CSC304 Lecture 6

Game Theory : Minimax Theorem via Expert Learning

2-Player Zero-Sum Games

- Reward of P2 = - Reward of P1
  - Matrix A s.t. $A_{i,j}$ is reward to P1 when P1 chooses her i^{th} action and P2 chooses her j^{th} action
  - Mixed strategy profile $(x_1, x_2) \rightarrow$ reward to P1 is $x_1^T\, A\, x_2$

- Minimax Theorem: For all A,

$$\max_{x_1} \min_{x_2} x_1^T\, A\, x_2 = \min_{x_2} \max_{x_1} x_1^T\, A\, x_2$$

  - Proof through online expert learning!

Online Expert Learning

- Setup:
  - On each day, we want to predict if a stock price will go up or down
  - n experts provide their predictions every day
    - Each expert says either up or down
  - Based on their advice, we make a final prediction
  - At the end of the day, we learn if our prediction was correct (reward = 1) or wrong (reward = 0)

- Goal:
  - Do almost as good as the best expert in hindsight!

Online Expert Learning

- Notation
  - n = #experts
  - Predictions and ground truth: 1 or 0
  - $m_i^{(T)}$ = #mistakes of expert i in first T steps
  - $M^{(T)}$ = #mistakes of the algorithm in first T steps

- Simplest idea:
  - Keep a weight for each expert
  - Use weighted majority of experts to make prediction
  - Decrease the weight of an expert whenever the expert makes a mistake

Online Expert Learning

- Weighted Majority:
  - Fix $\eta \leq 1/2$.
  - Start with $w_i^{(1)} = 1$.
  - In time step t, predict 1 if the total weight of experts predicting 1 is larger than the total weight of experts predicting 0, and vice-versa.
  - At the end of time step t, set $w_i^{(t+1)} \leftarrow w_i^{(t)} \cdot (1-\eta)$ for every expert that made a mistake.

Online Expert Learning

- Theorem: For every i and T,

$$M^{(T)} \leq 2(1+\eta)\, m_i^{(T)} + \frac{2\ln n}{\eta}$$

- Proof:
  - Consider a "potential function" $\Phi^{(t)} = \sum_i w_i^{(t)}$.
  - If the algorithm makes a mistake in round t, at least half of the weight decreases by a factor of $1-\eta$:

$$\Phi^{(t+1)} \leq \Phi^{(t)}\left(\frac{1}{2} + \frac{1}{2}(1-\eta)\right) = \Phi^{(t)}\left(1 - \frac{\eta}{2}\right)$$

Online Expert Learning

- Theorem: For every i and T,

$$M^{(T)} \leq 2(1+\eta)\, m_i^{(T)} + \frac{2\ln n}{\eta}$$

- Proof:
  - $\Phi^{(1)} = n$
  - Thus: $\Phi^{(T+1)} \leq n\left(1-\frac{\eta}{2}\right)^{M^{(T)}}$.
  - Weight of expert i: $w_i^{(T+1)} = (1-\eta)^{m_i^{(T)}}$
  - Use $\Phi^{(T+1)} \geq w_i^{(T+1)}$ and $-\ln(1-\eta) \leq \eta + \eta^2$ (as $\eta \leq 1/2$).

Online Expert Learning

- Beautiful!
  - Comparison to the best expert *in hindsight*.
  - At most (roughly) twice as many mistakes + small additive term
  - In the worst case over how experts make mistakes
    - No statistical assumptions.
  - Simple policy to implement.

- It can be shown that this bound is tight for any deterministic algorithm.

Randomized Weighted Majority

- Randomization $\Rightarrow$ beat the factor of 2
- Simple Change:
  - At the beginning of round t, let
    - $\Phi_1^{(t)}$ = total weight of experts predicting 1
    - $\Phi_0^{(t)}$ = total weight of experts predicting 0
  - Deterministic: predict 1 if $\Phi_1^{(t)} > \Phi_0^{(t)}$, 0 otherwise.
  - Randomized: predict 1 with probability $\frac{\Phi_1^{(t)}}{\Phi_1^{(t)}+\Phi_0^{(t)}}$, 0 with the remaining probability.

Randomized Weighted Majority

- Equivalently:
  - “Pick an expert with probability proportional to weight, and go with their prediction”
  - $\Pr[\text{picking expert } i \text{ in step } t] = p_i^{(t)} = \frac{w_i^{(t)}}{\Phi^{(t)}}$
- Let $b_i^{(t)} = 1$ if expert i makes a mistake in step t, 0 otherwise.
- Algorithm makes a mistake in round t with probability

$$\sum_i p_i^{(t)} b_i^{(t)} = \boldsymbol{p}^{(t)} \cdot \boldsymbol{b}^{(t)}$$

- $E[\text{\#mistakes after } T \text{ rounds}] = \sum_{t=1}^{T} \boldsymbol{p}^{(t)} \cdot \boldsymbol{b}^{(t)}$

Randomized Weighted Majority

$$\Phi^{(t+1)} = \sum_i w_i^{(t+1)} = \sum_i w_i^{(t)} \cdot \left(1 - \eta b_i^{(t)}\right)$$
$$= \Phi^{(t)} - \eta\, \Phi^{(t)} \sum_i p_i^{(t)} \cdot b_i^{(t)}$$
$$= \Phi^{(t)}\left(1 - \eta\, \boldsymbol{p}^{(t)} \cdot \boldsymbol{b}^{(t)}\right)$$
$$\leq \Phi^{(t)} \exp\left(-\eta\, \boldsymbol{p}^{(t)} \cdot \boldsymbol{b}^{(t)}\right)$$

- Applying iteratively:
$$\Phi^{(T+1)} \leq n \cdot \exp(-\eta \cdot E[\#\text{mistakes}])$$
- But $\Phi^{(T+1)} \geq w_i^{(T+1)} \geq (1-\eta)^{m_i^{(T)}}$
- QED!

Randomized Weighted Majority

- Theorem: For every i and T, the expected number of mistakes of randomized weighted majority in the first T rounds is

$$M^{(T)} \leq (1+\eta) m_i^{(T)} + \frac{2 \ln n}{\eta}$$

- Setting $\eta = \sqrt{\frac{\ln n}{T}}$: $M^{(T)} \leq m_i^{(T)} + O\left(\sqrt{T \cdot \ln n}\right)$
- We say that the algorithm has $O\left(\sqrt{T \cdot \ln n}\right)$ regret
- Sublinear regret in T
- Regret per round $\rightarrow 0$ as $T \rightarrow \infty$

How is this related to
the minimax theorem?!!

Minimax via Regret Learning

- Recall:

$$V_R = \max_{x_1} \min_{x_2} x_1^T A\, x_2$$
$$V_C = \min_{x_2} \max_{x_1} x_1^T A\, x_2$$

- Row player's guarantee: my reward $\geq V_R$
- Column player's guarantee: row player's reward $\leq V_C$
- Hence, $V_R \leq V_C$ (trivial direction)
- To prove: $V_R = V_C$

Minimax via Regret Learning

- Scale values in A to be in $[0,1]$.
  - Without loss of generality.

- Suppose for contradiction that $V_R = V_C - \delta, \delta > 0$.

- Suppose row player R uses randomized weighted majority (experts = row player's actions)
  - In each round, column player C responds by choosing her action that minimizes the row player's expected reward.

Minimax via Regret Learning

- After T iterations, row player's reward is:
  - $V \leq T \cdot V_R$
  - $V \geq$ "reward of best action in hindsight" $- O\left(\sqrt{T \cdot \ln n}\right)$
    - Reward of best action in hindsight $\geq T \cdot V_C$.
    - Why?
    - Suppose column player plays action j_t in round t
    - Equivalent to playing mixed strategy s in each round
      - s picks $t \in \{1, \dots, T\}$ at random and plays j_t
    - By definition of V_C, s cannot ensure that row player's reward is less than V_C
      - Then, there is an action of row player with E[reward] at least V_C against s

Minimax via Regret Learning

- After T iterations, row player's reward is:
  - $V \leq T \cdot V_R$
  - $V \geq T \cdot V_C - O\left(\sqrt{T \cdot \ln n}\right)$
  - $T \cdot V_R = T \cdot (V_C - \delta) \geq T \cdot V_C - O\left(\sqrt{T \cdot \ln n}\right)$
  - $\delta\, T \leq O\left(\sqrt{T \cdot \ln n}\right)$
  - Contradiction for sufficiently large T.

- QED!

Yao's Minimax Principle

- Goal:
  - Provide a lower bound on the expected running time that *any* randomized algorithm for a problem can achieve in the worst case over problem instances

- Note:
  - Expectation (in running time) is over randomization of the algorithm
  - The problem instance (worst case) is chosen to maximize this expected running time

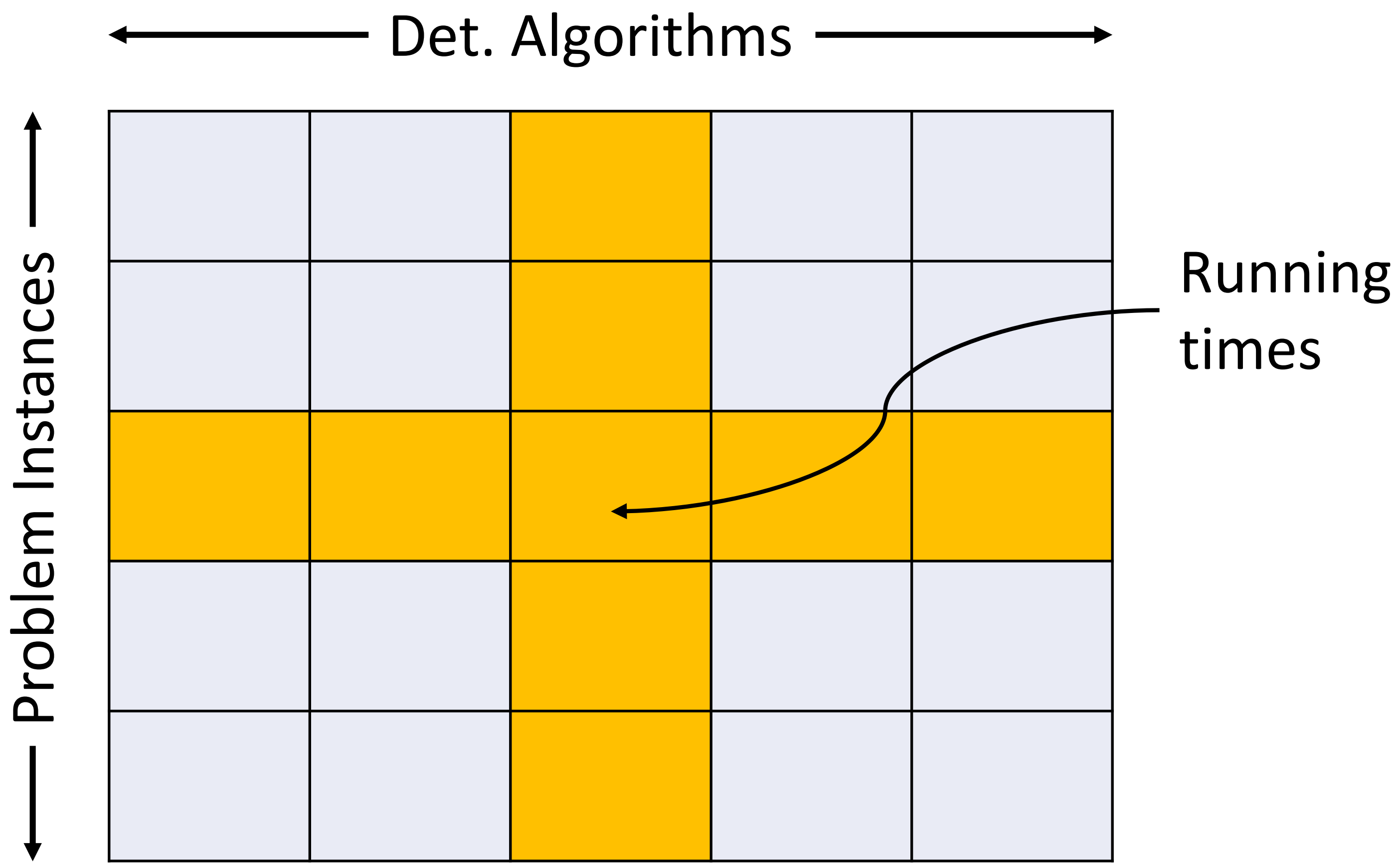
Yao's Minimax Principle

- Notation
  - Capital letters for "randomized", small for deterministic
  - d : a deterministic algorithm
  - R : a randomized algorithm
  - p : a problem instance
  - P : a distribution over problem instances
  - T : running time

- We are interested in

$$\min_R \max_p T(R, p)$$

Yao's Minimax Principle



Yao's Minimax Principle

- Minimax Theorem:

$$\min_R \max_p T(R,p) = \max_P \min_d T(d,P)$$

- So:
  - To lower bound the E[running time] of *any* randomized algorithm R on its worst-case instance p by a quantity Q…
  - Choose a distribution P over problem instances, and show that every det. algorithm d has expected running time at least Q on problems drawn from P