Announcements

- HW 1 is due now.
- HW 2 will be out in the coming two days.
- Project instruction will be out soon – please start to think about forming teams and thinking about topics
  - Project counts for 50% of the grade

CMSC 35401: The Interplay of Learning and Game Theory
(Autumn 2022)

MW Updates and Implications

Instructor: Haifeng Xu

Outline

- Regret Proof of MW Update
- Convergence to Minimax Equilibrium
- Convergence to Coarse Correlated Equilibrium

Recap: the Model of Online Learning

At each time step $t = 1, \cdots, T$, the following occurs in order:

1. Learner picks a distribution p_t over actions $[n]$
2. Adversary picks cost vector $c_t \in [0,1]^n$
3. Action $i_t \sim p_t$ is chosen and learner incurs cost $c_t(i_t)$
4. Learner observes c_t (for use in future time steps)

- Learner's goal: pick distribution sequence $p_1, \cdots, p_T$ to minimize expected cost $\mathbb{E}_{\forall t:\, i_t \sim p_t} \sum_{t \in [T]} c_t\,(i_t)$
  - Expectation over randomness of action

Measure Algorithms via Regret

- Regret – how much the learner regrets, had he known the cost vector $c_1, \cdots, c_T$ in hindsight
- Formally,

$$R_T = \mathbb{E}_{\forall t:\, i_t \sim p_t} \sum_{t \in [T]} c_t\,(i_t) \boxed{- \min_{i \in [n]} \sum_{t \in [T]} c_t(i)}$$

- Benchmark $\min_{i \in [n]} \sum_t c_t(i)$ is the learner utility had he known $c_1, \cdots, c_T$ and is allowed to take the best single action across all rounds
  - Can also use other benchmarks, but $\min_{i \in [n]} \sum_t c_t(i)$ is mostly used

An algorithm has no regret if $\frac{R_T}{T} \to 0$ as $T \to \infty$, i.e., $R_T = o(T)$.

Regret is an appropriate performance measure of online algorithms

- It measures exactly the loss due to not knowing the data in advance

The Multiplicative Weight Update Alg

Parameter: ϵ

Initialize weight $w_1(i) = 1, \forall i = 1, \cdots n$

For $t = 1, \cdots, T$

1. Let $W_t = \sum_{i \in [n]} w_t(i)$, pick action i with probability $w_t(i)/W_t$
2. Observe cost vector $c_t \in [0,1]^n$
3. For all $i \in [n]$, update $w_{t+1}(i) = w_t(i) \cdot (1 - \epsilon \cdot c_t(i))$

Theorem. MW Update with $\epsilon = \sqrt{\ln n / T}$ achieves regret at most $O(\sqrt{T \ln n})$ for the previously described online learning problem.

- Next, we prove the theorem

Intuition of the Proof

Parameter: ϵ

Initialize weight $w_1(i) = 1, \forall i = 1, \cdots n$

For $t = 1, \cdots, T$

1. Let $W_t = \sum_{i \in [n]} w_t(i)$, pick action i with probability $w_t(i)/W_t$
2. Observe cost vector $c_t \in [0,1]^n$
3. For all $i \in [n]$, update $w_{t+1}(i) = w_t(i) \cdot (1 - \epsilon \cdot c_t(i))$

- Relate decrease of weights to expected cost at each round
  - Expected cost at round t is $\bar{C}_t = \sum_{i \in [n]} p_t(i) \cdot c_t(i) = \frac{\sum_{i \in [n]} w_t(i) \cdot c_t(i)}{W_t}$
  - Propositional to the decrease of total weight at round t, which is

$$\sum_{i \in [n]} \epsilon \cdot w_t(i) c_t(i) = \epsilon W_t \cdot \bar{C}_t$$

- Proof idea: analyze how fast total weights decrease

Proof Step 1: How Fast do Total Weights Decrease?

Lemma 1. $W_{t+1} \leq W_t \cdot e^{-\epsilon \bar{C}_t}$ where $W_t = \sum_{i \in [n]} w_t(i)$ is the total weight at t and $\bar{C}_t$ is the expected loss at time t.

$$\bar{C}_t = \sum_{i \in [n]} p_t(i) c_t(i) = \frac{\sum_{i \in [n]} w_t(i) c_t(i)}{W_t}$$

Proof

➢ Almost Immediate from update rule $w_{t+1}(i) = w_t(i) \cdot (1 - \epsilon \cdot c_t(i))$

$$W_{t+1} = \sum_{i \in [n]} w_{t+1}(i)$$

$$= \sum_{i \in [n]} w_t(i) \cdot (1 - \epsilon \cdot c_t(i))$$

$$= W_t - \epsilon \cdot \sum_{i \in [n]} w_t(i) \cdot c_t(i)$$

$$= W_t - \epsilon \cdot W_t \bar{C}_t = W_t(1 - \epsilon \cdot \bar{C}_t)$$

$$\leq W_t \cdot e^{-\epsilon \cdot \bar{C}_t}$$ since $1 - \delta \leq e^{-\delta}, \forall \delta \geq 0$

Proof Step 1: How Fast do Total Weights Decrease?

Lemma 1. $W_{t+1} \le W_t \cdot e^{-\epsilon \bar{C}_t}$ where $W_t = \sum_{i\in[n]} w_t(i)$ is the total weight at t and $\bar{C}_t$ is the expected loss at time t.

$$\bar{C}_t = \sum_{i\in[n]} p_t(i) c_t(i) = \frac{\sum_{i\in[n]} w_t(i) c_t(i)}{W_t}$$

Corollary 1. $W_{T+1} \le n e^{-\epsilon \sum_{t=1}^{T} \bar{C}_t}$.

$$\begin{aligned} W_{T+1} &\le W_T \cdot e^{-\epsilon \bar{C}_T} \\ &\le [W_{T-1} \cdot e^{-\epsilon \bar{C}_{T-1}}] \cdot e^{-\epsilon \bar{C}_T} \\ &= W_{T-1} \cdot e^{-\epsilon [\bar{C}_T + \bar{C}_{T-1}]} \\ &\quad \cdots \\ &= W_1 \cdot e^{-\epsilon \cdot \sum_{t=1}^{T} \bar{C}_t} \\ &= n \cdot e^{-\epsilon \cdot \sum_{t=1}^{T} \bar{C}_t} \end{aligned}$$

Proof Step 2: Lower Bounding W_{T+1}

Lemma 2. $W_{T+1} \geq e^{-T\epsilon^2} \cdot e^{-\epsilon \sum_{t=1}^{T} c_t(i)}$ for any action i.

$$W_{T+1} \geq w_{T+1}(i)$$

$$= w_1(i)\big(1 - \epsilon c_1(i)\big)\big(1 - \epsilon c_2(i)\big) \dots \big(1 - \epsilon c_T(i)\big)$$ by MW update rule

$$\geq \Pi_{t=1}^{T} e^{-\epsilon c_t(i) - \epsilon^2 [c_t(i)]^2}$$ by fact $1 - \delta \geq e^{-\delta - \delta^2}$

Proof Step 2: Lower Bounding W_{T+1}

Lemma 2. $W_{T+1} \geq e^{-T\epsilon^2} \cdot e^{-\epsilon \sum_{t=1}^{T} c_t(i)}$ for any action i.

$W_{T+1} \geq w_{T+1}(i)$

$= w_1(i)\big(1 - \epsilon c_1(i)\big)\big(1 - \epsilon c_2(i)\big) \ldots \big(1 - \epsilon c_T(i)\big)$ by MW update rule

$\geq \Pi_{t=1}^{T} e^{-\epsilon c_t(i) - \epsilon^2 [c_t(i)]^2}$ by fact $1 - \delta \geq e^{-\delta - \delta^2}$

$\geq e^{-T\epsilon^2} \cdot e^{-\epsilon \sum_{t=1}^{T} c_t(i)}$ relax $[c_t(i)]^2$ to 1

Proof Step 3: Combing the Two Lemmas

Corollary 1. $W_{T+1} \leq n e^{-\epsilon \sum_{t=1}^{T} \bar{C}_t}$.

Lemma 2. $W_{T+1} \geq e^{-T\epsilon^2} \cdot e^{-\epsilon \sum_{t=1}^{T} c_t(i)}$ for any action i.

- Therefore, for any i we have

$$e^{-T\epsilon^2} \cdot e^{-\epsilon \sum_{t=1}^{T} c_t(i)} \leq n e^{-\epsilon \sum_{t=1}^{T} \bar{C}_t}$$

$$\Leftrightarrow \quad -T\epsilon^2 - \epsilon \sum_{t=1}^{T} c_t(i) \leq \ln n - \epsilon \sum_{t=1}^{T} \bar{C}_t$$

take "ln" on both sides

$$\Leftrightarrow \quad \sum_{t=1}^{T} \bar{C}_t - \sum_{t=1}^{T} c_t(i) \leq \frac{\ln n}{\epsilon} + T\epsilon$$

rearrange terms

Taking $\epsilon = \sqrt{\ln n / T}$, we have

$$\sum_{t=1}^{T} \bar{C}_t - \min_i \sum_{t=1}^{T} c_t(i) \leq 2\sqrt{T \ln n}$$

Lower Bound I

$(\ln n)$ term is necessary

- Consider any $T \approx \ln(n-1)$
- Will construct a series of random costs such that there is a perfect action yet any algorithm will have expected cost $T/2$
  - At $t = 1$, randomly pick half actions to have cost 1 and remaining actions have cost 0
  - At $t = 2, 3, \cdots, T$: among perfect actions so far, randomly pick half of them to have cost 1 and remaining actions have cost 0
- Since $T < \ln(n)$, at least one action remains perfect at the end
- But any algorithm suffers expected cost $1/2$ at each round (why?); The total cost will be $T/2$
- Costs are stochastic, not adversarial? → Will be provably worse when costs become adversarial
  - Just FYI: A formal proof is by Yao's minimax principle

Lower Bound 2

$(\sqrt{T})$ term is necessary

- Consider 2 actions only, still stochastic costs
- For $t = 1, \cdots, T$, cost vector $c_t = (0,1)$ or $(1,0)$ uniformly at random
  - c_t's are independent across t's
- Any algorithm has 50% chance of getting cost 1 at each round, and thus suffers total expected cost $T/2$
- What about the best action in hindsight?
  - From action 1's perspective, its costs form a $0-1$ bit sequence, each bit drawn independently and uniformly at random
  - $c[1] = \sum_{t\in T} c_t(1)$ is $Binomial(T, \frac{1}{2})$ and $c(2) = T - c[1]$
  - The cost of best action in hindsight is $\min(c[1], T - c[1])$
  - $\mathbb{E}\min(c[1], T - c[1]) = \frac{T}{2} - \Theta(\sqrt{T})$

Remarks

- Some MW description uses $w_{t+1}\,(i) = w_t(i) \cdot e^{-\epsilon\,\cdot c_t(i)}$. Analysis is similar due to the fact $e^{-\epsilon} \approx 1 - \epsilon$ for small $\epsilon \in [0,1]$
- The same algorithm also works for $c_t \in [-\rho, \rho]$ (still use update rule $w_{t+1}\,(i) = w_t(i) \cdot (1 - \epsilon\ \cdot c_t(i))$). Analysis is the same
- MW update is a very powerful technique – it can also be used to solve, e.g., LP, semidefinite programs, SetCover, Boosting, etc.
  - Because it works for arbitrary cost vectors
  - Next, we show how it can be used to compute equilibria of games where the “cost vector” will be generated by other players

Outline

- Regret Proof of MW Update
- Convergence to Minimax Equilibrium
- Convergence to Coarse Correlated Equilibrium

Online learning – A natural way to play repeated games

Repeated game: the same game played for many rounds

- Think about how you play rock-paper-scissor repeatedly
- In reality, we play like online learning
  - You try to analyze the past patterns, then decide which action to respond, possibly with some randomness
  - This is basically online learning!

Repeated Zero-Sum Games with No-Regret Players

Basic Setup:

- A zero-sum game with payoff matrix $U \in \mathbb{R}^{m \times n}$
- Row player maximizes utility and has actions $[m] = \{1, \cdots, m\}$
  - Column player thus minimizes utility
- The game is played repeatedly for T rounds
- Each player uses an online learning algorithm to pick a mixed strategy at each round

Repeated Zero-Sum Games with No-Regret Players

- From row player's perspective, the following occurs in order at round t
  - Picks a mixed strategy $x_t \in \Delta_m$ over actions in $[m]$
  - Her opponent, the column player, picks a mixed strategy $y_t \in \Delta_n$
  - Action $i_t \sim x_t$ is chosen and row player receives utility $U(i_t, y_t) = \sum_{j \in [n]} y_t(j) \cdot U(i_t, j)$
  - Row player learns y_t (for future use)
- Column player has a symmetric perspective, but will think of $U(i,j)$ as his cost

Difference from online learning: utility/cost vector determined by the opponent, instead of being arbitrarily chosen

Repeated Zero-Sum Games with No-Regret Players

- Expected total utility of row player $\sum_{t=1}^{T} U(x_t, y_t)$
  - Note: $U(x_t, y_t) = \sum_{i,j} U(i,j) x_t(i) y_t(j) = (x_t)^T U y_t$

- Regret of row player is

$$\max_{i \in [m]} \sum_{t=1}^{T} U(i, y_t) - \sum_{t=1}^{T} U(x_t, y_t)$$

- Regret of column player is

$$\sum_{t=1}^{T} U(x_t, y_t) - \min_{j \in [n]} \sum_{t=1}^{T} U(x_t, j)$$

From No Regret to Minimax Theorem

Next, we give another proof of the minimax theorem, using the fact that no regret algorithms exist (e.g., MW update)

From No Regret to Minimax Theorem

- Assume both players use no-regret learning algorithms
- For row player, we have

$$R_T^{row} = \max_{i \in [m]} \sum_{t=1}^{T} U(i, y_t) - \sum_{t=1}^{T} U(x_t, y_t)$$

$$\Leftrightarrow \frac{1}{T} \sum_{t=1}^{T} U(x_t, y_t) + \frac{R_T^{row}}{T} = \frac{1}{T} \max_{i \in [m]} \sum_{t=1}^{T} U(i, y_t)$$

$$= \max_{i \in [m]} U\left(i, \frac{\sum_t y_t}{T}\right)$$

$$\geq \min_{y \in \Delta_n} \max_{i \in [m]} U(i, y)$$

From No Regret to Minimax Theorem

- Assume both players use no-regret learning algorithms
- For row player, we have

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) + \frac{R_T^{row}}{T} \geq \min_{y \in \Delta_n} \max_{i \in [m]} U(i, y)$$

- Similarly, for column player,

$$R_T^{column} = \sum_{t=1}^{T} U(x_t, y_t) - \min_{j \in [n]} \sum_{t=1}^{T} U(x_t, j)$$

implies

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) - \frac{R_T^{column}}{T} \leq \max_{x \in \Delta_m} \min_{j \in [n]} U(x, j)$$

From No Regret to Minimax Theorem

- Assume both players use no-regret learning algorithms
- For row player, we have

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) + \frac{R_T^{row}}{T} \geq \min_{y \in \Delta_n} \max_{i \in [m]} U(i, y)$$

- Similarly, for column player,

$$R_T^{column} = \sum_{t=1}^{T} U(x_t, y_t) - \min_{j \in [n]} \sum_{t=1}^{T} U(x_t, j)$$

implies

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) - \frac{R_T^{column}}{T} \leq \max_{x \in \Delta_m} \min_{j \in [n]} U(x, j)$$

- Let $T \rightarrow \infty$, no regret implies $\frac{R_T^{row}}{T}$ and $\frac{R_T^{column}}{T}$ tend to 0. We have

$$\min_{y \in \Delta_n} \max_{i \in [m]} U(i, y) \leq \max_{x \in \Delta_m} \min_{j \in [n]} U(x, j)$$

From No Regret to Minimax Theorem

- Assume both players use no-regret learning algorithms

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) + \frac{R_T^{row}}{T} \geq \min_{y \in \Delta_n} \max_{i \in [m]} U(i, y)$$

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) - \frac{R_T^{column}}{T} \leq \max_{x \in \Delta_m} \min_{j \in [n]} U(x, j)$$

$$\Rightarrow \min_{y \in \Delta_n} \max_{i \in [m]} U(i, y) \leq \max_{x \in \Delta_m} \min_{j \in [n]} U(x, j)$$

- Recall that min-max $\geq$ max-min also holds, because moving second will not be worse for the row player

Corollary. $\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t)$ converges to the game value

Convergence to Nash Equilibrium

Theorem. Suppose both players use no-regret learning algorithms with action sequence $\{x_t\}$ and $\{y_t\}$. Then $\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t)$ converges to the game value and $(\frac{\sum_{t=1}^{T} x_t}{T}, \frac{\sum_{t=1}^{T} y_t}{T})$ converges to NE of the game.

- Recall that (x^*, y^*) is a NE if and only if x^* is the maximin strategy and y^* is the minimax strategy
- From previous derivations

$$\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t) + \frac{R_T^{row}}{T} = \max_{i\in[m]} U\left(i, \frac{\sum_t y_t}{T}\right)$$

$$\geq \min_{y\in\Delta_n} \max_{i\in[m]} U(i, y)$$

- As $T \to \infty$, "$\geq$" becomes "$=$". So $\frac{\sum_t y_t}{T}$ solves the min-max problem
- Similarly, $\frac{\sum_t x_t}{T}$ solves the max-min problem

Remarks

- If both players use no regret algorithms with $O(\sqrt{T})$, then $\frac{1}{T}\sum_{t=1}^{T} U(x_t, y_t)$ converges to the game value at rate $\frac{R_T}{T} = \frac{1}{\sqrt{T}}$
- This convergence rate can be improved to $\frac{1}{T}$ by careful regularization of the no-regret algorithm
  - More readings: "*Fast Convergence of Regularized Learning in Games*" [NIPS'15 best paper]
  - Intuition: our no-regret algorithm assumes adversarial feedbacks but the other player is not really adversary – he uses another no-regret algorithm
  - This can be exploited to improve learning rate

Remarks

- Convergence of no-regret learning to NE is the key framework for designing the AI agent that beats top humans in Texas hold'em poker
  - Plus many other game solving techniques and engineering work
  - More reading: "*Safe and Nested Subgame Solving for Imperfect-Information Games*." [NeurIPS'17 best paper]

Exciting research is happening at this intersected space of Learning & Game Theory

Outline

- Regret Proof of MW Update
- Convergence to Minimax Equilibrium
- Convergence to Coarse Correlated Equilibrium

Recap: Normal-Form Games and CCE

- n players, denoted by set $[n] = \{1, \cdots, n\}$
- Player i takes action $a_i \in A_i$
- Player utility depends on the outcome of the game, i.e., an action profile $a = (a_1, \cdots, a_n)$
  - Player i receives payoff $u_i(a)$ for any outcome $a \in \Pi_{i=1}^{n} A_i$
- Coarse correlated equilibrium is an action recommendation policy

A recommendation policy π is a **coarse correlated equilibrium** if

$$\sum_{a \in A} u_i(a) \cdot \pi(a) \geq \sum_{a \in A} u_i(a'_i, a_{-i}) \cdot \pi(a), \forall\, a'_i \in A_i, \forall i \in [n].$$

That is, for any player i, following π's recommendations is better than opting out of the recommendation and "acting on his own".

Repeated Games with No-Regret Players

- The game is played repeatedly for T rounds
- Each player uses an online learning algorithm to select a mixed strategy at each round t
- For any player i's perspective, the following occurs in order at t
  - Picks a mixed strategy $x_i^t \in \Delta_{|A_i|}$ over actions in A_i
  - Any other player $j \neq i$ picks a mixed strategy $x_j^t \in \Delta_{|A_j|}$
  - Player i receives expected utility $u_i(x_i^t, x_{-i}^t) = \mathbb{E}_{a \sim (x_i^t, x_{-i}^t)} u_i(a)$
  - Player i learns x_{-i}^t (for future use)

Repeated Games with No-Regret Players

- Expected total utility of player i equals $\sum_{t=1}^{T} u_i(x_i^t, x_{-i}^t)$
- Regret of player i is

$$R_T^i = \max_{a_i \in A_i} \sum_{t=1}^{T} u_i(a_i, x_{-i}^t) - \sum_{t=1}^{T} u_i(x_i^t, x_{-i}^t)$$

From No Regret to CCE

Theorem. Suppose all players use no-regret learning algorithms with strategy sequence $\{x_i^t\}_{t\in[T]}$ for i. The following recommendation policy π^T converges to a CCE: $\pi^T(a) = \frac{1}{T}\sum_t \Pi_{i\in[n]} x_i^t(a_i)$, $\forall\, a \in A$.

Remarks:

- In mixed strategy profile $(x_1^t, x_2^t, \cdots, x_n^t)$, prob of a is $\Pi_{i\in[n]} x_i^t(a_i)$
- $\pi^T(a)$ is simply the average of $\Pi_{i\in[n]} x_i^t(a_i)$ over T rounds

From No Regret to CCE

Theorem. Suppose all players use no-regret learning algorithms with strategy sequence $\{x_i^t\}_{t\in[T]}$ for i. The following recommendation policy π^T converges to a CCE: $\pi^T(a) = \frac{1}{T}\sum_t \Pi_{i\in[n]} x_i^t(a_i)$, $\forall\, a \in A$.

Remarks:

- In mixed strategy profile $(x_1^t, x_2^t, \cdots, x_n^t)$, prob of a is $\Pi_{i\in[n]} x_i^t(a_i)$
- $\pi^T(a)$ is simply the average of $\Pi_{i\in[n]} x_i^t(a_i)$ over T rounds
- Player i's expected utility from π^T is

$$\sum_{a\in A} \left[\frac{1}{T}\sum_t \Pi_{i\in[n]} x_i^t(a_i)\right] \cdot u_i(a)$$

$$= \frac{1}{T}\sum_t \boxed{\sum_{a\in A} \Pi_{i\in[n]} x_i^t(a_i) \cdot u_i(a)}$$

$$= \frac{1}{T}\sum_t \boxed{u_i(x_i^t, x_{-i}^t)}$$

From No Regret to CCE

Theorem. Suppose all players use no-regret learning algorithms with strategy sequence $\{x_i^t\}_{t\in[T]}$ for i. The following recommendation policy π^T converges to a CCE: $\pi^T(a) = \frac{1}{T}\sum_t \Pi_{i\in[n]} x_i^t(a_i), \forall\, a \in A$.

Proof:

- The CCE condition requires for all player i

$$\frac{1}{T}\sum_t u_i(x_i^t, x_{-i}^t) \geq \frac{1}{T}\sum_t u_i\left(a_i, x_{-i}^t\right) \quad \forall a_i \in A_i \qquad (1)$$

- Regret

$$R_T^i = \max_{a_i \in A_i} \sum_{t=1}^T u_i\left(a_i, x_{-i}^t\right) - \sum_{t=1}^T u_i\left(x_i^t, x_{-i}^t\right) \qquad (2)$$

- Dividing Equation (2) by T and let $T \to \infty$ yields Condition (1) since $\lim_{T\to\infty} \frac{R_T^i}{T} \leq 0$ by definition of no regret

Next lecture:

- Study a stronger regret notion called "swap regret" – it uses a stronger benchmark
- Show any game with no-swap-regret players will converge to a correlated equilibrium
- Prove that any no-regret algorithm can be converted to a no-swap-regret algorithm, with slightly worse regret guarantee

Thank You

Haifeng Xu
University of Chicago
haifengxu@uchicago.edu