Imperfect-Information Games 1

Recap: Normal-Form Games

				
0.2		0	-1	+1
0.5		+1	0	-1
0.3		-1	+1	0

SIMULTANEOUS

(No turns)

Strategy for a player is just a probability distribution over actions

Correlated Equilibrium

- Mediator suggests actions to all players before play
- Correlated equilibrium if **everyone is incentivized to take suggested action**
- NE of stoplight game
  - Two pure strategy NE
  - Mixed NE: Go w/ prob. 1/11
- Correlated equilibrium
  - Could suggest mixture of (Stop, Go) and (Go, Stop)

	Stop	Go
Stop	0, 0	0, 1
Go	1, 0	-10, -10

$$\mathbb{E}_{a\sim D}[u_i(a)] \geq \mathbb{E}_{a\sim D}[u_i(a_i', a_{-i})|a_i]$$

Two-Player Zero-Sum Games

- NE doesn't have problems as in general-sum or multiplayer games
- In a sense, NE is optimal in that no opponent can exploit you
  - If I were to play any other strategy than ⅓, ⅓, ⅓ in rock paper scissors, you could exploit me
- NE can leave utility on the table against imperfect opponents
  - If you always play Rock, NE will still just play ⅓, ⅓, ⅓
- But this is a price usually worth paying when playing experts or other AI programs

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Computing NE in Two-Player Zero-Sum Imperfect Information Games (This Lecture)

1. LP for small games
2. Iterative Approaches
   - Self Play (doesn't converge)
   - Fictitious Play aka Follow the Leader (FTL)
3. No-Regret Algorithms
   - MWU aka Follow the Regularized Leader (FTRL)
   - Regret Matching
4. Optimism

	R	P	S
R	0	-2	1
P	2	0	-1
S	-1	1	0

LP Approach

- Payoff table U
- I choose a distribution s over my pure strategies
- After choosing my distribution, my opponent has expected values for each action given by sU
- Goal is to maximize utility of opponent best response
    - Called exploitability when subtracted from the game value

$$e(x_i) = u_i(x_i^*, x_{-i}^*) - \max_{x'_{-i}} u_i(x_i, x'_{-i})$$

$$e(x) = e(x_1) + e(x_2) = \max_{x'_1} u_1(x'_1, x_2) + \max_{x'_2} u_2(x_1, x'_2)$$

		R	P	S
2/3	R	0	-2	1
1/3	P	2	0	-1
	S	-1	1	0

	R	P	S
EV	2/3	-4/3	1/3

LP Formalization

$$\text{maximize} \quad \boxed{U_1^*}$$

Game value for player 1

$$\text{subject to} \quad \boxed{\sum_{j \in A_1} u_1(a_1^j, a_2^k) \cdot s_1^j} \geq U_1^* \qquad \forall k \in A_2$$

Expected value of action k for player 2

$$\boxed{\begin{array}{l} \displaystyle\sum_{j \in A_1} s_1^j = 1 \\ s_1^j \geq 0 \end{array}} \qquad \forall j \in A_1$$

s is probability simplex for player 1

LP Continued

- Solving our game results in the following
- We maximize the value that the opponent can get against us
- Any deviation would allow the opponent to exploit us more

		R	P	S
1/4	R	0	-2	1
1/4	P	2	0	-1
1/2	S	-1	1	0

	R	P	S
EV	0	0	0

Iterative Approaches

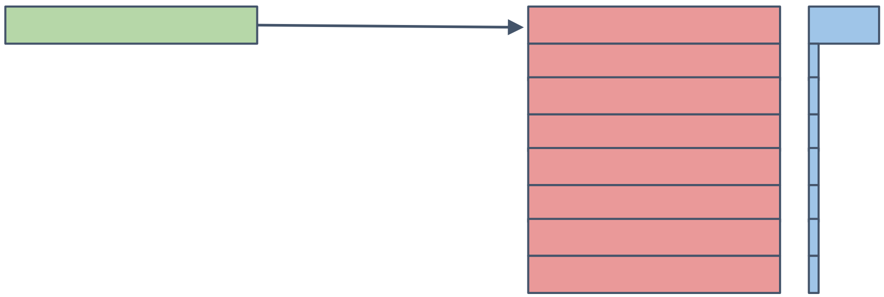
- Only small games can be solved via LP
- For larger games we need iterative approaches
- Most iterative approaches *approach* a NE
  - Can be stopped any time
- What we'll cover
  - Self Play (doesn't converge to NE)
  - Fictitious Play aka Follow the Leader (isn't no-regret)
  - Follow the Regularized Leader aka Replicator Dynamics aka Multiplicative Weights aka Hedge aka Mirror Descent
  - Regret Matching
  - Regret Matching Plus
  - Optimism

Self Play

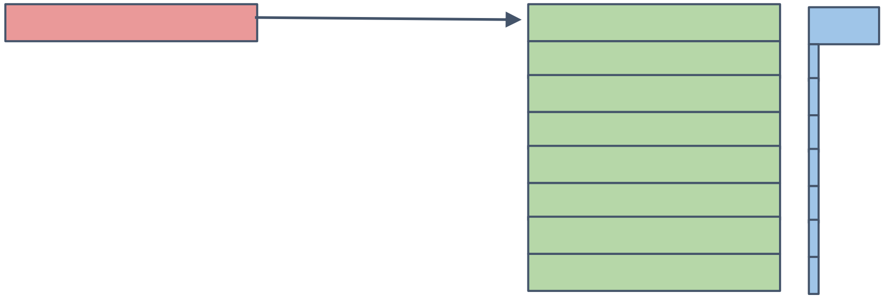
- Both players learn best response to opponent's latest strategy
- Does not converge to a Nash equilibrium even in small games
- Will continue to cycle in games without pure strategy NE

$$x_i^{T+1} = \arg\max_{x_i \in \Delta} u_i(x_i, x_{-i}^T)$$

Player 1 Best Responds to Player 2's Last Policy



Player 2 Best Responds to Player 1's Last Policy



Fictitious Play (Follow the Leader)

- Both players learn best response to opponent's average strategy
- Average strategy converges to a Nash equilibrium

$$x_i^{T+1} = \arg\max_{x_i \in \Delta} \sum_{t=1}^{T} u_i(x_i, x_{-i}^t)$$

No Regret Algorithms

- What if I’m playing a repeated game against someone who knows I am playing fictitious play?
- Then they would know exactly what my next move will be and could choose a best response every time
- Can we find iterative algorithms that will not be *too bad* even when the opponent knows the algorithm?
- No-regret algorithms do exactly this
  - And achieve faster convergence than FP as well!

Regret Minimization

for $t = 1, \dots, T$:

- Agent chooses an *action distribution* $x^t \in X \coloneqq \Delta^n$
- Environment chooses a *utility vector* $u^t \in [0,1]^n$
- Agent observes u^t and gets utility $\langle u^t, x^t \rangle$

Δ^n = set of distributions on n things
$= \{x \in \mathbb{R}^n : x \geq 0, \sum x_i = 1\}$

Agent goal: Minimize *regret*.

"How well do we do against **best, fixed** strategy in hindsight?"

$$R^T \coloneqq \max_{\hat{x} \in X} \boxed{\left\{ \sum_{t=1}^{T} \langle u^t, \hat{x} \rangle \right\}} - \boxed{\sum_{t=1}^{T} \langle u^t, x^t \rangle}$$

Maximum utility that was achievable by the **best fixed** action in hindsight

Utility that was actually accumulated

Goal: have R^T grow sublinearly with respect to time T, e.g., $R^T = O(\sqrt{T})$

No assumption on utilities!
Must handle adversarial environments

What does regret minimization have to do with zero-sum games?

Nash equilibrium in a 2-player 0-sum normal-form game with payoff matrix A:

$$\max_{x \in \Delta^m} \min_{y \in \Delta^n} x^\top A y$$

IDEA: Self-play. Make two regret minimizers play each other

for $t = 1, \dots, T$:

- $x^t \leftarrow$ request strategy from P1's regret minimizer
- $y^t \leftarrow$ request strategy from P2's regret minimizer
- Pass utility Ay^t to P1's regret minimizer
- Pass utility $-A^\top x^t$ to P2's regret minimizer

$$R_1^T \coloneqq \max_{\hat{x} \in \Delta^m} \left\{ \sum_{t=1}^{T} \langle A y^t, \hat{x} \rangle \right\} - \cancel{\sum_{t=1}^{T} \langle A y^t, x^t \rangle} \le O(\sqrt{T})$$

$$R_2^T \coloneqq \max_{\hat{y} \in \Delta^n} \left\{ \sum_{t=1}^{T} \langle -A^\top x^t, \hat{y} \rangle \right\} - \cancel{\sum_{t=1}^{T} \langle -A^\top x^t, y^t \rangle} \le O(\sqrt{T})$$

Add these two lines and divide by T to get the average

$$\max_{\hat{x} \in \Delta^m} \{\hat{x}^\top A \bar{y}\} - \min_{\hat{y} \in \Delta^n} \{\bar{x}^\top A \hat{y}\} \le O\left(\frac{1}{\sqrt{T}}\right)$$

where $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x^t$ and $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y^t$

TAKEAWAY

The average strategies converge to a Nash equilibrium!

Regret Minimization: Follow the Leader

First attempt: Follow the leader. That is, play the best action in hindsight so far:

$$x^{t+1} = \max_{x \in X} \sum_{\tau \leq t} \langle u^\tau, x \rangle$$

This does not work!

Counterexample: $n = 2$ actions,

$$u^t = \begin{cases} [1/2, 0] & t = 1 \\ [0, 1] & t > 1, \text{even} \\ [1, 0] & t > 1, \text{odd} \end{cases}$$

Best action in hindsight has utility $\approx T/2$

Follow-the-leader always plays the wrong action and therefore gets utility ≈ 0

Follow the Regularized Leader

- Add a regularization term
  - E.g. entropy
- This prevents each iterate from being deterministic
- The resulting algorithm is no-regret
- Intuitively, **updates toward high-regret actions, but not too much**

Follow the leader will always play deterministic actions

Follow the regularized leader will mix

$$x_i^{T+1} = \arg\max_{x_i \in \Delta}[\sum_{t=1}^{T} u_i(x_i, x_{-i}^t) + R(x)]$$

Follow the Regularized Leader

- Consider when regularization is entropy

$$x_i^{T+1} = \arg\max_{x_i \in \Delta}[\sum_{t=1}^{T} u_i(x_i, x_{-i}^t) - \sum_a x_i(a)\log(x_i(a))]$$

- Closed-form optimization of this objective results in the following:

$$x_i^{T+1}(a) = \frac{\exp \eta \sum_{t=1}^{T} u_i(a, x_{-i}^t)}{\sum_{a'} \exp \eta \sum_{t=1}^{T} u_i(a', x_{-i}^t)}$$

- Also called Multiplicative Weights Update (MWU), Hedge, Replicator Dynamics, Randomized Weighted Majority

Follow the leader will always play deterministic actions

Follow the regularized leader will mix

A Common Template for Regret Minimizers

- Given utility vectors $u^1, \ldots, u^t$, we compute the empirical regrets up to time t of each action:

$$r^t[a] := \sum_{\tau=1}^{t} u^\tau[a] - \langle u^\tau, x^\tau \rangle$$

- Then, intuitively the next strategy x^{t+1} gives mass to actions in a manner related to how much regret they have accumulated

A Common Template for Regret Minimizers

Empirical regret:

$$r^t[a] \coloneqq \sum_{\tau=1}^{t} u^\tau[a] - \langle u^\tau, x^\tau \rangle$$

Algorithm	Rule
Multiplicative Weights Update (MWU) (Aka Hedge, Replicator Dynamics, FTRL w/ entropy regularization)	$x^{t+1}[a] = \frac{\exp\{\eta r^t[a]\}}{\sum_{a'} \exp\{\eta r^t[a']\}}$
Regret Matching (RM)	$x^{t+1}[a] = \frac{\max\{0, r^t[a]\}}{\sum_{a'} \max\{0, r^t[a']\}}$

Note: MWU is a particular instance of a very general algorithm called “Online mirror descent”, which can be applied to all convex strategy sets and guarantees sublinear regret

- Then, intuitively the next strategy x^{t+1} gives mass to actions somewhat proportionally to how much regret they have accumulated

A Common Template for Regret Minimizers

Empirical regret:

$$r^t[a] \coloneqq \sum_{\tau=1}^{t} u^\tau[a] - \langle u^\tau, x^\tau \rangle$$
$$= r^{t-1}[a] + u^t[a] - \langle u^t, x^t \rangle$$

A simple modification is to, at every iteration, set a floor of 0 on the cumulative regret:

$$r_+^t[a] \coloneqq \max\{0, r_+^{t-1}[a] + u^t[a] - \langle u^t, x^t \rangle\}$$

Algorithm	Rule
Multiplicative Weights Update (MWU) (Aka Hedge, Replicator Dynamics, FTRL w/ entropy regularization)	$x^{t+1}[a] = \frac{\exp\{\eta r^t[a]\}}{\sum_{a'} \exp\{\eta r^t[a']\}}$
Regret Matching (RM)	$x^{t+1}[a] = \frac{\max\{0, r^t[a]\}}{\sum_{a'} \max\{0, r^t[a']\}}$
Regret Matching Plus (RM+)	$x^{t+1}[a] = \frac{\max\{0, r_+^t[a]\}}{\sum_{a'} \max\{0, r_+^t[a']\}}$

A Common Template for Regret Minimizers

All of these algorithms guarantee that after seeing any number T of utilities $u^1, \dots, u^T$, the regret cumulated by the algorithm satisfies

$$R^T \leq c \sqrt{\sum_{t=1}^{T} \|u^t\|_2^2}$$

Constant that depends on number of actions

Remember:
This holds without any assumption about the way the utilities are selected by the environment!

So, assuming that the utility vectors have bounded norms $\|u^t\| \leq B$ (this is always the case when playing finite games), then $R^T \leq cB\sqrt{T}$

Consequence: when using these algorithms in self-play in 2-player 0-sum games, the average strategy converges to a Nash equilibrium at a rate of $\frac{\sqrt{T}}{T} = \frac{1}{\sqrt{T}}$

State-of-the-Art Variant in Practice: Discounted RM (DRM)

- Linear RM (LRM)
  - Weight iteration t by t (in regrets and averaging)
  - RM+ floors regrets at 0. Can we combine this with linear RM? Theory: Yes. Practice: No! Does very poorly.
- But less-aggressive combinations do well: **Discounted RM**
  - On each iteration, multiply positive regrets by t^{α} / (t^{α}+1)
  - On each iteration, multiply negative regrets by t^{β} / (t^{β}+1)
  - Weight contributions toward average strategy on iteration t by t^{γ}
  - Worst-case convergence bound only a small constant worse than that of RM
  - For α = 1.5, β = 0, γ = 2, consistently outperforms RM+ in practice

[Brown & Sandholm, Solving Imperfect-Information Games via Discounted Regret Minimization, AAAI'19]

What Regret Minimizers are Used in Practice?

Multiplicative Weights Update (MWU)

- ✓ Special case of OMD, that works for general convex sets
- ✓ Widely used & understood
- ✗ Slow in practice for games
- ✗ Hyperparameters (stepsize)

- ✓ Can incorporate optimism about future losses to converge faster in 2-player 0-sum games

Regret Matching (RM) & Regret Matching+ (RM+)

- ✗ Only for **simplex** domains
- ✗ Not as well studied
- ✓ Tuned for game solving
- ✓ No hyperparameters
- ✓ Incredibly effective

- ? Unknown... Until recently
- ✓ Modern variants of this, such as DCFR, are the standard in tabular extensive-form game solving!

[Farina et al., Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent, AAAI 2021]

Optimistic Regret Minimizers

Algo	Standard (Non-Optimistic) Rule	Optimistic (Predictive) Rule
MWU	$x^{t+1}[a] = \dfrac{\exp\{\eta r^t[a]\}}{\sum_{a'} \exp\{\eta r^t[a']\}}$	$x^{t+1}[a] = \dfrac{\exp\{\eta(r^t[a] + u^t[a] - \langle u^t, x^t\rangle)\}}{\sum_{a'} \exp\{\eta(r^t[a'] + u^t[a'] - \langle u^t, x^t\rangle)\}}$
RM	$x^{t+1}[a] = \dfrac{\max\{0, r^t[a]\}}{\sum_{a'} \max\{0, r^t[a']\}}$	$x^{t+1}[a] = \dfrac{\max\{0, r^t[a] + u^t[a] - \langle u^t, x^t\rangle\}}{\sum_{a'} \max\{0, r^t[a'] + u^t[a'] - \langle u^t, x^t\rangle\}}$
RM+	$x^{t+1}[a] = \dfrac{\max\{0, r_+^t[a]\}}{\sum_{a'} \max\{0, r_+^t[a']\}}$	$x^{t+1}[a] = \dfrac{\max\{0, r_+^t[a] + u^t[a] - \langle u^t, x^t\rangle\}}{\sum_{a'} \max\{0, r_+^t[a'] + u^t[a'] - \langle u^t, x^t\rangle\}}$

Typically, one-line change in implementation

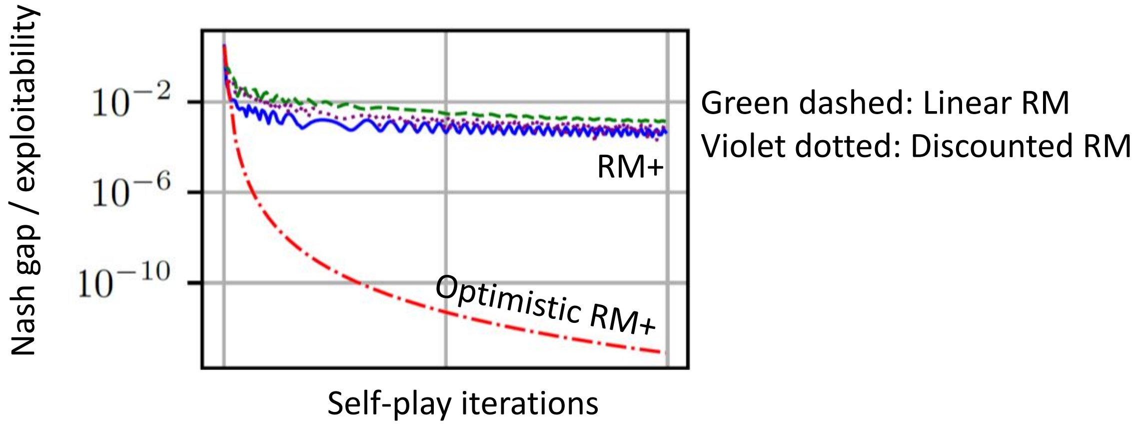
All of these algorithms guarantee that after seeing any number T of utilities $u^1, \dots, u^T$, the regret cumulated by the algorithm satisfies

$$R^T \leq c \sqrt{\sum_{t=2}^{T} \|u^t - u^{t-1}\|_2^2 + (\langle u^t, x^t\rangle - \langle u^{t-1}, x^{t-1}\rangle)^2}$$

Remember:
This holds without any assumption about the way the utilities are selected by the environment!

Takeaway message: still $\approx \sqrt{T}$ regret, but much smaller when there is little change to the utilities over time

Empirical Performance



(RM was omitted as it is typically much slower than RM+)

[Farina, Kroer, and Sandholm; Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent, AAAI'21]

Practical State-of-the-Art

- In general, Discounted RM and Optimistic RM+ are the fastest in practice
  - For some games, like poker, Discounted RM is empirically consistently faster than Optimistic RM+
  - For many other games, Optimistic RM+ is significantly faster

[Farina, Kroer, and Sandholm; Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent, AAAI'21]

References

Blackwell Approachability (used in the correctness proof of RM/RM+):

- [Blackwell, An analog of the minmax theorem for vector payoffs, Pacific J. of Math. 1956]

Regret Matching and Regret Matching Plus:

- [Hart & Mas-Colell, A Simple Adaptive Procedure Leading to Correlated Equilibrium, Econometrica 2000]
- [Tammelin, Solving large imperfect information games using CFR+, ArXiv 2014]
- [Bowling et al., Heads-up Limit Hold'em Poker is Solved, Science 2015]

Predictivity:

- [Chiang et al., Online optimization with gradual variations, COLT 2012]
- [Rakhlin & Sridharan, Online Learning with Predictable Sequences, COLT 2013]
- [Farina et al., Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent, ArXiv 2020]