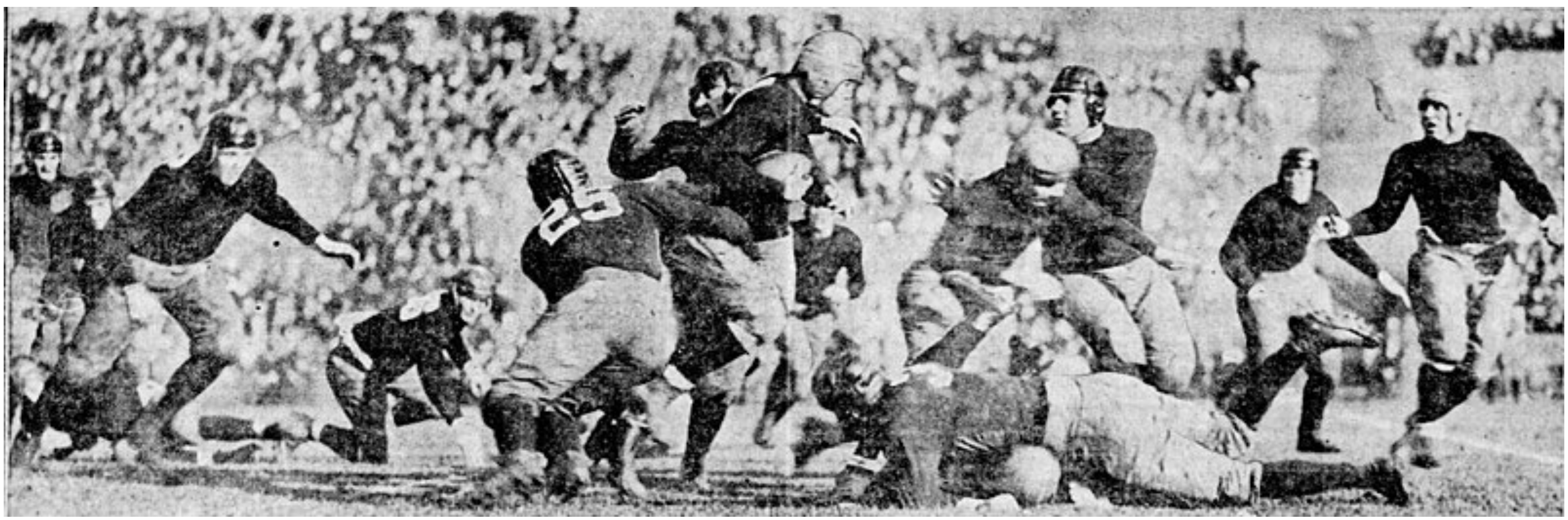
CS 440/ECE448 Lecture 32: Mechanism Design & Repeated Games

Mark Hasegawa-Johnson



Public domain. https://commons.wikimedia.org/wiki/File:Indiana_In_Early_Fight,_But_Succumbs_To_Chicago,_November_10,_1923_(cropped).jpg

Outline

- Adversarial learning review
- Mechanism design
- Goals that might motivate mechanism design
- Repeated games

Adversarial learning

- Suppose both Alice and Bob are using mixed strategies:

$$\boldsymbol{\alpha} = \begin{bmatrix} 1-\sigma(a) \\ \sigma(a) \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} 1-\sigma(b) \\ \sigma(b) \end{bmatrix}$$

- On successive days, they each try to improve their strategies using gradient ascent:

$$\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow \begin{bmatrix} a \\ b \end{bmatrix} + \eta \begin{bmatrix} \frac{\partial E[r_A]}{\partial a} \\ \frac{\partial E[r_B]}{\partial b} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + \eta \begin{bmatrix} \left(\frac{\partial \boldsymbol{\alpha}}{\partial a}\right)^T \boldsymbol{A}\boldsymbol{\beta} \\ \boldsymbol{\alpha}^T \boldsymbol{B} \frac{\partial \boldsymbol{\beta}}{\partial b} \end{bmatrix}$$

…but it's an Unstable Equilibrium!!

If we start off the equilibrium, $\boldsymbol{\alpha}^T = \left[\frac{1}{2}, \frac{1}{2}\right], \boldsymbol{\beta}^T = \left[\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon\right]$, what happens?

$$\frac{\partial E[r_A]}{\partial a} = \left(\frac{\partial \boldsymbol{\alpha}}{\partial a}\right)^T \boldsymbol{A\beta} = \begin{bmatrix} 1 - \frac{\partial \sigma(a)}{\partial a} \\ \frac{\partial \sigma(a)}{\partial a} \end{bmatrix}^T \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 - \sigma(b) \\ \sigma(b) \end{bmatrix}$$

$$= \left[\frac{3}{4}, \frac{1}{4}\right] \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} - \epsilon \\ \frac{1}{2} + \epsilon \end{bmatrix} = +\epsilon$$

Derivative is positive! $a \leftarrow a + \eta \frac{\partial E[r_A]}{\partial a} = a + \epsilon$! Bob starts away from equilibrium $\Longrightarrow$ Alice is driven to also leave the equlibrium

Does adversarial learning always enter an infinite loop?

No, it has various behaviors depending on the game.

- Paparazzi Game:
  - Start at the equilibrium ⇒ Stays at the equilibrium
  - Start anywhere else ⇒ Enters an infinite loop
- Game of Chicken or Stag Hunt:
  - Start at the mixed equilibrium ⇒ Stays at the equilibrium
  - Start anywhere else ⇒ Converges to one of the two pure-strategy equilibria
- Prisoner's Dilemma (or any game with dominant strategies):
  - Start anywhere ⇒ Converges to the pure-strategy equilibrium

Try the quiz!

Try the quiz!

Outline

- Adversarial learning review
- Mechanism design
- Goals that might motivate mechanism design
- Repeated games

Mechanism design

- Using game theory, we can predict how rational agents will behave
- Suppose we want them to behave in a particular way
- Can we change the rules of the game to get the desired behavior?

Example: Mixed equilibrium

- Suppose we want to Alice and Bob to choose actions with action probabilities given by the vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$.
- Suppose the reward matrices are initialized to

$$\boldsymbol{Q}_A = \begin{bmatrix} q_{A00} & q_{A01} \\ q_{A10} & q_{A11} \end{bmatrix}, \boldsymbol{Q}_B = \begin{bmatrix} q_{B00} & q_{B01} \\ q_{B10} & q_{B11} \end{bmatrix}$$

- Suppose we want to change $\boldsymbol{Q}_A$, $\boldsymbol{Q}_B$ to some new set of reward matrices $\boldsymbol{R}_A$, $\boldsymbol{R}_B$ so that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a Nash equilibrium.
- What is the smallest modification that will make $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ a Nash equilibrium?

		Bob	
		Defect	Cooperate
Alice	Defect	q_{B00} / q_{A00}	q_{B01} / q_{A01}
	Cooperate	q_{B10} / q_{A10}	q_{B10} / q_{A11}

How do we know if it's equilibrium?

- (p_A, p_B) is a Nash equilibrium if

$$\boldsymbol{\alpha}^T \boldsymbol{R}_B \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 0$$

$$[-1,1] \boldsymbol{R}_A \boldsymbol{\beta} = 0$$

- We want to choose $\boldsymbol{R}_A$, $\boldsymbol{R}_B$ that are close to $\boldsymbol{Q}_A$, $\boldsymbol{Q}_B$, but that make those equations true.
- How can we do that?

		Bob: Defect	Bob: Cooperate
Alice	Defect	r_{A00}, r_{B00}	r_{A01}, r_{B01}
Alice	Cooperate	r_{A10}, r_{B10}	r_{A11}, r_{B10}

Solution using gradient descent

One way we can solve this problem is by starting with $\boldsymbol{R}_A = \boldsymbol{Q}_A, \boldsymbol{R}_B = \boldsymbol{Q}_B$, and then gradually improving the fit to the desired Nash equilibrium.

Define the loss to be:

$$\mathcal{L} = \frac{1}{2}\left(\boldsymbol{\alpha}^T \boldsymbol{R}_B \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right)^2 + \frac{1}{2}([-1,1]\boldsymbol{R}_A \boldsymbol{\beta})^2$$

Learn $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ using gradient descent with some step size η:

$$\boldsymbol{R}_A \leftarrow \boldsymbol{R}_A - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{R}_A}$$

$$\boldsymbol{R}_B \leftarrow \boldsymbol{R}_B - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{R}_B}$$

… until we reach $\mathcal{L} = 0$.

Outline

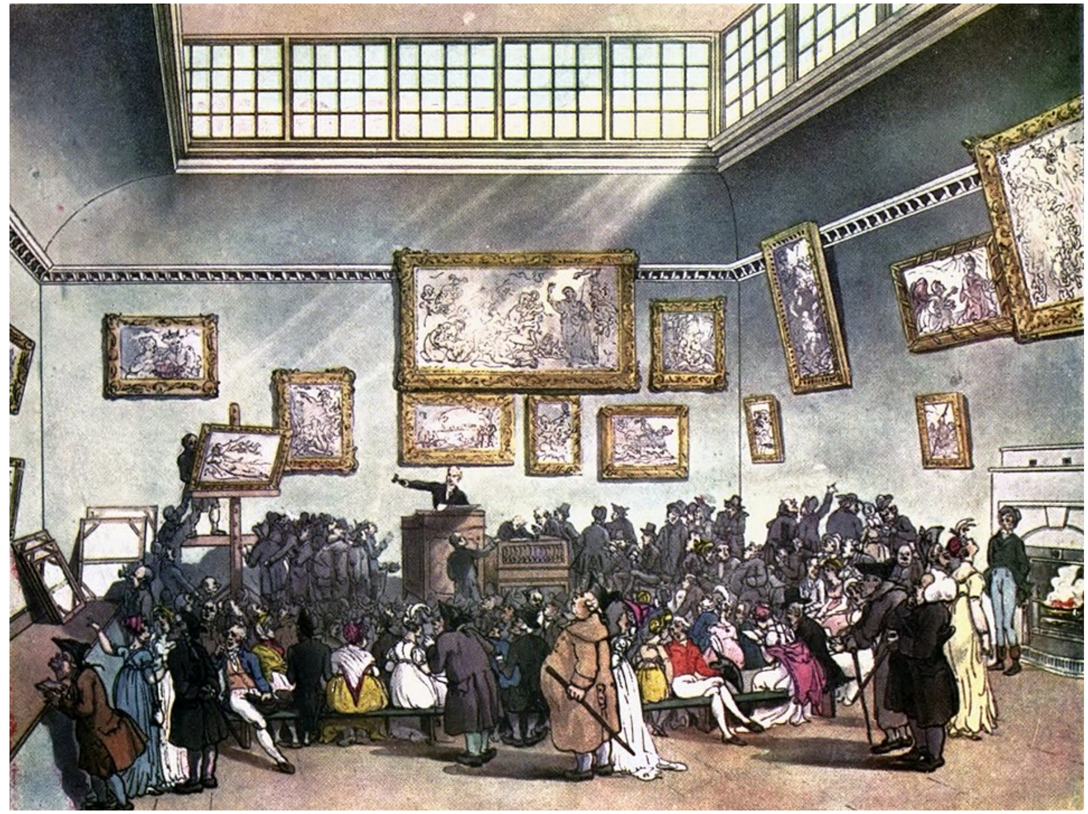
- Adversarial learning review
- Mechanism design
- Goals that might motivate mechanism design
- Repeated games

Goals that might motivate mechanism design

- Goal: Players should always cooperate
  - Design: Rewards that make “C” a dominant strategy
- Goal: Players should always defect
  - Design result: Prisoner’s dilemma
- Goal: Stable Nash equilibrium
  - Design: symmetric rewards
- Goal: Encourage players to choose different responses
  - Design: rewards that discourage similar actions
- Goal: Learn what the reward function is for each player
  - Design: A reward system that encourages revealing your reward function

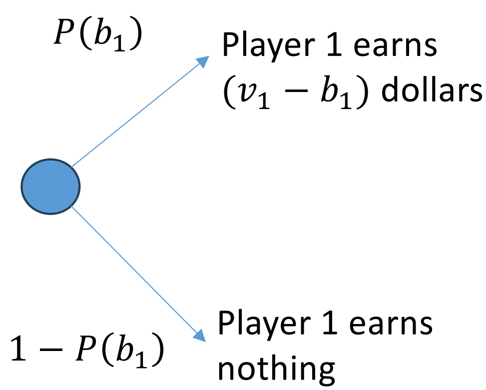
The auction game: Learn the players’ reward functions

- The object being auctioned is worth v_i to the i^{th} player
- The i^{th} player offers to pay b_i for the item (”bid”)
- The auctioneer knows the bid (b_i), not the value (v_i)
- What if auctioneer needs to know the true value of the object?



https://commons.wikimedia.org/wiki/File:Microcosm_of_London_Plate_006_-_Auction_Room,_Christie%27s_(colour).jpg

Nash equilibrium of a classic auction



- If player 1 makes the highest bid, then player 1 pays b_1 dollars for an object of value v_1, thereby earning $v_1 - b_1$
- If their bid is not highest, they neither gain nor lose anything
- Player 1 can guess the bid probabilities of the other players, so he can guess the probability that a bid will win: call that $P(b_1)$. As a function of this guess, the optimum bid is
$$b_1^* = \underset{b_1}{\operatorname{argmax}} E[\text{earn}] = \underset{b_1}{\operatorname{argmax}} P(b_1)(v_1 - b_1)$$
- This quantity is positive only if $b_1 < v_1$!
- The optimum strategy always bids less then the object is worth. You never make a bid equal to the true value.

Recurring auction: Knowledge is worth more than money

- Some resources (oil, advertising) are sold by the same organization once per day (or once per minute)
- The auctioneer wants to know how much the resource is worth
- …and is willing to sacrifice a little revenue to find out

https://commons.wikimedia.org/wiki/File:The_Ladies%27_home_journal_(1948)_(14785694143).jpg

Vickrey auction (second-price auction)

- Player 0 bids b_0 dollars
- Player 1 bids b_1 dollars, $b_1 > b_0$
- Player 1 wins the auction but only pays ***only pays the auctioneer $\boldsymbol{b_0}$ dollars, not $\boldsymbol{b_1}$***. Since the object is worth v_1, player 1 earns $v_1 - \boldsymbol{b_0}$ dollars.
- Player 1's optimum bid is

$$b_1^* = \underset{b_1}{\operatorname{argmax}} E[\text{earn}] = \underset{b_1}{\operatorname{argmax}} P(b_1)(v_1 - \boldsymbol{b_0})$$

- This quantity is positive whenever $v_1 > \boldsymbol{b_0}$.
- Player 1's dominant strategy is to bid $b_1 > b_0$ if and only if $v_1 > b_0$, i.e., the optimum bid is $b_1 = v_1$.
- The optimum strategy for every player is to bid exactly the value, to them, of the object being auctioned, no more, no less.

Outline

- Mechanism design
- Goals that might motivate mechanism design
- Repeated games

Review: Markov decision process

- $s \in \mathcal{S}$: state of the environment (could be int, real, tuple, whatever)
  - $r(s) \in \mathbb{R}$: reward received in state s
  - $u(s) \in \mathbb{R}$: utility of state s = expected discounted sum of all future rewards
- $a \in \mathcal{A}$: action (usually $\mathcal{A}$ is a discrete finite set)
  - $\pi: \mathcal{S} \rightarrow \mathcal{A}$: policy = best action for each state
- The optimum action is given by Bellman's equation:

$$u(s) = r(s) + \gamma \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s')$$

Rewards and utility for simultaneous games

- $\boldsymbol{a} = [a_1, a_2]^T$ = Actions of player 1 and player 2
- $r(s) = [r_1(s), r_2(s)]^T$ = Rewards received by players 1 and 2
- $\boldsymbol{u}(s) = [u_1(s), u_2(s)]^T$ = Utility of state s for player 1, player 2
- $P(s'|s, \boldsymbol{a}, b)$ = Probability of a transition to s' from s if player 1 chooses action a_1 and player 2 chooses action a_2

Bellman's equation for repeated simultaneous two-player games

Bellman's equation becomes something like:

$$\boldsymbol{u}(s) = \boldsymbol{r}(s) + \gamma \max_{\boldsymbol{a}} \sum_{s'} P(S_{t+1} = s' | S_t = s, \boldsymbol{a}) \boldsymbol{u}(s')$$

- But watch out … that "max" is multidimensional! What does that mean?
- It means "find a Nash equilibrium!!"
- Key point, however: The "max" maximizes $\sum_{s'} P(s'|s, \boldsymbol{a}) \boldsymbol{u}(s')$, not $\boldsymbol{r}(s)$; the Nash equilibrium is usually different.

Repeated games and the rational basis for cooperation

The Iterated Prisoner's Dilemma and The Evolution of Cooperation

https://www.youtube.com/watch?v=BOvAbjfJ0x0&t=331s

Summary

- Adversarial learning review

$$\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow \begin{bmatrix} a \\ b \end{bmatrix} + \eta \begin{bmatrix} \frac{\partial E[r_A]}{\partial a} \\ \frac{\partial E[r_B]}{\partial b} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} + \eta \begin{bmatrix} \left(\frac{\partial \boldsymbol{\alpha}}{\partial a}\right)^T \boldsymbol{A}\boldsymbol{\beta} \\ \boldsymbol{\alpha}^T \boldsymbol{B} \frac{\partial \boldsymbol{\beta}}{\partial b} \end{bmatrix}$$

- Mechanism design
  - Design rewards so that rational behavior benefits the game designer
- Second-price auction: Auctioneer sacrifices profit for information

$$b_1^* = \underset{b_1}{\operatorname{argmax}}\, E[\text{earn}] = \underset{b_1}{\operatorname{argmax}}\, P(b_1)(v_1 - \boldsymbol{b_0})$$

- Repeated games (max = find a Nash equilibrium):

$$\boldsymbol{u}(s) = \boldsymbol{r}(s) + \gamma \max_{\boldsymbol{a}} \sum_{s'} P(S_{t+1} = s' | S_t = s, \boldsymbol{a}) \boldsymbol{u}(s')$$