Game Theory

Dynamic games of complete information

Xiang Sun

2021 Summer

Motivating Example 1: Grenade Game

- Consider a two-move game between two players.
  - First, player 1 decides whether to give \$1000 to player 2.
  - Second, after observing the choice of player 1, player 2 chooses whether to explode a grenade that will kill both of them.
- Player 2 can threaten player 1 by saying "Give the money to me, otherwise I will explode the grenade to kill you!"
- Question:
  - What should player 1 do in the first place?
  - Is player 2's threat credible to player 1?
  - What is the outcome of this simple game?

Motivating Example 2: The Farmer and The Snake

- On a winter evening, a farmer found a snake frozen with cold.
  - The farmer wanted to save the snake, which would make himself happy.
  - But he was worried if the snake would bite him after it was saved.
  - Believing that the snake would be grateful, the farmer saved it.
  - However, when the snake was recovered, it bit and killed the farmer immediately.
- Question: Why shouldn't the farmer save the snake?

Introduction

- The two examples differ from the games that we have studied before: players take actions sequentially, rather than simultaneously.
- These are examples of dynamic games (动态博弈).
- The central issue of dynamic games is credibility (可信性).
- We want to study dynamic games of complete information.
  - Dynamic: sequential choice or repeated play
  - Complete information: each player's payoff function is common knowledge among all players
- Remark: We can also translate these situations into static games of complete information, and solve NE therein. However, such modelings loss the key feature that the players move sequentially.

Introduction

- Two types of dynamic games of complete information (完全信息):
  1. Dynamic games of complete and perfect information (完美信息)
  2. Dynamic games of complete and imperfect information (不完美信息)
- In static games of complete information, we use normal-form (标准式) representation to describe a game.
- Now we use extensive-form (扩展式) representation for dynamic games.
- In particular, we will draw game trees (博弈树).

Games of Perfect Information

Consider a two-player and two-stage game.

- Player 1 chooses an action L or R.
- Player 2 observes player 1's action and then chooses an action L' or R'.
- Each path (a combination of two actions) in the following tree is followed by two payoffs: the first for player 1 and the second for player 2.

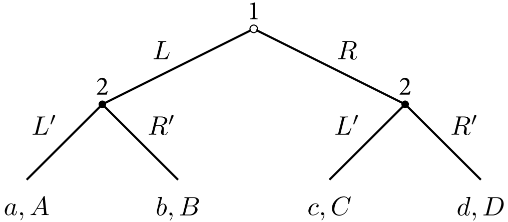


Figure: Extensive-form representation using a game tree

Games of Perfect Information

- The above game is an example of dynamic games of complete and perfect information.
- This type of games takes the following form:
  - Player 1 chooses an action a_1 from the feasible set A_1;
  - Player 2 observes a_1 and then chooses an action a_2 from the feasible set A_2;
  - Payoffs are $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$.
- Note that
  - A_2 may depend on the action a_1, i.e., $A_2(a_1)$.
  - Some action a_1 may even end the game, so that $A_2(a_1)$ is an empty set (i.e., no choice of player 2).
  - The action a_1 is perfectly observed by player 2.

Games of Perfect Information

In Example 1:

- $A_1 = \{\text{give \$1000}, \text{don't give}\}$;
- $A_2(\text{give \$1000}) = A_2(\text{don't give}) = \{\text{explode}, \text{don't explode}\}$.

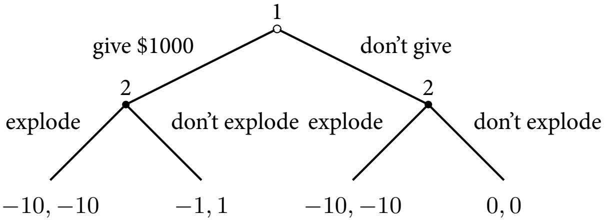


Figure: Game tree for Example 1

Games of Perfect Information

In Example 2:

- $A_1 = \{\text{save, don't save}\}$;
- $A_2(\text{save}) = \{\text{bite, don't bite}\}$.

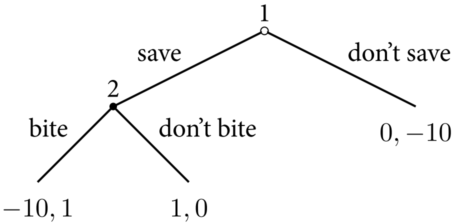


Figure: Game tree for Example 2

Games of Perfect Information

- Some key features of dynamic games of complete and perfect information:
  1. the moves occur in sequence;
  2. all previous moves are observed before the next move is chosen;
  3. the players' payoffs from each combination of moves are common knowledge.
- How to solve this type of games?

  Which actions should be taken?
- We use backwards induction (逆向归纳).

Backwards Induction

- In the second stage, player 2 observes the action (say a_1) chosen by player 1 in the first stage, and then chooses an action by solving

$$\max_{a_2 \in A_2} u_2(a_1, a_2).$$

- Assume this optimization problem has a unique solution, denoted by $R_2(a_1)$.

  This is player 2's best response to player 1's action a_1.
- For Example 1,

$$\begin{aligned} R_2(\text{give } \$1000) &= \text{don't explode}, \\ R_2(\text{don't give}) &= \text{don't explode}. \end{aligned}$$

Backwards Induction

- In the first stage, knowing player 2's best response, player 1's problem becomes

$$\max_{a_1 \in A_1} u_1\left(a_1, R_2(a_1)\right).$$

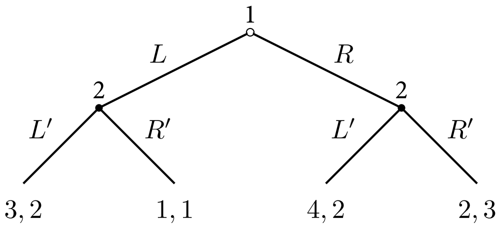
- Assume it also has a unique solution, denoted by a_1^*.
- For Example 1, $a_1^* =$ don't give and $R_2(a_1^*) =$ don't explode.
- We call $\left(a_1^*, R_2(a_1^*)\right)$ the backwards-induction outcome (逆向归纳的结果) of the game.

Backwards Induction

- In Example 1:
  - $R_2(\text{give } \$1000) = \text{don't explode}$, $R_2(\text{don't give}) = \text{don't explode}$.
  - $a_1^* = \text{don't give}$ and $R_2(a_1^*) = \text{don't explode}$.
  - The backwards-induction outcome is "(don't give, don't explode)".
- In Example 2:
  - $R_2(\text{save}) = \text{bite}$.
  - $a_1^* = \text{don't save}$.
  - The backwards-induction outcome is "don's save".

Backwards-induction outcome

- In the backwards-induction outcome, a_1^* is determined by maximizing $u_1(a_1, R_2(a_1))$, and $a_2^* = R_2(a_1^*)$.
- However, a_1^* may not maximize $u_1(a_1, a_2^*)$.
- Consider the following game:



- $R_2(L) = L'$ and $R_2(R) = R'$
- The backwards-induction outcome is (L, L').
- Given L', the best response of player 1 is R, not L.

Stackelberg Model of Duopoly

Consider a dominant firm moving first and a follower moving second.

- The game is played as follows:
  - Firm 1 chooses a quantity $q_1 \geq 0$.
  - Firm 2 observes q_1 and then chooses a quantity $q_2 \geq 0$.
  - The payoff of firm i is the profit

    $$\pi_i(q_1, q_2) = q_i[P(Q) - c],$$

    where $Q = q_1 + q_2$ and

    $$P(Q) = \begin{cases} a - Q, & \text{if } Q < a; \\ 0, & \text{if } Q \geq a. \end{cases}$$

- How to find the backwards-induction outcome?

Stackelberg Model of Duopoly

- First, find the best response function $R_2(q_1)$ for firm 2, i.e., for any given q_1, find q_2 that solves

$$\max_{q_2 \geq 0} \pi_2(q_1, q_2),$$

  where

$$\pi_2(q_1, q_2) = \begin{cases} q_2(a - q_1 - q_2 - c), & \text{if } q_1 + q_2 < a; \\ -cq_2, & \text{if } q_1 + q_2 \geq a. \end{cases}$$

- Then we have

$$R_2(q_1) = \begin{cases} \frac{a-c-q_1}{2}, & \text{if } q_1 < a - c; \\ 0, & \text{if } q_1 \geq a - c. \end{cases}$$

- $R_2(q_1)$ is the same as that in the Cournot model.

Stackelberg Model of Duopoly

- Second, firm 1 knows $R_2(q_1)$ and solves

$$\max_{q_1 \geq 0} \pi_1\left(q_1, R_2(q_1)\right),$$

where

$$\pi_1\left(q_1, R_2(q_1)\right) = \begin{cases} q_1\left[a - q_1 - \frac{a-q_1-c}{2} - c\right], & \text{if } q_1 < a - c; \\ q_1[a - q_1 - c], & \text{if } a - c \leq q_1 < a; \\ -cq_1, & \text{if } q_1 \geq a. \end{cases}$$

Stackelberg Model of Duopoly

- Clearly, for $q_1 > a - c$, firm 1's profit is always negative.
- Thus we only need to solve

$$\max_{a-c \geq q_1 \geq 0} q_1 \left[a - q_1 - \frac{a - q_1 - c}{2} - c \right],$$

which leads to the following first-order condition

$$a - c - 2q_1 = 0.$$

- The optimal choice of firm 1 is

$$q_1^* = \frac{a - c}{2}.$$

Stackelberg Model of Duopoly

- The quantity chosen by firm 2 is

$$q_2^* = R_2(q_1^*) = \frac{a-c}{4}.$$

- The market price is

$$P^* = a - q_1^* - q_2^* = c + \frac{a-c}{4}.$$

- Firms' profits and the total profit are

$$\pi_1^* = \frac{(a-c)^2}{8},\ \pi_2^* = \frac{(a-c)^2}{16},\ \text{and } \Pi^* = \frac{3(a-c)^2}{16}.$$

- Question: What's the game tree of this game?

Cournot model vs. Stackelberg model

Variable	Cournot Model	Stackelberg Model
q_1^*	$\frac{a-c}{3}$	$\frac{a-c}{2}$
q_2^*	$\frac{a-c}{3}$	$\frac{a-c}{4}$
π_1^*	$\frac{(a-c)^2}{9}$	$\frac{(a-c)^2}{8}$
π_2^*	$\frac{(a-c)^2}{9}$	$\frac{(a-c)^2}{16}$
Π^*	$\frac{2(a-c)^2}{9}$	$\frac{3(a-c)^2}{16}$
P^*	$c+\frac{a-c}{3}$	$c+\frac{a-c}{4}$

Games of Imperfect Information

- Consider the following simple two-stage game:
  - Players 1 and 2 simultaneously choose actions a_1 and a_2 from the feasible sets A_1 and A_2, respectively.
  - Players 3 and 4 observe the outcome of the first stage (a_1, a_2) and then simultaneously choose actions a_3 and a_4 from the feasible sets A_3 and A_4, respectively.
  - Payoffs are $u_i(a_1, a_2, a_3, a_4)$ for $i = 1, 2, 3, 4$.
- This game differs from the two-stage game with perfect information, since there are simultaneous moves within each stage.

Games of Imperfect Information

- We solve this game by using the idea of backwards induction.
- For each given (a_1, a_2), players 3 and 4 try to find the Nash equilibrium in stage 2. (why NE here?)
- Assume the second-stage game has a unique Nash equilibrium

$$\left(a_3^*(a_1, a_2), a_4^*(a_1, a_2)\right).$$

- Then, player 1 and player 2 play a simultaneous-move game with payoffs

$$u_i\left(a_1, a_2, a_3^*(a_1, a_2), a_4^*(a_1, a_2)\right), \text{ for } i = 1, 2.$$

Games of Imperfect Information

- Suppose (a_1^*, a_2^*) is the unique Nash equilibrium of this simultaneous-move game.
- Then

$$\big(a_1^*, a_2^*, a_3^*(a_1^*, a_2^*), a_4^*(a_1^*, a_2^*)\big)$$

is the subgame-perfect outcome (子博弈精炼结果) of the two-stage game.

Bank Runs

- Two investors have each deposited \$5 millions with a bank. The bank has invested these deposits in a long-term project.
- If the bank is forced to liquidate its investment before the project matures, a total of \$8 millions can be recovered.
- If the bank allows the investment to reach maturity, the project will pay out a total of \$16 millions.
- There are two dates at which the investors can make withdrawals at the bank: Date 1 is before the bank's investment matures and Date 2 is after.
- Suppose there is no discounting.

Bank Runs

- Players' payoffs in date 1:

	Withdraw	Don't
Withdraw	$4, 4$	$5, 3$
Don't	$3, 5$	next stage

- Players' payoffs in date 2:

	Withdraw	Don't
Withdraw	$8, 8$	$11, 5$
Don't	$5, 11$	$8, 8$

Bank Runs

We work backwards:

- At date 2, in the unique Nash equilibrium, both withdraw and each obtains \$8.
- At date 1, they play the following game:

	Withdraw	Don't
Withdraw	$4, 4$	$5, 3$
Don't	$3, 5$	$8, 8$

- There are 2 pure-strategy Nash equilibria of this game:
  1. Both withdraw and each obtains \$4;
  2. Both don't and each obtains \$8.

Bank Runs

- There are 2 subgame-perfect outcomes of the original two-stage game:
  1. Both withdraw at date 1 to obtain \$4 $\rightarrow$ the case of bank run;
  2. Both don't withdraw at date 1 but do at date 2, and obtain \$8.
- Although there are two possible subgame-perfect outcomes, only the second one is efficient.
- This model does not predict when bank runs will occur, but does show that they can occur as an equilibrium outcome.
- Question: What's the game tree of this game?

Normal-form Representation of Games

In static games, we consider normal-form representation to describe a game.

Definition

The normal-form (标准式) representation of a game specifies

1. the players in the game;
2. the strategies available to each player;
3. the payoff received by each player for each combination of strategies that could be chosen by the players.

Extensive-form Representation of Games

In dynamic games, we need to use extensive-form representation.

Definition

The extensive-form (扩展式) representation of a game specifies:

(1) the players in the game;

(2a) when each player has the move;

(2b) what each player can do at each of his or her opportunities to move;

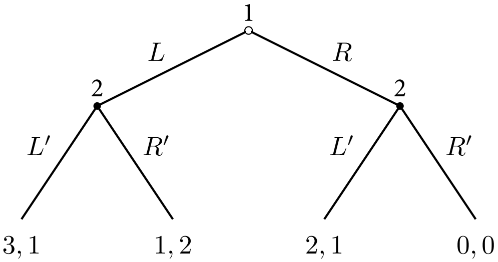
(2c) what each player knows at each of his or her opportunities to move;

(3) the payoffs received by each player for each combination of moves that could be chosen by the players.

Note that (2a)–(2c) describe strategies (策略) of each player in detail.

Extensive-form Representation of Games

- We use game trees for extensive-form representations.
- Example 3:



Extensive-form Representation of Games

- In Example 3, the game tree begins with a decision node (节点) for player 1, which is also the initial node (初始节点) of the game.
- After player 1's choice (L or R) is made, player 2's decision node is reached. And player 2 needs to decide whether to choose L' or R'.
- A terminal node (终止节点) is reached after player 2's move (i.e., the game ends), and payoffs of players are realized.

Information Set

- A dynamic game of complete and perfect information is a game in which the players move in sequence, all previous moves are observed before the next move is chosen, and payoffs are common knowledge.
- Such games can be easily represented by a game tree.

- For games with imperfect information, some previous moves are not observed by the player with the current move.
- To present this kind of ignorance of previous moves and to describe what each player knows at each of his/her move, we introduce the notion of a player's information set (信息集).

Information Set

Definition

An information set (信息集) for a player is a collection of decision nodes satisfying:

(i) The player needs to move at every node in the information set.

(ii) When the play of the game reaches a node in the information set, the player with the move does not know which node in the set has (or has not) been reached.

- (ii) implies that the player must have the same set of feasible actions at each decision node in an information set; Otherwise the player could infer from the set of actions available that some node(s) had or had not been reached.
- Any two nodes from different information sets of a player can be distinguished from each other.

Information Set

- 谈论信息集时，需要指明所考虑的参与者；
  不可脱离参与者来空谈信息集。
- 对于每个参与者，他每个信息集中的所有节点所提供的信息是相同的。
- 对于每个参与者，他的所有节点可以按照“是否属于同一个信息集”进行分组：
  - 同一组中的节点，属于同一个信息集；
  - 不同组中的节点，属于不同的信息集。
- 信息集的特殊形式：仅包含一个节点。

Information Set: Perfect vs. Imperfect

- In an extensive-form game, a collection of decision nodes, which constitutes an information set, is connected by a dotted line.
- We can use information set to differentiate perfect and imperfect information.
- A game is
  - of perfect information (完美信息) if every information set is a singleton;
  - of imperfect information (不完美信息) if there is at least one non-singleton information set.

Information Set: Static Game

- 动态场景自然可以建模为动态博弈，静态场景亦可以建模为动态博弈；

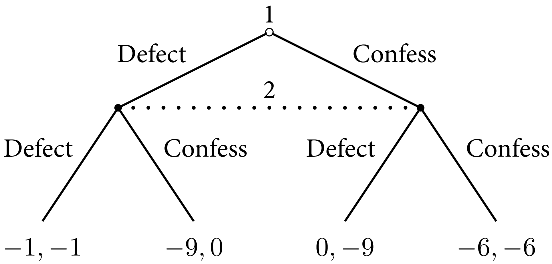
  这时需要信息集来描述“静态”这个特点。
- Let's consider a two-player simultaneous-move (static) game as follows:
  1. Player 1 chooses $a_1 \in A_1$;
  2. Player 2 does not observe player 1's move but chooses an $a_2 \in A_2$;
  3. Payoffs are $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$.
- We need an information set to describe player 2's ignorance of player 1's actions.
- The above static game of complete information can be represented as a dynamic game of complete but imperfect information.

Information Set: Example 4

- Example 4: Prisoners' Dilemma.
- The normal-form representation is

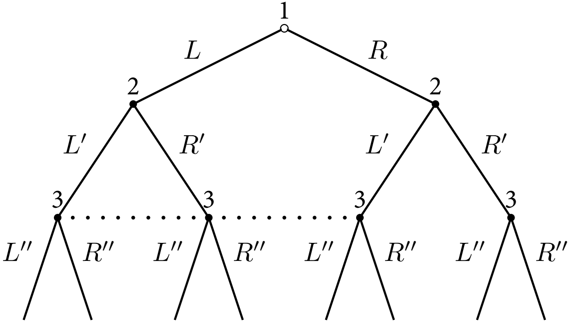
		Prisoner 2	
		Defect	Confess
Prisoner 1	Defect	$-1, -1$	$-9, 0$
	Confess	$0, -9$	$-6, -6$

- The extensive-form representation of Example 4 is:



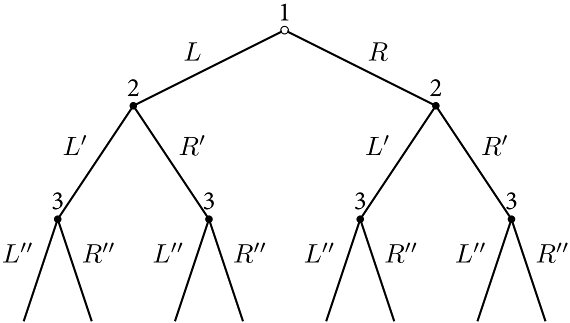
Information Set: Example 5

In the following Example 5, Player 3 has a non-singleton information set and a singleton information set.



Information Set: Example 6

In the following Example 6, Player 3 has 4 singleton information sets.



Strategy

Definition

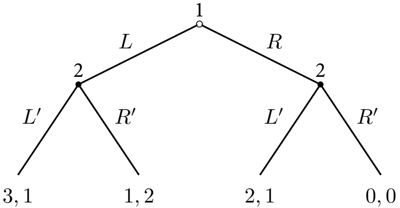
A strategy (策略) for a player is a complete plan of actions. It specifies a feasible action for the player in every contingency in which the player might be called on to act.

- For dynamic games with complete information:

  A player's strategy is a function which assigns an action to each information set (not each decision node) belonging to the player.
- An action and a strategy do not make a big difference in static games, while they do in dynamic games.

Strategy: Example 3

In Example 3:



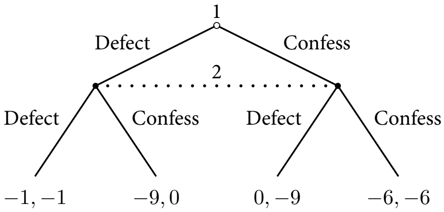
- Player 1 has 2 actions (and also 2 strategies): L and R.
- Player 2 has 2 actions: L' and R', but 4 strategies:

$$(L', L'); (L', R'); (R', L'); (R', R').$$

- For example, the strategy (L', R') means:
  - if player 1 plays L, then player 2 plays L';
  - if player 1 plays R, then player 2 plays R'.

Strategy: Example 4

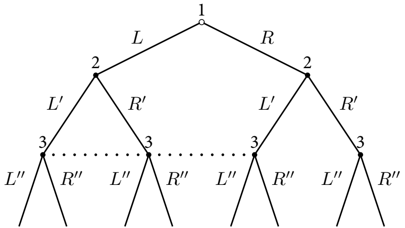
In Example 4:



- Both players have two actions and also two strategies: Defect and Confess.

Strategy: Example 5

In Example 5:



- Player 1 has two strategies: L and R.
- Player 2 has four strategies:

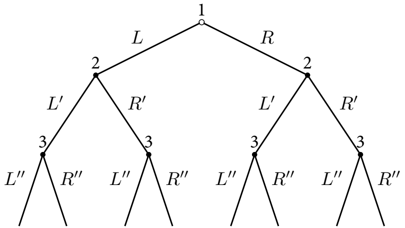
$$(L', L'); (L', R'); (R', L'); (R', R').$$

- Player 3 has four strategies:

$$(L'', L''); (L'', R''); (R'', L''); (R'', R'').$$

Strategy: Example 6

In Example 6:

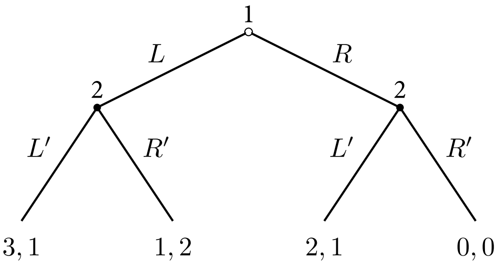


- Player 3 has 16 strategies.
- For instance, the strategy (L'', R'', R'', L'') means:
    - if player 1 plays L and player 2 plays L', then player 3 plays L'';
    - if player 1 plays L and player 2 plays R', then player 3 plays R'';
    - if player 1 plays R and player 2 plays L', then player 3 plays R'';
    - if player 1 plays R and player 2 plays R', then player 3 plays L''.

Strategy: Cournot and Stackelberg

- In the Cournot model of duopoly, firm i's action and strategy is the same, i.e., $q_i \geq 0$.
- In the Stackelberg model, the action and strategy for firm 1 (the leader) is again $q_1 \geq 0$.
- How about firm 2 (the follower)? How many information sets does firm 2 have?
- Firm 2's action is $q_2 \geq 0$, but its strategy is a function $q_2(q_1) \geq 0$ for any $q_1 \geq 0$.

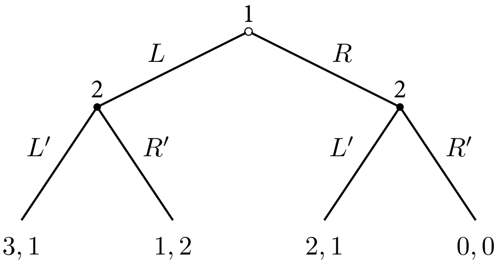
Problems of NE



Question:

- What's the normal-form representation of this game?
- What's the NE?

Problems of NE

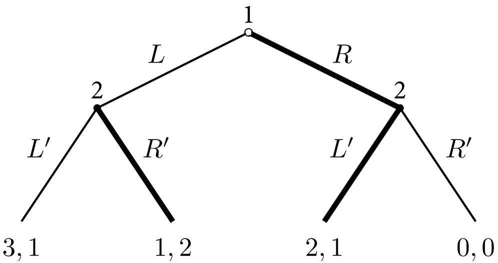


Player 2

Player 1	$L'L'$	$L'R'$	$R'L'$	$R'R'$
L	3, 1	3, 1	1, 2	1, 2
R	2, 1	0, 0	2, 1	0, 0

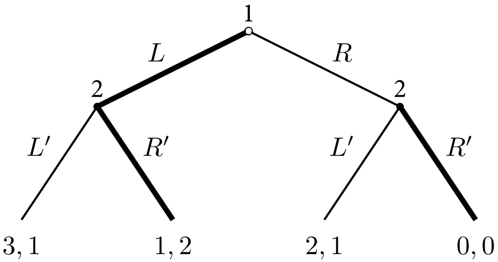
There are two NE: $(L, R'R')$ and $(R, R'L')$.

Problems of NE



- NE $(R, R'L')$ seems okay.
- It respects the spirit of backwards induction.

Problems of NE



- NE $(L, R'R')$ has a problem: No matter which action is chosen by Player 1, player 2 must choose L' at the right node.
- Interpretation: Player 2 tells player 1: if you choose R, I will choose R' (threat), then each of us will get 0.
- This threat is non-creditable: Player 1 should not believe that player 2 will choose R' after observing R.
- Key: Ignorance of dynamic feature.

Subgame

To capture the dynamic feature, we consider subgames.

Definition

A subgame (子博弈) in an extensive-form game

(a) begins at a decision node n that is a singleton information set (but is not the game's initial node);

(b) includes all the decision and terminal nodes following node n in the game tree (but no nodes that do not follow n);

(c) does not cut any information sets (i.e., if a decision node n' follows n in the game tree, then all other nodes in the information set containing n' must also follow n, and so must be included in the subgame).

Subgame

- Example 3 has 2 subgames.
- Example 4 has no subgame.

  Player 2's decision nodes are in the same non-singleton information set.
- Example 5 has only 1 subgame, beginning at player 3's decision node following R and R'.

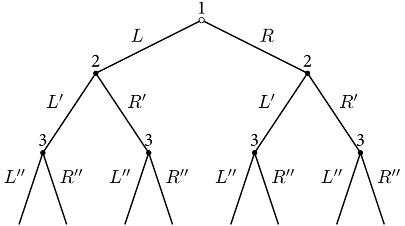
  The subtrees beginning at player 2's decision nodes violate (c).
- Example 6 has 6 subgames.

Strategy Profile: From Game To Subgame

- We have a strategy profile $s = (s_1, \ldots, s_n)$ for the given game.
- When we focus on a subgame, we shall only consider the relevant part of the strategy profile s.
- Relevant part of s specifies the "complete" plans (or strategy profile) for the players in that subgame.

Strategy Profile: From Game To Subgame

In Example 6:



- We have a strategy profile $\big(L, (L', R'), (L'', R'', R'', L'')\big)$.
- We turn to the subgame beginning at player 2's right decision node.
- The relevant part is $\big(R', (R'', L'')\big)$ or $\big(-, R', (R'', L'')\big)$.

  It is a strategy profile for this subgame.

  We can discuss whether it is "reasonable", within this subgame.
- One can repeat this procedure for every subgame.

Subgame-Perfect Nash Equilibrium

Definition (Selten, 1965)

A Nash equilibrium is subgame-perfect (子博弈精炼), or is said to be a subgame-perfect Nash equilibrium (子博弈精炼均衡) if the players' strategies constitute a Nash equilibrium in every subgame.

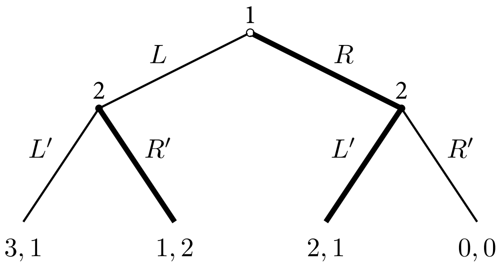
$$
\begin{tikzcd}
\text{Game} \arrow[rr] \arrow[d] & & \text{Subgame(s)} \\
\text{NE} \arrow[r, "\text{relevant part}"'] & \text{Strategy profile of subgame} \arrow[ur, "\text{NE?}"'] &
\end{tikzcd}
$$

Subgame-Perfect Nash Equilibrium

- It can be shown that any finite dynamic game of complete information has a subgame-perfect Nash equilibrium, perhaps in mixed-strategies.
- To find subgame-perfect Nash equilibria,
  - we first need to find Nash equilibria in each subgame,
  - then use backwards-induction to solve for the whole game.

Subgame-Perfect Nash Equilibrium

- In Example 1, there are two subgames:
    - in the left subgame, the Nash equilibrium involves the player 2 choosing R';
    - in the right subgame, the Nash equilibrium involves the player 2 choosing L'.
- The subgame-perfect Nash equilibrium is $\big(R, (R', L')\big)$.
- We can use thick lines to represent the equilibrium paths.



Equilibrium vs. Outcome

- Subgame-perfect Nash equilibrium is closely related to two previous concepts:
  1. backwards-induction outcome;
  2. subgame-perfect outcome.

  两者仅存在术语的差别，没有什么本质区别；混用没有问题。
- What's the difference between an equilibrium and an outcome?
- An equilibrium is a collection of players' strategies (strategy profile), while an outcome is a collection of players' actions.

Equilibrium vs. Outcome

- Consider the following two-stage game of complete and perfect information:
  1. Player 1 chooses an action $a_1 \in A_1$;
  2. Player 2 observes a_1 and then chooses an action $a_2 \in A_2$;
  3. Payoffs are $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$.
- The best response $R_2(a_1)$ solves $\max_{a_2 \in A_2} u_2(a_1, a_2)$.
- a_1^* solves $\max_{a_1 \in A_1} u_1\big(a_1, R_2(a_1)\big)$.

Equilibrium vs. Outcome

- The backwards-induction outcome is $\big(a_1^*, R_2(a_1^*)\big)$.
- The subgame-perfect Nash equilibrium is $\big(a_1^*, R_2(\cdot)\big)$.
- Note that $R_2(a_1^*)$ is an action, while $R_2(\cdot)$ is a strategy for player 2.
- In Example 1:
  - (R, L') is the backwards-induction outcome,
  - while $\big(R, (R', L')\big)$ is the subgame-perfect Nash equilibrium.
- In the Stackelberg model:
  - The backwards-induction outcome is (q_1^*, q_2^*), where $q_1^* = \frac{a-c}{2}$ and $q_2^* = \frac{a-c}{4}$,
  - while the subgame-perfect Nash equilibrium is $\big(q_1^*, R_2(q_1)\big)$, where $R_2(q_1) = \frac{a-c-q_1}{2}$.

Equilibrium vs. Outcome

- Consider the following two-stage game of complete but imperfect information:
  - Players 1 and 2 simultaneously choose actions a_1 and a_2 from the feasible sets A_1 and A_2, respectively.
  - Players 3 and 4 observe the outcome of the first stage (a_1, a_2) and then simultaneously choose actions a_3 and a_4 from the feasible sets A_3 and A_4, respectively.
  - Payoffs are $u_i(a_1, a_2, a_3, a_4)$ for $i = 1, 2, 3, 4$.
- For each given (a_1, a_2), players 3 and 4 play the Nash equilibrium in stage 2

$$\left(a_3^*(a_1, a_2), a_4^*(a_1, a_2)\right).$$

Equilibrium vs. Outcome

- Then, player 1 and player 2 play a simultaneous-move game with payoffs
$$u_i\big(a_1, a_2, a_3^*(a_1, a_2), a_4^*(a_1, a_2)\big), i = 1, 2.$$
- Suppose (a_1^*, a_2^*) is the unique Nash equilibrium in stage 1.
- Then the subgame-perfect outcome is
$$\big(a_1^*, a_2^*, a_3^*(a_1^*, a_2^*), a_4^*(a_1^*, a_2^*)\big).$$
- The subgame-perfect Nash equilibrium is
$$\big(a_1^*, a_2^*, a_3^*(a_1, a_2), a_4^*(a_1, a_2)\big).$$

Nash Equilibrium vs. Subgame-Perfect Nash Equilibrium

- A Nash equilibrium may not be subgame-perfect.
- In Example 3, the normal-form representation is

		Player 2			
		(L', L')	(L', R')	(R', L')	(R', R')
Player 1	L	$3, 1$	$3, 1$	$1, 2$	$1, 2$
	R	$2, 1$	$0, 0$	$2, 1$	$0, 0$

- Two Nash equilibria: $\big(L, (R', R')\big)$ and $\big(R, (R', L')\big)$.
- Only one subgame-perfect Nash equilibrium: $\big(R, (R', L')\big)$.

Nash Equilibrium vs. Subgame-Perfect Nash Equilibrium

- The Nash equilibrium $\big(R, (R', L')\big)$ is subgame-perfect, because R' and L' are the optimal strategies in the left and right subgames, respectively, where player 2 is the only player.
- On the other hand, the Nash equilibrium $\big(L, (R', R')\big)$ is not subgame-perfect, because when player 1 chooses R, R' is not optimal to player 2 in the right subgame, i.e., R' is not a Nash equilibrium in that subgame.
- One can think the strategy (R', R') by player 2 as a threat to player 1.

Nash Equilibrium vs. Subgame-Perfect Nash Equilibrium

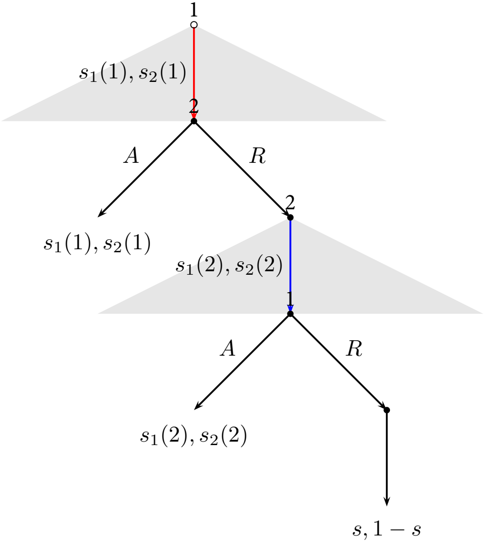
- Nash equilibria that rely on non-credible threats or promises can be eliminated by the requirement of subgame perfection.
- Subgame-perfect Nash equilibrium is a refinement of Nash equilibrium, i.e.,

$$\{\text{Subgame-perfect Nash equilibria}\} \subseteq \{\text{Nash equilibria}\}$$

Sequential Bargaining Game

- Suppose players 1 and 2 are bargaining over one dollar.
- They discount payoffs received a period later by a discount factor δ, where $0 < \delta < 1$.
- Consider the following three-period bargaining game:
  - (1a) In the first period, player 1 proposes $s_1(1)$ for himself and $s_2(1)$ for player 2.
  - (1b) Player 2 either accepts the offer to end the game or rejects the offer to continue the game.
  - (2a) In the second period, player 2 proposes $s_1(2)$ for player 1 and $s_2(2)$ for himself.
  - (2b) Player 1 either accepts the offer to end the game or rejects the offer to continue the game.
  - (3) In the third period, player 1 receives a share s of the dollar, leaving $1 - s$ to player 2.

Game Tree



Value

- Let $s_1(3) = s$ and $s_2(3) = 1 - s$.
- In general, in period t, $s_1(t)$ and $s_2(t)$ are offered to players 1 and 2. The offers satisfy
$$s_1(t) + s_2(t) = 1.$$
- The present value (现值) of payoff to player i is $\delta^{t-1} s_i(t)$ if the bargaining is ended in period t.
- We use backwards induction to solve the game.

Analysis: Step 2

In the second period, player 2 is at the move.

- The payoff to player 1 in period 3 is s.
  $\Longrightarrow$ Player 1 accepts iff $s_1(2) \geq \delta s$.
- If player 2 wants player 1 to accept, the best choice is $s_1(2) = \delta s$, which results the payoff $1 - \delta s$.
- If player 2 were to make player 1 not accept, the payoff is always $\delta(1 - s)$, which is less than $1 - \delta s$.
- The best choice of player 2 is $s_1(2) = \delta s$, where player 1 will accept.

Analysis: Step 1

In the first period, player 1 is at the move.

- The payoff to player 2 in period 2 is $1 - \delta s$.
  $\Longrightarrow$ Player 2 accepts iff $s_2(1) \geq \delta(1 - \delta s)$.
- If player 1 wants player 2 to accept, the best choice is $s_2(1) = \delta(1 - \delta s)$, which results the payoff $1 - \delta(1 - \delta s)$.
- If player 1 were to make player 2 not accept, the payoff is always $\delta\delta s$, which is less than $1 - \delta(1 - \delta s)$.
- The best choice of player 1 is $s_2(1) = \delta(1 - \delta s)$, where player 2 will accept.

Backwards-induction outcome

- The backwards-induction outcome of the three-period bargaining game:
- Player 1 offers the settlement

$$
\begin{aligned}
s_1^*(1) &= 1 - \delta(1 - \delta s), \\
s_2^*(1) &= \delta(1 - \delta s).
\end{aligned}
$$

- Player 2 accepts the offer, and the game ends.

Summary

- We have considered dynamic games of complete information.
- Two basic questions:
  1. How to describe a dynamic situation $\rightarrow$ extensive-form representation.
  2. How to solve a dynamic game? Why to introduce SPNE?
- Backwards induction vs. SPNE.

- 静态场景 $\rightarrow$ 静态模型 $\rightarrow$ 标准式
  静态场景 $\rightarrow$ 动态模型（带有不完美信息）$\rightarrow$ 扩展式（没有失去场景的特点）
- 动态场景 $\rightarrow$ 动态模型 $\rightarrow$ 扩展式
  动态场景 $\rightarrow$ 静态模型 $\rightarrow$ 标准式（失去场景的动态特点）