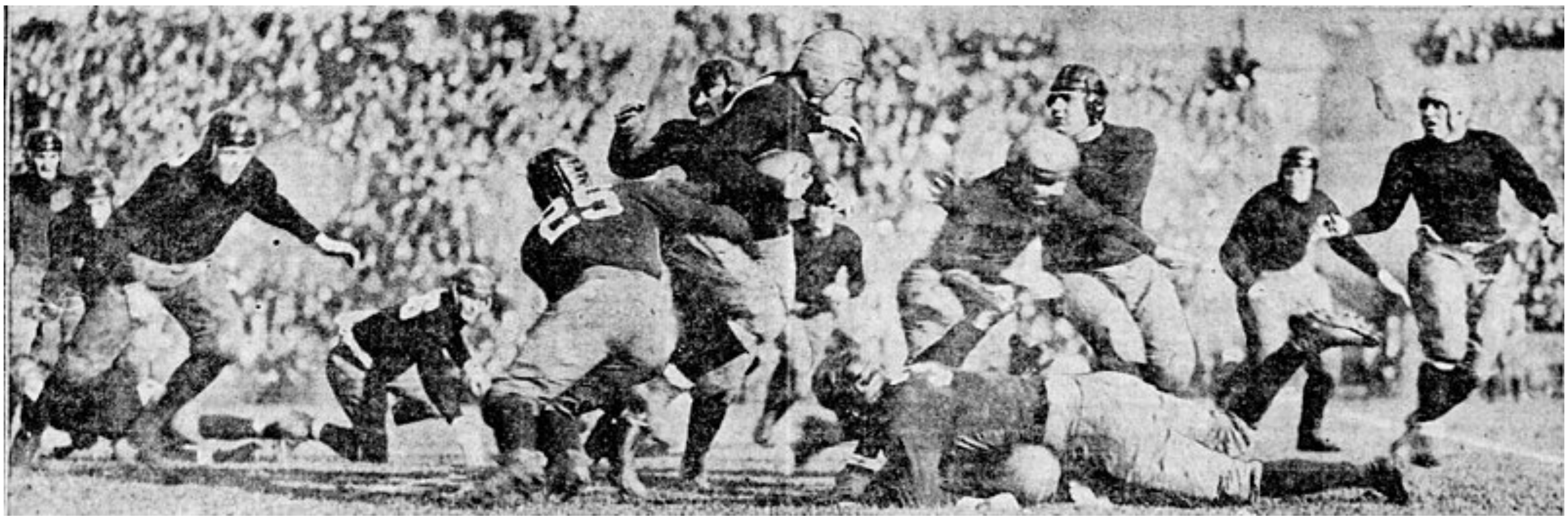
CS 440/ECE448 Lecture 24: Repeated Games

Mark Hasegawa-Johnson, 3/2024

These slides are in the public domain.



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

Outline

- Mathematical foundations
- Repeated games and the rational basis for cooperation
- Learning an episodic game
- Learning a sequential game

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:

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

Review: Expectiminimax

- In a two-player, zero-sum game, each player wins exactly as much as the other player loses.
- The player trying to maximize $u(s)$ is called "Max," the player trying to minimize $u(s)$ is called "Min."
- Bellman's equation:

$$u(s) = \begin{cases} r(s) + \gamma \max_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s') & s \text{ is a max state} \\ r(s) + \gamma \min_a \sum_{s'} P(S_{t+1} = s' | S_t = s, a) u(s') & s \text{ is a min state} \end{cases}$$

Simultaneous games

- If players play simultaneously, then the best strategy might be to choose a move at random (e.g., game of Chicken: make sure opponent can't predict your action with certainty)
- Instead of a scalar $\pi(s)$ =action, we can use

$$\boldsymbol{\pi}(s) = \begin{bmatrix} \pi_1(s) \\ \vdots \\ \pi_{|\mathcal{A}|}(s) \end{bmatrix}, \boldsymbol{\varphi}(s) = \begin{bmatrix} \varphi_1(s) \\ \vdots \\ \varphi_{|\mathcal{B}|}(s) \end{bmatrix}$$

- $\pi_a(s)$ =probability that player 1 chooses action a in state s
- $\varphi_b(s)$ =probability that player 2 chooses action b in state s

$$0 \leq \pi_a(s) \leq 1, \sum_{a=1}^{|\mathcal{A}|} \pi_a(s) = 1, \qquad 0 \leq \varphi_b(s) \leq 1, \sum_{b=1}^{|\mathcal{B}|} \varphi_b(s) = 1,$$

Rewards and utility for simultaneous games

- $r_1(s, a, b)$ = Reward that player 1 receives in state s if player 1 chooses action a and player 2 chooses action b
- $r_2(s, a, b)$ = Reward that player 2 receives in state s if player 1 chooses action a and player 2 chooses action b
- $u_1(s)$ = Utility of state s for player 1
- $u_2(s)$ = Utility of state s for player 2
- $P(s'|s, a, b)$ = Probability of a transition to s' from s if player 1 chooses action a and player 2 chooses action b

Bellman’s equation for repeated simultaneous two-player games

The probability of action a is $\pi_a(s)$, the probability of action b is $\varphi_b(s)$, and the probability of a transition to state s’ is $P(s'|s,a,b)$, so the expected sum of all future rewards under policies $\boldsymbol{\pi}$ and $\boldsymbol{\varphi}$ is:

$$u_1(s) = \sum_{a,b} \pi_a(s)\varphi_b(s)\left(r_1(s,a,b) + \gamma \sum_{s'} P(s'|s,a,b)u_1(s')\right)$$

$$u_2(s) = \sum_{a,b} \pi_a(s)\varphi_b(s)\left(r_2(s,a,b) + \gamma \sum_{s'} P(s'|s,a,b)u_2(s')\right)$$

The best policies, for each player, are:

$$\boldsymbol{\pi}(s) = \operatorname*{argmax}_{\boldsymbol{\pi}} u_1(s)$$

$$\boldsymbol{\varphi}(s) = \operatorname*{argmax}_{\boldsymbol{\varphi}} u_2(s)$$

Outline

- Mathematical foundations
- Repeated games and the rational basis for cooperation
- Learning an episodic game
- Learning a sequential game

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

Image © This Place blog

Outline

- Mathematical foundations
- Repeated games and the rational basis for cooperation
- Learning an episodic game
- Learning a sequential game

Learning an episodic game

Repeated games can be either episodic or sequential:

- Sequential: your actions can change the environment s, which changes the reward $r_1(s, a, b)$ and the other player's strategy $\boldsymbol{\varphi}(s)$
- Episodic: your actions don't change the environment. Your reward is $r_1(a, b)$, your strategy is $\boldsymbol{\pi}$, your opponent's reward is $r_2(a, b)$, and their strategy is $\boldsymbol{\varphi}$.

Repeating an episodic game allows us to iteratively optimize our strategy vector $\boldsymbol{\pi}$, using methods kind of like gradient descent.

Example: The lunch game

- Alice and Bob have agreed that they should always meet at the boat club for lunch.
- Each day, each of them must decide to either:
  - Cooperate: go to the boat club for lunch
  - Defect: go to the museum for lunch
- If they both cooperate, they eat lunch together
- If they both defect, they eat lunch together
- If one cooperates and the other defects, they each eat lunch alone

Public domain
https://commons.wikimedia.org/wiki/File:Pierre-Auguste_Renoir_-_Lunch_at_the_Restaurant_Fournaise_(The_Rowers%27_Lunch)_-_1922.437_-_Art_Institute_of_Chicago.jpg

Public domain
https://commons.wikimedia.org/wiki/File:Berthe_Morisot_-_After_Lunch,_1881.jpg

Example: The lunch game

- Alice prefers reading to talking
  - If they eat lunch together, she gets 1 happiness point
  - If they eat lunch alone, she gets 2 happiness points
- Bob prefers talking to reading
  - If they eat lunch together, he gets 2 happiness points
  - If they eat lunch alone, he gets 1 happiness point

		Bob	
		Defect	Cooperate
Alice	Defect	2 1	1 2
	Cooperate	1 2	2 1

Example: The lunch game

- Alice's strategy is $\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}$
  - $\pi_2 = \frac{1}{1+e^{-x}}$ is the probability she cooperates
  - $\pi_1 = 1 - \pi_2$
- Bob's strategy is $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$
  - $\varphi_2 = \frac{1}{1+e^{-y}}$ is the probability he cooperates
  - $\varphi_1 = 1 - \varphi_2$

Alice \ Bob	Defect	Cooperate
Defect	1, 2	2, 1
Cooperate	2, 1	1, 2

Example: The lunch game

- Alice's strategy is $\boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}$
- Bob's strategy is $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$

		Bob	
		Defect	**Cooperate**
Alice	**Defect**	1, 2	2, 1
	Cooperate	2, 1	1, 2

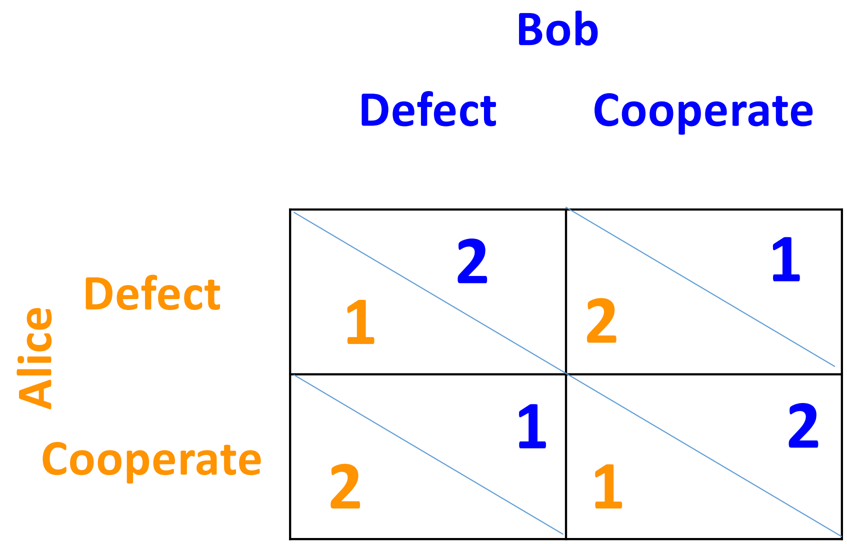
- Alice's expected reward is

$$u_1 = \boldsymbol{\pi}^T \boldsymbol{R_1} \boldsymbol{\varphi} = [\pi_1, \pi_2] \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$$

- Bob's expected reward is

$$u_2 = \boldsymbol{\pi}^T \boldsymbol{R_2} \boldsymbol{\varphi} = [\pi_1, \pi_2] \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$$

The Nash Equilibrium



- Alice's expected reward is

$$u_1 = \boldsymbol{\pi}^T \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \boldsymbol{\varphi}$$

- Bob's expected reward is

$$u_2 = \boldsymbol{\pi}^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \boldsymbol{\varphi}$$

- The Nash equilibrium is:

$$\boldsymbol{\pi} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \qquad \boldsymbol{\varphi} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

…you can verify that this is a Nash equilibrium by noticing that

- If $\boldsymbol{\pi} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$, then Bob has no preference between cooperating and defecting, so he can choose at random.
- If $\boldsymbol{\varphi} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$, then Alice has no preference, and can choose at random.

Simultaneous gradient ascent

$$u_1 = \boldsymbol{\pi}^T \boldsymbol{R_1} \boldsymbol{\varphi}, \qquad u_2 = \boldsymbol{\pi}^T \boldsymbol{R_2} \boldsymbol{\varphi}, \qquad \boldsymbol{\pi} = \begin{bmatrix} e^{-x}/(1+e^{-x}) \\ 1/(1+e^{-x}) \end{bmatrix}, \qquad \boldsymbol{\varphi} = \begin{bmatrix} e^{-y}/(1+e^{-y}) \\ 1/(1+e^{-y}) \end{bmatrix}$$

- Can we use some type of machine learning algorithm to find the "optimum values" of $\boldsymbol{\pi}$ and $\boldsymbol{\varphi}$, i.e., the Nash equilibrium?
- Alice chooses $\boldsymbol{\pi}$ to maximize u_1 for any given $\boldsymbol{\varphi}$
- Bob chooses $\boldsymbol{\varphi}$ to maximize u_2 for any given $\boldsymbol{\pi}$
- One thing we can try is "simultaneous gradient ascent:" adjust x and y in order to maximize both u_1 and u_1 simultaneously:

$$\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \end{bmatrix} + \eta \begin{bmatrix} \boldsymbol{\nabla_x} u_1 \\ \boldsymbol{\nabla_y} u_2 \end{bmatrix}$$

- (…where $\boldsymbol{\nabla_x} u_1$ is general notation for the gradient of u_1 w.r.t. x. In this case, since x is a scalar, $\boldsymbol{\nabla_x} u_1 = \partial u_1 / \partial x$)

Try the quiz!

Try the quiz:
https://us.prairielearn.com/pl/course_instance/147925/assessment/2408126

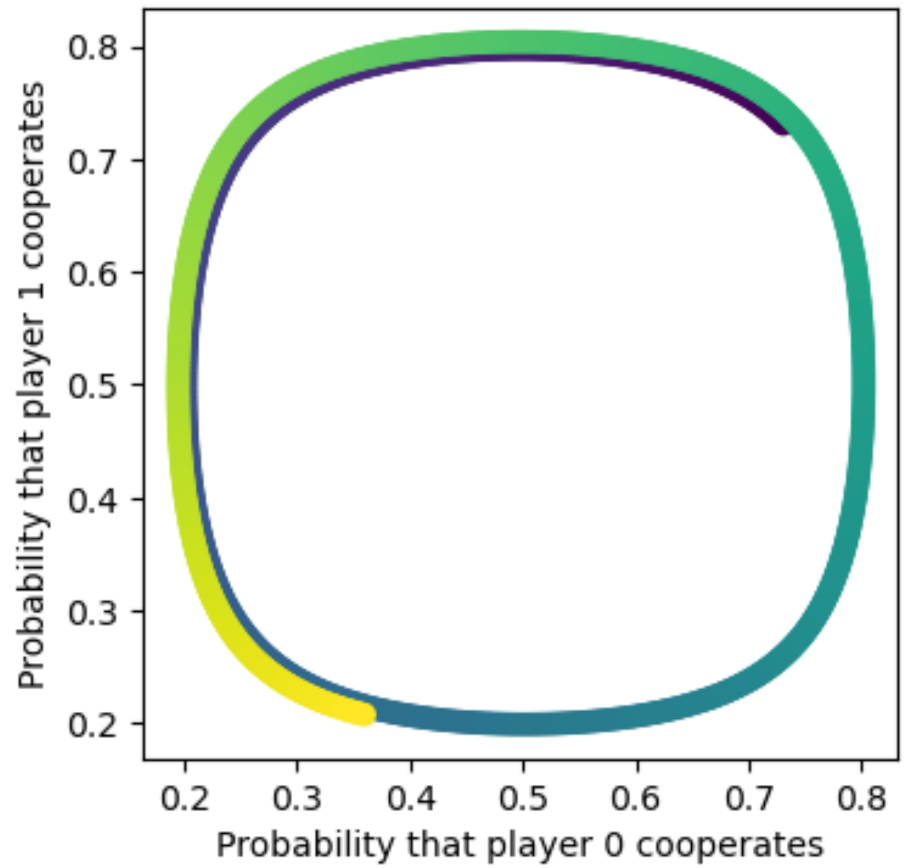
Simultaneous gradient ascent

- Surprisingly, simultaneous gradient ascent fails.
- The graph at right is the sequence of vectors

$$\begin{bmatrix}\pi_2\\ \varphi_2\end{bmatrix} = \begin{bmatrix}1/(1+e^{-x})\\ 1/(1+e^{-y})\end{bmatrix}$$

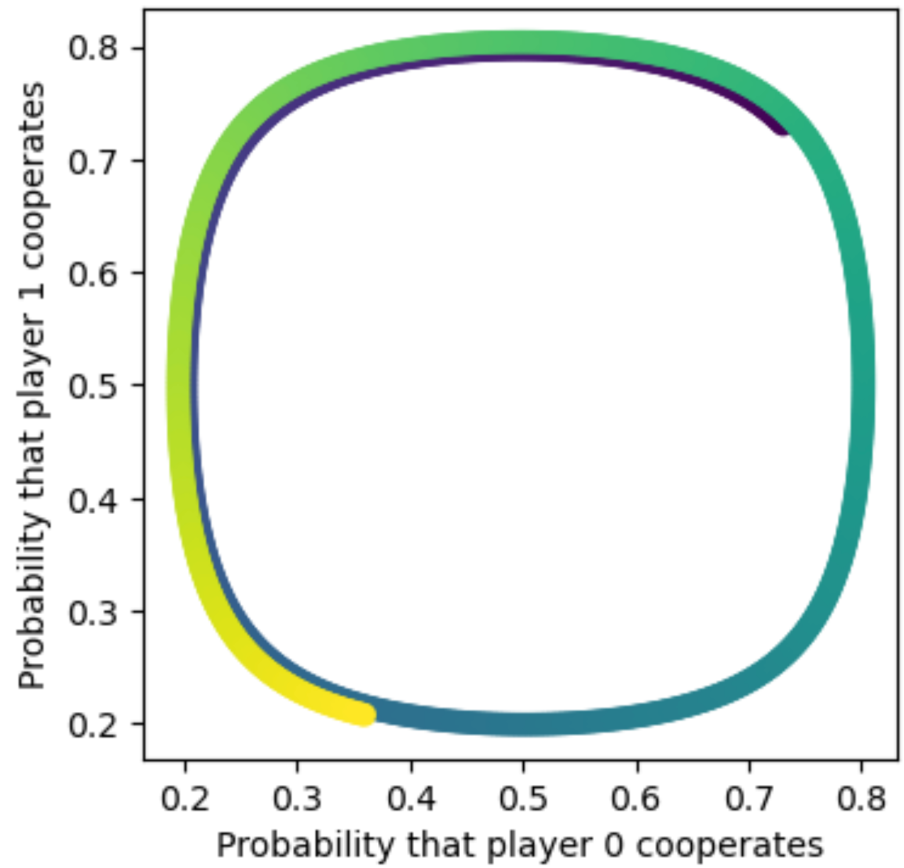
...obtained using

$$\begin{bmatrix}x\\ y\end{bmatrix} \leftarrow \begin{bmatrix}x\\ y\end{bmatrix} + \eta \begin{bmatrix}\partial u_1/\partial x\\ \partial u_2/\partial y\end{bmatrix}$$



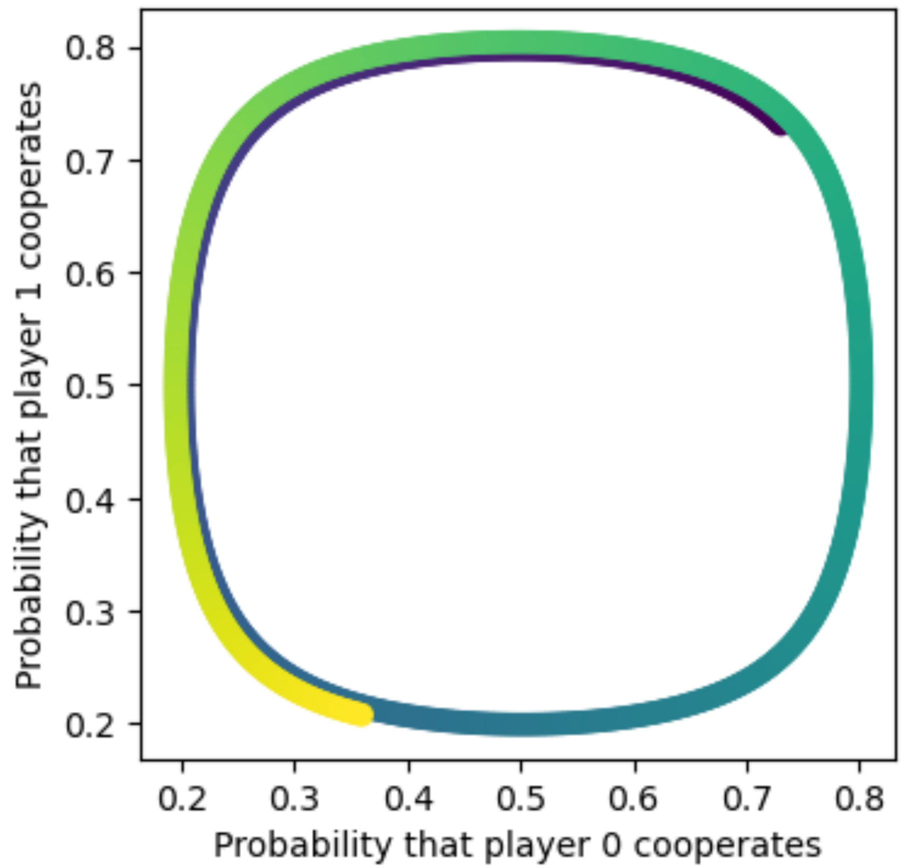
Simultaneous gradient ascent

- Why does it never converge?
- If Bob is at the boat club w/prob $\varphi_2 < 0.5$, then Alice increases x (so she can eat alone more often)
- If Alice is at the boat club w/prob $\pi_2 > 0.5$, then Bob increases y (so he can eat with her more often)
- If Bob is at the boat club w/prob $\varphi_2 > 0.5$, then Alice decreases x (so she can eat alone more often)
- If Alice is at the boat club w/prob $\pi_2 < 0.5$, then Bob decreases y (so he can eat with her more often)
- … and so on, forever.



Digression: orbital mechanics

- This is exactly like the orbit of a spaceship around a planet (the equations are the same)
- We can make it converge the same way we would make a spaceship's orbit decay: apply friction



The symplectic correction

- The solution is to apply friction. The friction term we apply is something that (Balduzzi et al., 2018) called "the symplectic correction " (named after orbital mechanics a.k.a. symplectic mechanics). It looks like this:

$$\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \end{bmatrix} + \eta(\boldsymbol{I} + \boldsymbol{C}) \begin{bmatrix} \partial u_1/\partial x \\ \partial u_2/\partial y \end{bmatrix}$$

- The matrix $\boldsymbol{C}$ is called the symplectic correction. It is $\boldsymbol{C} = \lambda(\boldsymbol{H}^T - \boldsymbol{H})$, where λ is a scalar, and $\boldsymbol{H}$ is something called the Hessian:

$$\boldsymbol{H} = \begin{bmatrix} \partial^2 u_1/\partial x^2 & \partial^2 u_1/\partial x \partial y \\ \partial^2 u_2/\partial x \partial y & \partial^2 u_2/\partial y^2 \end{bmatrix}$$

Corrected gradient ascent

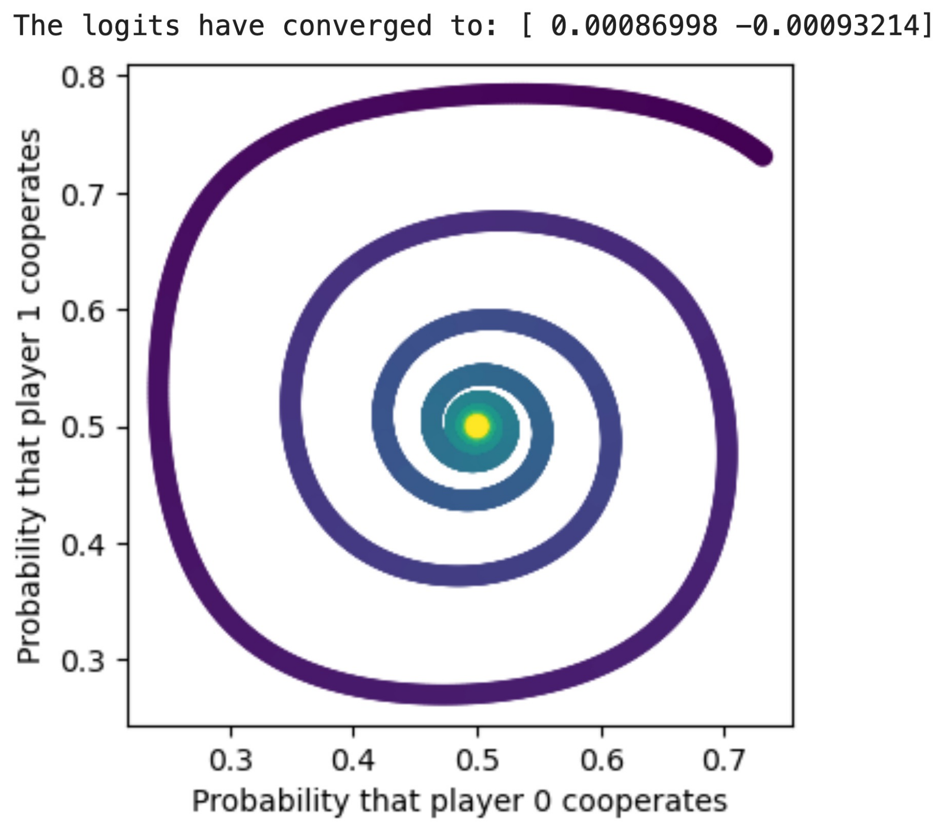
- The graph at right is the sequence of vectors

$$\begin{bmatrix}\pi_2\\ \varphi_2\end{bmatrix} = \begin{bmatrix}1/(1+e^{-x})\\ 1/(1+e^{-y})\end{bmatrix}$$

…obtained using

$$\begin{bmatrix}x\\ y\end{bmatrix} \leftarrow \begin{bmatrix}x\\ y\end{bmatrix} + \eta(\boldsymbol{I}+\boldsymbol{C})\begin{bmatrix}\partial u_1/\partial x\\ \partial u_2/\partial y\end{bmatrix}$$

- As you can see, the correction causes it to “fall” toward the Nash equilibrium at $\boldsymbol{\pi} = \begin{bmatrix}0.5\\ 0.5\end{bmatrix}$, $\boldsymbol{\varphi} = \begin{bmatrix}0.5\\ 0.5\end{bmatrix}$.



Outline

- Mathematical foundations
- Repeated games and the rational basis for cooperation
- Learning an episodic game
- Learning a sequential game

What about tit-for-tat?

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

Image © This Place blog

The Iterated Prisoner's Dilemma and The Evolution of Cooperation

$$\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \end{bmatrix} + \eta(\boldsymbol{I} + \boldsymbol{C}) \begin{bmatrix} \partial u_1/\partial x \\ \partial u_2/\partial y \end{bmatrix}$$

…works if we only want each player to maximize their reward one game at a time, without thinking about future games.

- Strategies like tit-for-tat are a little more complicated: Tit-for-tat remembers how its opponent played last time, and retaliates if its opponent defected.

What about tit-for-tat?

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

Image © This Place blog

The Iterated Prisoner's Dilemma and The Evolution of Cooperation

We can model tit-for-tat by supposing that:

1. the reward matrix depends only on the actions, not the state

$$r_1(s_t, a_t, b_t) = r_1(a_t, b_t)$$
$$r_2(s_t, a_t, b_t) = r_2(a_t, b_t)$$

2. each player remembers a "state" variable consisting of the other player's recent move:

$$s_t = (a_{t-1}, b_{t-1})$$

$$P(S_{t+1} = s' | S_t = s, a, b) = \begin{cases} 1 & s' = (a, b) \\ 0 & \text{otherwise} \end{cases}$$

Under these simplifications, the learning algorithm is four times harder than that of the episodic game: we have to learn $\boldsymbol{\pi}(s_t)$ and $\boldsymbol{\varphi}(s_t)$ separately for each state.

What about tit-for-tat?

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

Image © This Place blog

The Iterated Prisoner's Dilemma and The Evolution of Cooperation

The MP08 extra credit will ask you to create a strategy for a sequential game. You can learn it if you want to, or you can just specify it. For example, the tit-for-tat strategy is

$$\boldsymbol{\pi}(s) = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & b_{t-1} = 1 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & b_{t-1} = 2 \text{ or } t = 1 \end{cases}$$

Conclusions

- Policy probabilities:

$$\pi_a(s) = \Pr(A_t = a | S_t = s)$$
$$\varphi_b(s) = \Pr(B_t = b | S_t = s)$$

$$u_1(s) = \sum_{a,b} \pi_a(s)\varphi_b(s)\left(r_1(s,a,b) + \gamma \sum_{s'} P(s'|s,a,b)u_1(s')\right)$$
$$\boldsymbol{\pi}(s) = \operatorname*{argmax}_{\boldsymbol{\pi}} u_1(s)$$

- Learning episodic games using corrected gradient ascent:

$$\begin{bmatrix} x \\ y \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \end{bmatrix} + \eta(\boldsymbol{I} + \boldsymbol{C}) \begin{bmatrix} \partial u_1/\partial x \\ \partial u_2/\partial y \end{bmatrix}$$