which we call them Pile R and Pile S. Pile R has 6 coins and Pile S has 9 coins. Alice, Bob and Chris choose one of the piles simultaneously. The coins in each pile is split equally among the players who choose the pile.

To specify the payoffs of a three-player game, one common way is for each action of Alice, specify the payoffs to the three players in a "tri-matrix", like below:

Alice chooses Pile R

		Chris' action	
		R	S
Bob's action	R	(2,2,2)	(3,3,9)
	S	(3,9,3)	(6,4.5,4.5)

Alice chooses Pile S

		Chris' action	
		R	S
Bob's action	R	(9,3,3)	(4.5,6,4.5)
	S	(4.5,4.5,6)	(3,3,3)

When the game grows larger with more players and more actions, the above ad-hoc ways of specifying game payoffs are no longer feasible. In general, we can specify any game as follows.

Definition 1.4. Let n denote the number of players of a normal-form game. Let $\mathcal{A}_i$ denote the action space of player i. Let $\mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n$ denote the space of joint action profiles. The normal-form game is specified by the payoff function

$$u : \mathcal{A} \rightarrow \mathbb{R}^n .$$

For any joint action profile $(a_1, a_2, \ldots, a_n) \in \mathcal{A}$, $u(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$ specifies the payoffs to the n players. Let $u_i(a_1, a_2, \ldots, a_n)$ denote the payoff to player i at the joint action profile.

As a quick example, for the game in Example 1.3, $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}_3 = \{R, S\}$, and the payoff function is given by:

$$u(R,R,R) = (2,2,2) \quad u(R,R,S) = (3,3,9) \quad u(R,S,R) = (3,9,3) \quad u(R,S,S) = (6,4.5,4.5)$$
$$u(S,R,R) = (9,3,3) \quad u(S,R,S) = (4.5,6,4.5) \quad u(S,S,R) = (4.5,4.5,6) \quad u(S,S,S) = (3,3,3)$$

The complexity of specifying a game increases exponentially with the number of players. Suppose each player has two possible actions in an n-player game, then the game's dimension is $2^n \times n$, because there are 2^n possible joint action profiles, and we need to specify n payoffs for each joint action profile.

2 More Examples of Games and Action Spaces

When we attempt to model a situation as a game, the first thing to do is to identify the players and their action spaces. We have seen some examples of games in which the action space of each player is small. In general, a player can have many (possibly infinite) choices of actions. We will give more examples to illuminate how the action spaces can be formulated mathematically.

Example 2.1. There are two teams competing in a horse racing game. Team A has three horses with speeds 1, 3, 5 respectively, while Team B has three horses with speeds 2, 4, 6 respectively. The game proceeds in three rounds. In each round, each team sends out one horse to race; the horse with higher speed wins the round. A team cannot send the same horse to race in two different rounds. The two teams are required to submit the order of their horses to a judge at the start of the game. The team that wins more rounds than the other team is the overall winner.

Let h_j denote the horse with speed j. If Team A's ordering of its horses is $[h_3, h_1, h_5]$, while Team B's ordering of its horses is $[h_4, h_6, h_2]$, then Team B wins rounds 1 and 2, so Team B is the overall winner.

There are two players of the game, namely Team A and Team B. Team A's action space is the set of permutations of its three horses, i.e.,

$$\mathcal{A}_1 = \{[h_1, h_3, h_5],\ [h_1, h_5, h_3],\ [h_3, h_1, h_5],\ [h_3, h_5, h_1],\ [h_5, h_1, h_3],\ [h_5, h_3, h_1]\} .$$

Analogously, Team B's action space is

$$\mathcal{A}_2 = \{[h_2, h_4, h_6],\ [h_2, h_6, h_4],\ [h_4, h_2, h_6],\ [h_4, h_6, h_2],\ [h_6, h_2, h_4],\ [h_6, h_4, h_2]\} .$$

Some payoff function values are given below:

$$u([h_3, h_1, h_5], [h_4, h_6, h_2]) = (-1, 1) \qquad u([h_3, h_1, h_5], [h_4, h_2, h_6]) = (-1, 1)$$
$$u([h_3, h_5, h_1], [h_2, h_4, h_6]) = (1, -1) \qquad u([h_1, h_3, h_5], [h_6, h_2, h_4]) = (1, -1)$$

Example 2.2. Some players participate in *simultaneous sealed-bid auctions.* There are k items being auctioned. Each player i submits a sealed bid $\$b_{ij}$ for item j, where b_{ij} is a non-negative integer. There are two constraints. First, each player has a budget $\$m_i$ and she is not allowed to set total bid exceeding $\$m_i$. Second, for each item j she has a valuation $\$v_{ij}$, and she does not want her bid on item j exceeding $\$v_{ij}$. The action space of player i can be formally specified as

$$\mathcal{A}_i = \left\{ (b_{i1}, b_{i2}, \ldots, b_{ik}) \in \mathbb{Z}^k \;\middle|\; \sum_{j=1}^{k} b_{ij} \le m_i \text{ and } \forall j,\ 0 \le b_{ij} \le v_{ij} \right\}.$$

The cardinality of $\mathcal{A}_i$ can be large even for small values of k, m_i and v_{ij}. For instance, if $k = 4$, $m_i = 100$ and $v_{i1} = v_{i2} = v_{i3} = v_{i4} = 50$, then $|\mathcal{A}_i|$ is approximately 3.4 million.

We have not discussed how the payoff function is determined. This depends on how the items are allocated after the players make their bids. Is first-price auction or second-price auction used for each item? What is the tie-breaking rule when there are two or more highest bids? Is there a reserved price for an item? Is there a limit on the maximum number of items a player can be allocated? Depending on the wide variety of contexts, the payoff functions can be vastly different.

3 Time and Information

In the examples above, the players make a choice simultaneously at a single time. However, it is easy to come up with games which require making choices at different times. Moreover, the choice of a player at a certain time t will depend on the information she receives beforehand, e.g., the choices made by herself and other players before time t. Time and information are two crucial aspects.

Let's use a naive example to illustrate how time and information can change a game drastically. The two-player Rock-Paper-Scissors game in Example 1.1 requires Alice and Bob to choose actions simultaneously. The game is *fair* among the two players: as we shall see later, both players receive zero payoffs in expectation at the unique Nash equilibrium of the game. However, if the game is modified so that Alice chooses first, and Bob makes his choice after observing Alice's choice, then the game becomes unfair and boring: Bob can always win no matter what Alice chose. The game is entirely changed when Bob makes his choice in a later time with more information.

Normal-form games are defined with the presumption that all players make their choices simultaneously, so you may wonder if we need a new game model to handle various time and information aspects. Indeed, there is a clean model of *extensive-form games*, for which time and information are represented using *Kuhn trees*. We may discuss extensive-form games later; students who are eager to know about extensive form games can read the lecture notes of Prof. Thomas Ferguson. [1]

Nevertheless, extensive-form game model is not necessary, since the normal-form game model already provides sufficient machinery to handle the time and information aspects.[1] The idea is simple: before the start of the game, each player writes a computer program which will make all choices of actions throughout the game on behalf of the player. For the computer program to work under any circumstance, it specifies the choice to be made for *any* possible information that may be gathered just before the decision time.

Example 3.1. We revisit the coin split game in Example 1.3, but we modify the game as below. Alice will choose a pile first and let Bob and Chris know her decision. Then Bob and Chris choose their piles simultaneously. Alice's action space is $\mathcal{A}_1 = \{R, S\}$.

When Bob is about to choose his pile, he has the information of Alice's choice. But before the game starts, Bob does not know the choice of Alice. His computer program must be able to make a decision under any possible circumstance. His computer program should be like the one below:

```
if Alice_choice == R:
    Bob_choice = S
else:
    Bob_choice = R
```

The green parts in the above program can be changed. For each green part, there are two possible choices, namely R or S. Thus, there are $2^2 = 4$ possible computer programs, and hence Bob's action space has cardinality 4. Symmetrically, Chris's action space is identical to Bob's one.

Test Your Intuition: Which pile should Alice choose?

Example 3.2. We revisit the horse racing game in Example 2.1, but instead of requesting the teams to submit their orderings before the game starts, each team will send out one horse simultaneously just before each round. After the first round, the players know which horse their opponent has sent out. This information will be used by them to decide which horse to send next.

We specify the action space of Team A. Just before the first round, there is no information available, Team A can choose either h_1, h_3 or h_5 for the first round. Just before the second round, Team A gathers the information of which horse Team B has sent for the first round. For each of the possible information gathered, Team A should specify their choices for the second round (the choice for the third round is automatic since there will be only one horse left). Thus, each action in Team A's action space will look like a computer program below:

```
TeamA_first_choice = h3

if TeamB_first_choice == h2:
    TeamA_second_choice = h1
    TeamA_third_choice = h5
elif TeamB_first_choice == h4:
    TeamA_second_choice = h5
    TeamA_third_choice = h1
else:
    TeamA_second_choice = h5
    TeamA_third_choice = h1
```

[1]However, we note that extensive-form of a game is often easier to analyze than normal-form of the same game.

The green parts in the above program can be changed. There are three choices for `TeamA_first_choice`. For each possibility of `TeamB_first_choice`, there are two choices for `TeamA_second_choice`. Thus, the size of the action space is $3 \times 2^3 = 24$.

Test Your Intuition: Do the formats (submit orderings at the very beginning, or send horses round-by-round) make a *real* difference to the game?

4 Randomness

Another important aspect of games is randomness, which is essential to make many games interesting. As what we have done to handle time and information, an action in a game with randomness is like a computer program, which specifies the choice to be made for any possible random signal.

Example 4.1. Alice and Bob plays the following card game. There are six cards with labels 1 to 6 respectively. The game comprises of the following steps:

1. Each player gets two random cards. Each player can only see the labels on their own cards.
2. Alice chooses one of her cards to show Bob.
3. Bob bets on whether the sum of Alice two cards is larger, equal, or smaller than the sum of his own two cards.
4. After knowing Bob's bet, Alice bets on whether the sum of her two cards is larger, equal, or smaller than the sum of Bob's two cards.
5. Both players show their cards to their opponent. If Bob bets correctly but Alice bets incorrectly, Bob wins \$3 from Alice. If Alice bets correctly but Bob bets incorrectly, Alice wins \$1 from Bob. Otherwise, both players receive \$0.

We specify the action space of Alice. Her first decision is in Step 2. Just before that, the only information she can gather is the two random cards she got. There are $\binom{6}{2} = 15$ combinations of this random signal. For each possible signal, she chooses one of her two cards to show Bob. Her second decision is in Step 4. Just before that, she knows the extra information about Bob's bet, which has 3 possibilities. For each possibility, she decides her bet. Overall, the size of Alice's action space is $(2 \times 3^3)^{15} \approx 9.68 \times 10^{25}$. Each action will look like a computer program below:

```
switch (Alice's two cards):
  ··· # cases skipped
  case {4,6}:          # this demos one combination; the other 14
                       # combinations are analogous
    card_show = 4      # specify which card to show Bob
    switch (bob_bet):  # after knowing Bob's bet, Alice decides her bet
      case ''larger'':
        alice_bet = "equal"
      case ''equal'':
        alice_bet = "larger"
      case ''smaller'':
        alice_bet = "equal"
  ··· # cases skipped
```

Test Your Intuition: Which player has an advantage of winning money from their opponent?

References

[1] Thomas S. Ferguson. Game theory, 2000. https://www.cs.cmu.edu/afs/cs/academic/class/15859-s05/www/ferguson/mat.pdf.