Game Theory 15-451 11/24/09

- Zero-sum games
- General-sum games

Shall we play a game?

Game Theory and Computer Science

Plan for Today

- 2-Player Zero-Sum Games (matrix games)
  - Minimax optimal strategies
  - Minimax theorem and proof
- General-Sum Games (bimatrix games)
  - notion of Nash Equilibrium
- Proof of existence of Nash Equilibria
  - using Brouwer's fixed-point theorem

test material
not test materia

2-player zero-sum game recap

Consider the following scenario…

- Shooter has a penalty shot. Can choose to shoot left or shoot right.
- Goalie can choose to dive left or dive right.
- If goalie guesses correctly, (s)he saves the day. If not, it's a goooooaaaaall!
- Vice-versa for shooter.

2-Player Zero-Sum games

- Two players R and C. Zero-sum means that what's good for one is bad for the other.
- Game defined by matrix with a row for each of R's options and a column for each of C's options. Matrix tells who wins how much.
  - an entry (x,y) means: x = payoff to row player, y = payoff to column player. "Zero sum" means that y = -x.
- E.g., penalty shot:

	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

goalie
shooter
GOAALLL!!!
No goal

Minimax-optimal strategies

- Minimax optimal strategy is a (randomized) strategy that has the best guarantee on its expected gain, over choices of the opponent. [maximizes the minimum]
- I.e., the thing to play if your opponent knows you well.

	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

goalie
shooter
GOAALLL!!!
No goal

Minimax-optimal strategies

- In class on Linear Programming, we saw how to solve for this using LP.
  - polynomial time in size of matrix if use poly-time LP alg.
- I.e., the thing to play if your opponent knows you well.

shooter \ goalie	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

(1,-1): GOAALLL!!! ; (0,0): No goal

Minimax-optimal strategies

- What are the minimax optimal strategies for this game?

Minimax optimal strategy for both players is 50/50. Gives expected gain of ½ for shooter (-½ for goalie). Any other is worse.

shooter \ goalie	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

(1,-1): GOAALLL!!! ; (0,0): No goal

Minimax-optimal strategies

- How about penalty shot with goalie who's weaker on the left?

Minimax optimal for shooter is (2/3,1/3). Guarantees expected gain at least 2/3.
Minimax optimal for goalie is also (2/3,1/3). Guarantees expected loss at most 2/3.

shooter \ goalie	Left	Right
Left	(½,-½)	(1,-1)
Right	(1,-1)	(0,0)

(1,-1): GOAALLL!!! ; (½,-½): 50/50

Shall we play a game...?

I put either a quarter or dime in my hand. You guess. If you guess right, you get the coin. Else you get nothing.

All right!

Summary of game

Value to guesser

guess \ hide	D	Q
D	10	0
Q	0	25

Should guesser always guess Q? 50/50?

What is minimax optimal strategy?

Summary of game

Value to guesser

guess \ hide	D	Q
D	10	0
Q	0	25

If guesser always guesses Q, then hider will hide D. Value to guesser = 0.
If guesser does 50/50, hider will still hide D.
E[value to guesser] = ½(10) + ½(0) = 5

Summary of game

Value to guesser

guess \ hide	D	Q
D	10	0
Q	0	25

If guesser guesses 5/7 D, 2/7 Q, then:

- if hider hides D, E[value] = (5/7)*10 ~ 7.1
- if hider hides Q, E[value] = 50/7 also.

Summary of game

Value to guesser

guess \ hide	D	Q
D	10	0
Q	0	25

What about hider?

Minimax optimal strategy: 5/7 D, 2/7 Q. Guarantees expected loss at most 50/7, no matter what the guesser does.

Interesting. The hider has a (randomized) strategy *he* can reveal with expected loss ≤ 50/7 against any opponent, and the guesser has a strategy *she* can reveal with expected gain ≥ 50/7 against any opponent.

Minimax Theorem (von Neumann 1928)

- Every 2-player zero-sum game has a unique value V.
- Minimax optimal strategy for R guarantees R's expected gain at least V.
- Minimax optimal strategy for C guarantees C's expected loss at most V.

Counterintuitive: Means it doesn't hurt to publish your strategy if both players are optimal. (Borel had proved for symmetric 5x5 but thought was false for larger games)

Can use notion of minimax optimality to explain bluffing in poker

Simplified Poker (Kuhn 1950)

- Two players A and B.
- Deck of 3 cards: 1,2,3.
- Players ante $1.
- Each player gets one card.
- A goes first. Can bet $1 or pass.
  - If A bets, B can call or fold.
  - If A passes, B can bet $1 or pass.
    - If B bets, A can call or fold.
- High card wins (if no folding). Max pot $2.

- Two players A and B. 3 cards: 1,2,3.
- Players ante $1. Each player gets one card.
- A goes first. Can bet $1 or pass.
  - If A bets, B can call or fold.
  - If A passes, B can bet $1 or pass.
    - If B bets, A can call or fold.

Writing as a Matrix Game

- For a given card, A can decide to
  - Pass but fold if B bets. [PassFold]
  - Pass but call if B bets. [PassCall]
  - Bet. [Bet]
- Similar set of choices for B.

Can look at all strategies as a big matrix…

	[FP,FP,CB]	[FP,CP,CB]	[FB,FP,CB]	[FB,CP,CB]
[PF,PF,PC]	0	0	-1/6	-1/6
[PF,PF,B]	0	1/6	-1/3	-1/6
[PF,PC,PC]	-1/6	0	0	1/6
[PF,PC,B]	-1/6	-1/6	1/6	1/6
[B,PF,PC]	-1/6	0	0	1/6
[B,PF,B]	1/6	-1/3	0	-1/2
[B,PC,PC]	1/6	-1/6	-1/6	-1/2
[B,PC,B]	0	-1/2	1/3	-1/6
	0	-1/3	1/6	-1/6

And the minimax optimal strategies are…

- A:
  - If hold 1, then 5/6 PassFold and 1/6 Bet.
  - If hold 2, then ½ PassFold and ½ PassCall.
  - If hold 3, then ½ PassCall and ½ Bet.

  Has both bluffing and underbidding…
- B:
  - If hold 1, then 2/3 FoldPass and 1/3 FoldBet.
  - If hold 2, then 2/3 FoldPass and 1/3 CallPass.
  - If hold 3, then CallBet

Minimax value of game is –1/18 to A.

Matrix games and Algorithms

- Gives a useful way of thinking about guarantees on algorithms for a given problem.
- Think of rows as different algorithms, columns as different possible inputs. (E.g., sorting)
- M(i,j) = cost of algorithm i on input j.
- Algorithm design goal: good strategy for row player. Lower bound: good strategy for adversary.

One way to think of upper-bounds/lower-bounds: on value of this game

Matrix games and Algorithms

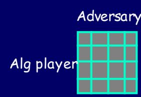
- Gives a useful way of thinking about guarantees on algorithms for a given problem.
- Think of rows as different algorithms, columns as different possible inputs. (E.g., sorting)
- M(i,j) = cost of algorithm i on input j.
- Algorithm design goal: good strategy for row player. Lower bound: good strategy for adversary.

Of course matrix may be HUGE. But helpful conceptually.

Matrix games and Algs

- What is a deterministic alg with a good worst-case guarantee?
  - A row that does well against all columns.
- What is a lower bound for deterministic algorithms?
  - Showing that for each row i there exists a column j such that M(i,j) is bad.
- How to give lower bound for randomized algs?
  - Give randomized strategy for adversary that is bad for all i. Must also be bad for all distributions over i.

E.g., hashing



- Rows are different hash functions.
- Cols are different sets of n items to hash.
- M(i,j) = #collisions incurred by alg i on set j.

We saw:

- For any row, can reverse-engineer a bad column (if universe of keys is large enough).
- Universal hashing is a randomized strategy for row player that has good behavior for every column.
  - For any set of inputs, if you randomly construct hash function in this way, you won't get many collisions in expectation.

We are now below the red line from slide 2

General-Sum Games

- Zero-sum games are good formalism for design/analysis of algorithms.
- General-sum games are good models for systems with many participants whose behavior affects each other's interests
  - E.g., routing on the internet
  - E.g., online auctions

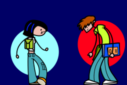
General-sum games

- In general-sum games, can get win-win and lose-lose situations.
- E.g., "what side of sidewalk to walk on?":

person walking towards you

you

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1,-1)	(1,1)



General-sum games

- In general-sum games, can get win-win and lose-lose situations.
- E.g., "which movie should we go to?":

	Bruno	New Moon
Bruno	(8,2)	(0,0)
New Moon	(0,0)	(2,8)

No longer a unique "value" to the game.

Nash Equilibrium

- A Nash Equilibrium is a stable pair of strategies (could be randomized).
- Stable means that neither player has incentive to deviate on their own.
- E.g., "what side of sidewalk to walk on":

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1,-1)	(1,1)

NE are: both left, both right, or both 50/50.

Nash Equilibrium

- A Nash Equilibrium is a stable pair of strategies (could be randomized).
- Stable means that neither player has incentive to deviate on their own.
- E.g., "which movie to go to":

	Bruno	New Moon
Bruno	(8,2)	(0,0)
New Moon	(0,0)	(2,8)

NE are: both B, both NM, or (80/20,20/80)

Uses

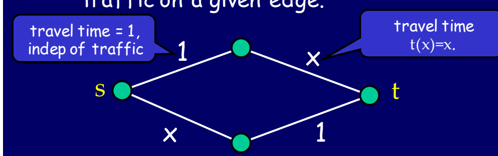
- Economists use games and equilibria as models of interaction.
- E.g., pollution / prisoner's dilemma:
  - (imagine pollution controls cost $4 but improve everyone's environment by $3)

	don't pollute	pollute
don't pollute	(2,2)	(-1,3)
pollute	(3,-1)	(0,0)

Need to add extra incentives to get good overall behavior.

NE can do strange things

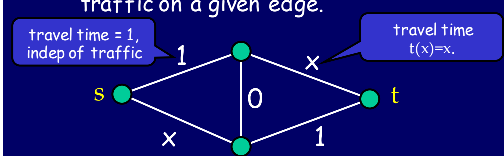
- Braess paradox:
  - Road network, traffic going from s to t.
  - travel time as function of fraction x of traffic on a given edge.



Fine. NE is 50/50. Travel time = 1.5

NE can do strange things

- Braess paradox:
  - Road network, traffic going from s to t.
  - travel time as function of fraction x of traffic on a given edge.



Add new superhighway. NE: everyone uses zig-zag path. Travel time = 2.

Existence of NE

- Nash (1950) proved: any general-sum game must have at least one such equilibrium.
  - Might require randomized strategies (called "mixed strategies")
- This also yields minimax thm as a corollary.
  - Pick some NE and let V = value to row player in that equilibrium.
  - Since it's a NE, neither player can do better even knowing the (randomized) strategy their opponent is playing.
  - So, they're each playing minimax optimal.

Existence of NE

- Proof will be non-constructive.
- Unlike case of zero-sum games, we **do not know any** polynomial-time algorithm for finding Nash Equilibria in $n \times n$ general-sum games. [known to be "PPAD-hard"]
- Notation:
  - Assume an nxn matrix.
  - Use $(p_1,\ldots,p_n)$ to denote mixed strategy for row player, and $(q_1,\ldots,q_n)$ to denote mixed strategy for column player.

Proof

- We'll start with Brouwer's fixed point theorem.
  - Let S be a compact convex region in R^n and let $f: S \rightarrow S$ be a continuous function.
  - Then there must exist $x \in S$ such that $f(x)=x$.
  - x is called a "fixed point" of f.
- Simple case: S is the interval [0,1].
- We will care about:
  - S = {(p,q): p,q are legal probability distributions on 1,...,n}. I.e., S = $\text{simplex}_n \times \text{simplex}_n$

Proof (cont)

- S = {(p,q): p,q are mixed strategies}.
- Want to define f(p,q) = (p',q') such that:
  - f is continuous. This means that changing p or q a little bit shouldn't cause p' or q' to change a lot.
  - Any fixed point of f is a Nash Equilibrium.
- Then Brouwer will imply existence of NE.

Try #1

- What about f(p,q) = (p',q') where p' is best response to q, and q' is best response to p?
- Problem: not necessarily well-defined:
  - E.g., penalty shot: if p = (0.5,0.5) then q' could be anything.

	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

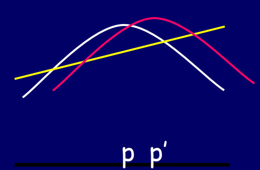
Try #1

- What about f(p,q) = (p',q') where p' is best response to q, and q' is best response to p?
- Problem: also not continuous:
  - E.g., if p = (0.51, 0.49) then q' = (1,0). If p = (0.49,0.51) then q' = (0,1).

	Left	Right
Left	(0,0)	(1,-1)
Right	(1,-1)	(0,0)

Instead we will use...

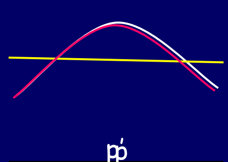
- f(p,q) = (p',q') such that:
  - q' maximizes [(expected gain wrt p) - $||q-q'||^2$]
  - p' maximizes [(expected gain wrt q) - $||p-p'||^2$]



Note: quadratic + linear = quadratic.

Instead we will use...

- f(p,q) = (p',q') such that:
  - q' maximizes [(expected gain wrt p) - $||q-q'||^2$]
  - p' maximizes [(expected gain wrt q) - $||p-p'||^2$]



Note: quadratic + linear = quadratic.

Instead we will use...

- f(p,q) = (p',q') such that:
  - q' maximizes [(expected gain wrt p) - $||q-q'||^2$]
  - p' maximizes [(expected gain wrt q) - $||p-p'||^2$]

- f is well-defined and continuous since quadratic has unique maximum and small change to p,q only moves this a little.
- Also fixed point = NE. (even if tiny incentive to move, will move little bit).
- So, that's it!