10-806 Foundations of Machine Learning and Data Science

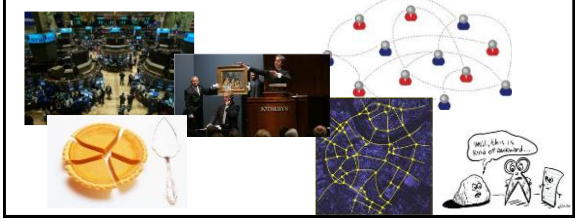
Lecturer: Avrim Blum 10/28/15, 11/2/15

Learning and Game Theory

- Zero-sum games, Minimax Optimality & Minimax Thm; Connection to Boosting & Regret Minimization
- General-sum games, Nash equilibrium and Correlated equilibrium; Internal/Swap Regret Minimization

Game theory

- Field developed by economists to study social & economic interactions.
  - Wanted to understand why people behave the way they do in different economic situations. Effects of incentives. Rational explanation of behavior.



Game theory

- Field developed by economists to study social & economic interactions.
  - Wanted to understand why people behave the way they do in different economic situations. Effects of incentives. Rational explanation of behavior.

- "Game" = interaction between parties with their own interests. Could be called "interaction theory".
- Important for understanding/improving large systems:
  - Internet routing, social networks, e-commerce
  - Problems like spam etc.

Game Theory: Setting

- Have a collection of participants, or *players*.
- Each has a set of choices, or *strategies* for how to play/behave.
- Combined behavior results in *payoffs* (satisfaction level) for each player.

Start by talking about important case of 2-player zero-sum games

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 Row and Col. Zero-sum means that what's good for one is bad for the other.
- Game defined by matrix with row for each of Row's options and a column for each of Col's options. Matrix R gives row player's payoffs, C gives column player's payoffs, R + C = 0.
- E.g., penalty shot [Matrix R]:

shooter \ goalie	Left	Right
Left	0	1
Right	1	0

goalie

shooter

GOAALLL!!!

No goal

Minimax-optimal strategies

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

goalie

shooter

	Left	Right
Left	0	1
Right	1	0

GOAALLL!!!

No goal

Minimax-optimal strategies

- What are the minimax optimal strategies for this game?

Minimax optimal strategy for shooter is 50/50. Guarantees expected payoff $\geq \frac{1}{2}$ no matter what goalie does.
Minimax optimal strategy for goalie is 50/50. Guarantees expected shooter payoff $\leq \frac{1}{2}$ no matter what shooter does.

goalie

shooter

	Left	Right
Left	0	1
Right	1	0

GOAALLL!!!

No goal

Minimax-optimal strategies

- How about for goalie who is 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.

goalie

shooter

	Left	Right
Left	$\frac{1}{2}$	1
Right	1	0

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.

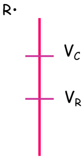
Counterintuitive: Means it doesn't hurt to publish your strategy if both players are optimal. (Borel had proved for symmetric 5x5 but thought was false for larger games)

Minimax-optimal strategies

- Claim: no-regret strategies will do nearly as well or better against any sequence of opponent plays.
  - Do nearly as well as best fixed choice in hindsight.
  - Implies do nearly as well as best distrib in hindsight
  - Implies do nearly as well as minimax optimal!

Proof of minimax thm using RWM

- Suppose for contradiction it was false.
- This means some game G has $V_C > V_R$:
  - If Column player commits first, there exists a row that gets the Row player at least V_C.
  - But if Row player has to commit first, the Column player can make him get only V_R.
- Scale matrix so payoffs to row are in [-1,0]. Say $V_R = V_C - \delta$.



Proof contd

- Now, consider playing randomized weighted-majority alg as Row, against Col who plays optimally against Row's distrib.
- In T steps, in expectation,
  - Alg gets $\geq$ [best row in hindsight] – $2(T\log n)^{1/2}$
  - BRiH $\geq T \cdot V_C$ [Best against opponent's empirical distribution]
  - Alg $\leq T \cdot V_R$ [Each time, opponent knows your randomized strategy]
  - Gap is δT. Contradicts assumption once $\delta T > 2(T\log n)^{1/2}$, or $T > 4\log(n)/\delta^2$.

What if two regret minimizers play each other?

- Then their time-average strategies must approach minimax optimality.

1. If Row's time-average is far from minimax, then Col has strategy that in hindsight substantially beats value of game.
2. So, by Col's no-regret guarantee, Col must substantially beat value of game.
3. So Row will do substantially worse than value.
4. Contradicts no-regret guarantee for Row.

Boosting & game theory

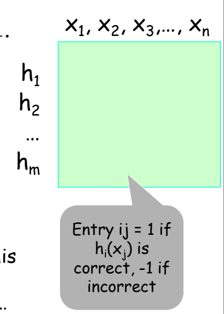
- Suppose I have an algorithm A that for any distribution (weighting fn) over a dataset S can produce a rule $h \in H$ that gets < 45% error.
- Adaboost gives a way to use such an A to get error $\rightarrow 0$ at a good rate, using weighted votes of rules produced.
- How can we see that this is even possible?

Boosting & game theory

- Let's assume the class H is finite.
- Think of a matrix game where columns indexed by examples in S, rows indexed by h in H.
- $M_{ij} = 1$ if $h_i(x_j)$ is correct, else $M_{ij} = -1$.

Boosting & game theory

- Assume for any D over cols, exists row s.t. E[payoff] ≥ 0.1.
- Minimax implies exists a weighting over rows s.t. for every x_i, expected payoff ≥ 0.1.
- So, $sgn(\sum_t \alpha_t h_t)$ is correct on all x_i. Weighted vote has L_1 margin at least 0.1.
- AdaBoost gives you a way to get this with only access via weak learner. But this at least implies existence…



Internal/Swap Regret and Correlated Equilibria

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)

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.

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.

What if all players minimize regret?

- In zero-sum games, empirical frequencies quickly approaches minimax optimal.
- In general-sum games, does behavior quickly (or at all) approach a Nash equilibrium?
  - After all, a Nash Eq is exactly a set of distributions that are no-regret wrt each other. So *if the distributions stabilize*, they must converge to a Nash equil.
- Well, unfortunately, no.

A bad example for general-sum games

- Augmented Shapley game from [Zinkevich04]:
  - First 3 rows/cols are Shapley game (rock / paper / scissors but if both do same action then both lose).
  - 4^{th} action "play foosball" has slight negative if other player is still doing r/p/s but positive if other player does 4^{th} action too.

  RWM will cycle among first 3 and have no regret, but do worse than only Nash Equilibrium of both playing foosball.

- We didn't really expect this to work given how hard NE can be to find…

A bad example for general-sum games

- [Balcan-Constantin-Mehta12]:
  - Failure to converge even in Rank-1 games (games where R+C has rank 1).
  - Interesting because one can find equilibria efficiently in such games.

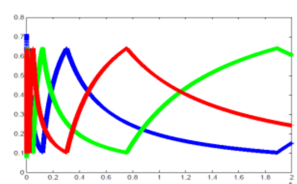


Figure 4. c_is of symmetric Shapley game with $a = 10$, $b = 1$

What *can* we say?

If algorithms minimize "internal" or "swap" regret, then empirical distribution of play approaches *correlated* equilibrium.

- Foster & Vohra, Hart & Mas-Colell,...
- Though doesn't imply play is stabilizing.

What are internal/swap regret and correlated equilibria?

More general forms of regret

1. "best expert" or "external" regret:
   - Given n strategies. Compete with best of them in hindsight.
2. "sleeping expert" or "regret with time-intervals":
   - Given n strategies, k properties. Let S_i be set of days satisfying property i (might overlap). Want to simultaneously achieve low regret over each S_i.
3. "internal" or "swap" regret: like (2), except that S_i = set of days in which we chose strategy i.

Sleeping experts

- A natural generalization of our regret goal (thinking of driving) is: what if we also want that on **rainy** days, we do nearly as well as the best route for **rainy** days.
- And on **Mondays**, do nearly as well as best route for **Mondays**.
- More generally, have N "rules" (on Monday, use path P). Goal: simultaneously, for each rule i, guarantee to do nearly as well as it *on the time steps in which it fires.*
- For all i, want $E[cost_i(alg)] \leq (1+\varepsilon)cost_i(i) + O(\varepsilon^{-1}\log N)$.
  ($cost_i(X)$ = cost of X on time steps where rule i fires.)
- ***Can we get this?***

Sleeping experts algo & analysis (all on one slide)

- Start with all rules at weight 1.
- At each time step, of the rules i that fire, select one with probability $p_i \propto w_i$.
- Update weights:
  - If didn't fire, leave weight alone.
  - If did fire, raise or lower depending on performance compared to weighted average:
    - $r_i = [\sum_j p_j\, cost(j)]/(1+\varepsilon) - cost(i)$
    - $w_i \leftarrow w_i(1+\varepsilon)^{r_i}$
  - So, if rule i does exactly as well as weighted average, its weight drops a little. Weight increases if does better than weighted average by more than a $(1+\varepsilon)$ factor. This ensures sum of weights doesn't increase.
- Final $w_i = (1+\varepsilon)^{E[cost_i(alg)]/(1+\varepsilon)-cost_i(i)}$. So, exponent $\leq \log_{1+\epsilon} N$.
- So, $E[cost_i(alg)] \leq (1+\varepsilon)cost_i(i) + O(\varepsilon^{-1}\log N)$.

Internal/swap-regret

- E.g., each day we pick one stock to buy shares in.
  - Don't want to have regret of the form "every time I bought IBM, I should have bought Microsoft instead".
- Formally, swap regret is wrt optimal function $f:\{1,\ldots,n\}\rightarrow\{1,\ldots,n\}$ such that every time you played action j, it plays f(j).

Weird... why care?

"Correlated equilibrium"

- Distribution over entries in matrix, such that if a trusted party chooses one at random and tells you your part, you have no incentive to deviate.
- E.g., Shapley game.

	R	P	S
R	-1,-1	-1,1	1,-1
P	1,-1	-1,-1	-1,1
S	-1,1	1,-1	-1,-1

In general-sum games, if all players have low swap-regret, then empirical distribution of play is apx correlated equilibrium.

Connection

- If all parties run a low swap regret algorithm, then empirical distribution of play is an apx correlated equilibrium.
  - Correlator chooses random time $t \in \{1,2,\ldots,T\}$. Tells each player to play the action j they played in time t (but does not reveal value of t).
  - Expected incentive to deviate: $\sum_j \Pr(j)(\text{Regret}|j)$ = swap-regret of algorithm
  - So, this suggests correlated equilibria may be natural things to see in multi-agent systems where individuals are optimizing for themselves

Correlated vs Coarse-correlated Eq

In both cases: a distribution over entries in the matrix. Think of a third party choosing from this distr and telling you your part as "advice".

"Correlated equilibrium"

- You have no incentive to deviate, even after seeing what the advice is.

"Coarse-Correlated equilibrium"

- If only choice is to see and follow, or not to see at all, would prefer the former.

Low external-regret $\Rightarrow$ apx coarse correlated equilib.

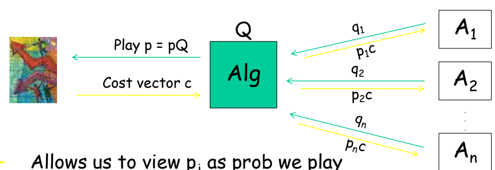
Internal/swap-regret, contd

Algorithms for achieving low regret of this form:

- Foster & Vohra, Hart & Mas-Colell, Fudenberg & Levine.
- Will present method of [BM05] showing how to convert any "best expert" algorithm into one achieving low swap regret.
- Unfortunately, #steps to achieve low swap regret is $O(n \log n)$ rather than $O(\log n)$.

Can convert any "best expert" algorithm A into one achieving low swap regret. Idea:

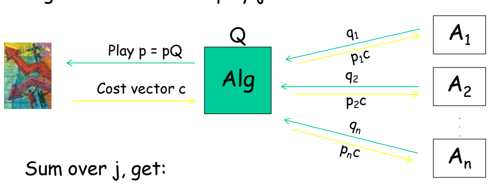
- Instantiate one copy A_j responsible for expected regret over times we play j.



- Allows us to view p_j as prob we play action j, or as prob we play alg A_j.
- Give A_j feedback of $p_j c$.
- A_j guarantees $\sum_t (p_j^t c^t)\cdot q_j^t \le \min_i \sum_t p_j^t c_i^t$ + [regret term]
- Write as: $\sum_t p_j^t (q_j^t \cdot c^t) \le \min_i \sum_t p_j^t c_i^t$ + [regret term]

Can convert any "best expert" algorithm A into one achieving low swap regret. Idea:

- Instantiate one copy A_j responsible for expected regret over times we play j.



- Sum over j, get:

$$\sum_t p^t Q^t c^t \le \sum_j \min_i \sum_t p_j^t c_i^t + n[\text{regret term}]$$

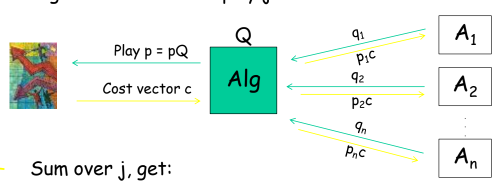
Our total cost

For each j, can move our prob to its own i=f(j)

- Write as: $\sum_t p_j^t (q_j^t \cdot c^t) \le \min_i \sum_t p_j^t c_i^t$ + [regret term]

Can convert any "best expert" algorithm A into one achieving low swap regret. Idea:

- Instantiate one copy A_j responsible for expected regret over times we play j.



- Sum over j, get:

$$\sum_t p^t Q^t c^t \le \sum_j \min_i \sum_t p_j^t c_i^t + n[\text{regret term}]$$

Our total cost

For each j, can move our prob to its own i=f(j)

- Get swap-regret at most n times orig external regret.