Advanced Algorithms (Fall 2026)

Multiplicative Weight Update（乘法权重更新）

Lecturers: 尹一通，刘景铖，栗师

Nanjing University

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

- Focus of this lecture: learning with experts online（专家建议学习）

- how to dynamically choose from among a set of "experts" in a way that compares favorably to the best expert

- Use it to solve zero-sum game（零和博弈） and linear programs（线性规划） approximately

Learning with Experts Online

- m experts, indexed by $[m]$
- a two-outcome event on each of following T days: up or down
  - Example: stock goes up or down? rain or not?
- on each day t:
  - m experts make predictions about day t
  - algorithm makes a prediction, knowing the predictions of the m experts
  - the outcome of day t reveals
- Goal: minimize the number of mistakes
  - Ideally, not too bad compared to the best expert.

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

A more general setting

- no predictions: experts pay penalties
- algorithm chooses to follow experts
- with randomness: follow a distribution（分布） over experts
- algorithm pays the weighted average penalty of experts
- goal: compare to the best expert

The General Setting

1: **for** $t \leftarrow 1, 2, \cdots, T$ **do**
2: algorithm chooses a distribution $p^t = (p_1^t, p_2^t, \cdots, p_m^t)$ over experts
3: the penalty vector（损失向量） $M^t \in [-1, 1]^m$ is revealed
4: each expert i pays penalty M_i^t
5: algorithm pays penalty $\langle p^t, M^t \rangle = \sum_{i=1}^m p_i^t M_i^t$

- Note: penalty can be negative: negative penalty = reward

Theorem Let $\epsilon \in (0, 1]$. If $T \geq \frac{\ln m}{\epsilon^2}$, then there is an algorithm that satisfies:

$$\frac{1}{T}\sum_{t=1}^{T}\langle p^t, M^t\rangle \leq \frac{1}{T}\sum_{t=1}^{T} M_i^t + 2\epsilon, \forall i \in [m].$$

Algorithm for the General Setting

1: $w_i^0 \leftarrow 1$ for every $i \in [m]$
2: **for** $t \leftarrow 1, 2, \cdots, T$ **do**
3: choose $p^t \leftarrow \frac{w^{t-1}}{|w^{t-1}|_1}$
4: the penalty vector $M^t \in [-1, 1]^m$ is revealed
5: each expert i pays penalty M_i^t
6: algorithm pays penalty $\langle p^t, M^t\rangle = \sum_{i=1}^{m} p_i^t M_i^t$
7: **for** every $i \in [m]$ **do**: $w_i^t \leftarrow w_i^{t-1} \cdot e^{-\epsilon \cdot M_i^t}$

Analysis of Algorithm

- $p_i^t = w_i^{t-1}/\Phi^{t-1}$, where $\Phi^t := \sum_{i=1}^m w_i^t$.
- Using $w_i^t = w_i^{t-1} e^{-\epsilon M_i^t}$,

$$\begin{aligned}\Phi^t &= \Phi^{t-1} \sum_i p_i^t e^{-\epsilon M_i^t} \\ &\leq \Phi^{t-1} \sum_i p_i^t \big(1 - \epsilon M_i^t + \epsilon^2 (M_i^t)^2\big) \\ &\leq \Phi^{t-1} \big(1 - \epsilon \langle p^t, M^t \rangle + \epsilon^2\big).\end{aligned}$$

- $e^z \leq 1 + z + z^2$ for $z \in [-1, 1]$.
- Here $0 < \epsilon \leq 1$, $|M_i^t| \leq 1$, and $\sum_i p_i^t = 1$.

Analysis of Algorithm

$$\begin{aligned}\Phi^t &\leq (1+\epsilon^2)\Phi^{t-1} - \epsilon \cdot \Phi^{t-1} \cdot \langle p^t, M^t \rangle \\ &= \left(1+\epsilon^2 - \epsilon \cdot \langle p^t, M^t \rangle\right) \cdot \Phi^{t-1} \\ &\leq \exp\left(-\epsilon \cdot \langle p^t, M^t \rangle + \epsilon^2\right) \cdot \Phi^{t-1} \\ &\qquad\qquad\qquad\qquad \text{as } 1 - x \leq e^{-x},\ \forall x \in \mathbb{R}\end{aligned}$$

$$\begin{aligned}\Phi^T &\leq \exp\left(\sum_{t=1}^{T}\left(-\epsilon \cdot \langle p^t, M^t \rangle + \epsilon^2\right)\right) \cdot \Phi^0 \\ &= \exp\left(-\epsilon \cdot \sum_{t=1}^{T} \langle p^t, M^t \rangle + T\epsilon^2\right) \cdot \Phi^0\end{aligned}$$

$$\Phi^T \leq \exp\Big(-\epsilon \cdot \sum_{t=1}^{T} \langle p^t, M^t \rangle + T\epsilon^2\Big) \cdot \Phi^0$$

- For any expert $i \in [m]$, $\Phi^T \geq w_i^T = \exp\Big(-\epsilon \sum_{t=1}^{T} M_i^t\Big)$.
- So, for every $i \in [m]$, we have (note that $\Phi^0 = m$)

$$\begin{aligned}
-\epsilon \cdot \sum_{t=1}^{T} M_i^t &\leq -\epsilon \cdot \sum_{t=1}^{T} \langle p^t, M^t \rangle + T\epsilon^2 + \ln m \\
\sum_{t=1}^{T} \langle p^t, M^t \rangle &\leq \sum_{t=1}^{T} M_i^t + T\epsilon + \frac{\ln m}{\epsilon} \\
\frac{1}{T} \sum_{t=1}^{T} \langle p^t, M^t \rangle &\leq \frac{1}{T} \sum_{t=1}^{T} M_i^t + \epsilon + \frac{\ln m}{T\epsilon} \\
&\leq \frac{1}{T} \sum_{t=1}^{T} M_i^t + 2\epsilon \qquad \Big[T \geq \frac{\ln m}{\epsilon^2}\Big]
\end{aligned}$$

Theorem Let $\epsilon \in (0, 1]$. If $T \geq \frac{\ln m}{\epsilon^2}$, then there is an algorithm that satisfies:

$$\frac{1}{T}\sum_{t=1}^{T}\langle p^t, M^t\rangle \leq \frac{1}{T}\sum_{t=1}^{T} M_i^t + 2\epsilon, \forall i \in [m].$$

Coro. Suppose each penalty in the game is in $[-\rho, \rho]$ (instead of $[-1, 1]$) for some $\rho > 0$. Let $\epsilon \in \big(0, 2\rho\big]$. If $T \geq \frac{4\rho^2 \ln m}{\epsilon^2}$, then there is an algorithm that satisfies

$$\frac{1}{T}\sum_{t=1}^{T}\langle p^t, M^t\rangle \leq \frac{1}{T}\sum_{t=1}^{T} M_i^t + \epsilon, \forall i \in [m].$$

Proof.

- scale penalties by $\frac{1}{\rho}$ so that each penalty is in $[-1, 1]$
- $+\epsilon$ before scaling $= +\frac{\epsilon}{\rho}$ after scaling
- Need $T \geq \frac{\ln m}{(\epsilon/2\rho)^2} = \frac{4\rho^2 \ln m}{\epsilon^2}$ and $\frac{\epsilon}{2\rho} \leq 1 \iff \epsilon \leq 2\rho$ □

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

Recall:

Zero-Sum Game

Input: a payoff（收益） matrix（矩阵） $M \in \mathbb{R}^{m \times n}, m, n \geq 1$, two players: row player（行玩家） R, column player（列玩家） C

Output: R plays a row $i \in [m]$, C plays a column $j \in [n]$
payoff of game is M_{ij}
R wants to minimize M_{ij}, C wants to maximize M_{ij}

Rock-Scissor-Paper Game

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

- By scaling, assume $M \in [-1, 1]^{m \times n}$ and $0 < \epsilon \leq 2$.

player	objective（目标函数）	game term	number	distribution
row player	minimize	expert	m	x
column player	maximize	event	n	y

Multiplicative weight update（乘法权重更新） for zero-sum games

let $w_i^0 = 1$ for every $i \in [m]$
for $t \leftarrow 1$ to T, where $T = \max\{1, \lceil 4 \ln m/\epsilon^2 \rceil\}$ **do**
    algorithm chooses distribution $x^t = \frac{w^{t-1}}{|w^{t-1}|_1}$
    let j^t be the $j \in [n]$ that maximizes $M(x^t, j)$
    event j^t happens:
        expert $i \in [m]$ pays penalty $M(i, j^t)$
        algorithm pays penalty $M(x^t, j^t)$
    $w_i^t \leftarrow w_i^{t-1} \cdot e^{-\epsilon \cdot M(i, j^t)/2}$ for every $i \in [m]$

player	objective	game term	number	distribution
row player	minimize	expert	m	x
column player	maximize	event	n	y

- Since $T \geq \frac{4 \ln m}{\epsilon^2}$, we have

$$\frac{1}{T} \sum_{t=1}^{T} M(x^t, j^t) \quad \leq \quad \min_{i \in [m]} \frac{1}{T} \sum_{t=1}^{T} M(i, j^t) + \epsilon$$

- $\hat{t}$: the $t \in [T]$ with minimum $M(x^t, j^t)$ $\qquad \hat{x} := x^{\hat{t}}$
- $\hat{y}$: uniform distribution（均匀分布） over multi-set（多重集）
  $\{j^1, j^2, \cdots, j^T\}$

$$\max_j M(\hat{x}, j) = M(\hat{x}, j^{\hat{t}}) \leq \frac{1}{T} \sum_{t=1}^{T} M(x^t, j^t)$$

$$\leq \min_i \frac{1}{T} \sum_{t=1}^{T} M(i, j^t) + \epsilon = \min_i M(i, \hat{y}) + \epsilon$$

player	objective	game term	number	distribution
row player	minimize	expert	m	x
column player	maximize	event	n	y

$$\max_j M(\hat{x}, j) \leq \min_i M(i, \hat{y}) + \epsilon$$

- λ^*: value of game（博弈值）

$$\lambda^* \leq \max_j M(\hat{x}, j) \leq \min_i M(i, \hat{y}) + \epsilon \leq \lambda^* + \epsilon$$

- Therefore $\hat{x}$ and $\hat{y}$ are approximately the optimum strategies（最优策略） for the row and column players.

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

Linear Program: Exact Version

Input: An "easy" polytope（多胞体） $K \subseteq \mathbb{R}^n$ (e.g., $K = [0,1]^n$)
normal linear constraints（线性约束） $Ax \geq b$, $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

Output: decide if $\{x \in K : Ax \geq b\} = \emptyset$,
if not, then output $x \in K$ with $Ax \geq b$

Linear Program: Approximate Version

Input: An "easy" polytope $K \subseteq \mathbb{R}^n$ (e.g., $K = [0,1]^n$)
normal linear constraints $Ax \geq b$, $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

Output: either claim $\{x \in K : Ax \geq b\} = \emptyset$,
or output $x \in K$ with $Ax \geq b - \epsilon \cdot \mathbf{1}$

- Note: in case there is no exact solution, but an approximate solution, algorithm can respond either way.

What Makes the Polytope K Easy?

- Assume K is nonempty and we can maximize any linear objective（线性目标函数） over K efficiently. A membership test（成员资格判定） alone does not suffice.
- Given $y \geq 0$, compute $x \in \arg\max_{z \in K} y^{\mathrm{T}} A z$.
- If $y^{\mathrm{T}} A x < y^{\mathrm{T}} b$, no point of K satisfies all constraints（约束） $Az \geq b$. Otherwise x satisfies the aggregated constraint（聚合约束）.
- Choose a known width bound（宽度界） $\rho > 0$ with $|A_i x - b_i| \leq \rho$ for every $x \in K$ and every row i.
- The iteration bound is $O(1 + \rho^2 \log(m)/\epsilon^2)$. It is polynomial in ρ/ϵ, which need not be polynomial in the binary input length（二进制输入长度）.

Approximate LP Solver using MWU

row of A	constraint	expert	dual solution （対偶解） y	m
column of A	variable		primal solution （原始解） x	n

- event = a point in K
- choose $\rho > 0$ with $|A_i x - b_i| \leq \rho$ for all $x \in K$, $i \in [m]$; take $0 < \epsilon \leq 2\rho$

1: $w_i^0 \leftarrow 1$ for every $i \in [m]$
2: **for** $t \leftarrow 1$ to T, for some T to be decided later **do**
3: $\quad y^t \leftarrow \frac{w^{t-1}}{|w^{t-1}|_1}$
4: $\quad$ **if** $\exists x^t \in K$ s.t $\langle y^t, Ax^t \rangle \geq \langle y^t, b \rangle$ **then** ▷ event x^t happens
5: $\quad\quad$ **for** every $i \in [m]$ **do**
6: $\quad\quad\quad$ expert i gets penalty $A_i x^t - b_i$
7: $\quad\quad\quad$ $w_i^t \leftarrow w_i^{t-1} \cdot e^{-\epsilon \cdot (A_i x^t - b_i)/(2\rho^2)}$
8: $\quad\quad$ our algorithm gets penalty $\langle y^t, Ax^t - b \rangle$
9: $\quad$ **else return** "empty"
10: **return** $\hat{x} = \frac{1}{T} \sum_{t=1}^{T} x^t$

- Counter-intuitive: the more satisfied a constraint is, the more penalty it gets.
- In every iteration, we only need to focus on one "aggregated" linear constraint
- If algorithm returns "empty", then the LP is not feasible

- $\rho > 0: \quad \rho \geq \sup_{x \in K} \max_i |A_i x - b_i|, \qquad \epsilon \in (0, 2\rho]$
  $T := \max\{1, \lceil 4\rho^2 \ln m/\epsilon^2 \rceil\}$
- $\forall i \in [m]$:

$$0 \leq \frac{1}{T}\sum_{t=1}^{T}\langle y^t, Ax^t - b\rangle \leq \frac{1}{T}\sum_{t=1}^{T}(A_i x^t - b_i) + \epsilon = A_i\hat{x} - b_i + \epsilon$$

- Therefore, $A_i\hat{x} \geq b_i - \epsilon, \forall i \in [m] \quad \Longleftrightarrow \quad A\hat{x} \geq b - \epsilon \cdot \mathbf{1}$

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

Outline

1. Learning with Expert Advice（专家建议学习）
2. The Multiplicative Weight Update Theorem
3. Solving Zero-Sum Games（零和博弈）
4. Approximate LP Feasibility（近似线性规划可行性）
5. Optional Material
   - Halving（折半算法） and Weighted Majority（加权多数算法）

When There Is a Perfect Expert

Lemma There is an algorithm that makes at most $\lceil \log_2 m \rceil$ mistakes, assuming there is a perfect expert, i.e., an expert that makes no mistakes.

Proof.

- The algorithm: only keep the experts who made no mistakes so far.
- Among all the experts, follow the majority.
- observation: when we made a mistake on a day, at least half of the remaining experts made a mistake on that day. □

General Case

- What if there is no perfect expert?
- Weighted majority（加权多数算法）: give weights to experts. When an expert made a mistake, halve his weight. In each step, follow the weighted majority.

Weighted Majority

1: $w_i^0 \leftarrow 1, \forall i \in [m]$
2: **for** $t = 1, 2, \cdots, T$ **do**
3: **if** $\sum_{i:\text{predicts up}} w_i^{t-1} \geq \sum_{i:\text{predicts down}} w_i^{t-1}$ **then**
4: predict "up"
5: **else**
6: predict "down"
7: **for** every expert i **do**
8: **if** i make a mistake **then** $w_i^t \leftarrow \frac{w_i^{t-1}}{2}$ **else** $w_i^t \leftarrow w_i^{t-1}$

Analysis

- $\Phi^t := \sum_{i=1}^m w_i^t$.
- $M_i^t \in \{0,1\}, i \in [m], t \in [T]$: whether expert i made a mistake on day t
- $M^t \in \{0,1\}, t \in [T]$: whether our algorithm made a mistake

Obs. If $M^t = 1$ for some t, we have

$$\Phi^t \leq \Phi^{t-1} - \frac{\Phi^{t-1}}{4} = \frac{3}{4}\Phi^{t-1}$$

- So, $\Phi^T \leq (\frac{3}{4})^{\sum_{t=1}^T M^t} \Phi^0 = (\frac{3}{4})^{\sum_{t=1}^T M^t} m$.
- On the other hand, let k be the best expert

$$\Phi^T = \sum_{i=1}^m w_i^T \geq w_k^T \geq \left(\frac{1}{2}\right)^{\sum_{t=1}^T M_k^t}$$

$$\Phi^T \leq \left(\frac{3}{4}\right)^{\sum_{t=1}^{T} M^t} m \qquad \Phi^T \geq \left(\frac{1}{2}\right)^{\sum_{t=1}^{T} M_k^t}$$

$$\left(\frac{1}{2}\right)^{\sum_{t=1}^{T} M_k^t} \leq \Phi^T \leq \left(\frac{3}{4}\right)^{\sum_{t=1}^{T} M^t} m$$

$$(-\ln 2) \sum_{t=1}^{T} M_k^t \leq (-\ln \frac{4}{3}) \sum_{t=1}^{T} M^t + \ln m \qquad \text{(take logs)}$$

$$\sum_{t=1}^{T} M^t \leq \frac{\ln 2}{\ln 4/3} \sum_{t=1}^{T} M_k^t + \frac{\ln m}{\ln 4/3}$$

$$\leq 2.41 \sum_{t=1}^{T} M_k^t + 3.48 \ln m$$

$$\sum_{t=1}^{T} M^t \leq 2.41 \sum_{t=1}^{T} M_k^t + 3.48 \ln m$$

Make the first constant arbitrarily close to 2

- for $0 < \epsilon < 1$, when expert i makes a mistake on day t: $w_i^t \leftarrow w_i^{t-1} \cdot (1 - \epsilon)$
- when algorithm makes a mistake on day t: $\Phi^t \leq \Phi^{t-1} \cdot (1 - \frac{\epsilon}{2})$

$$\Phi^T \leq (1 - \epsilon/2)^{\sum_{t=1}^{T} M^t} \cdot m$$

- $\Phi^T \geq (1 - \epsilon)^{\sum_{t=1}^{T} M_k^t}$

$$\sum_{t=1}^{T} M^t \leq \frac{\ln(1-\epsilon)}{\ln(1-\epsilon/2)} \sum_{t=1}^{T} M_k^t - \frac{\ln m}{\ln(1-\epsilon/2)}$$

$$= (2 + O(\epsilon)) \sum_{t=1}^{T} M_k^t + O\left(\frac{1}{\epsilon}\right) \ln m.$$

Lemma For any constant $\epsilon > 0$ and any function $f(m)$, no deterministic algorithm（確定性算法） can achieve a multiplicative factor of $2 - \epsilon$ and additive factor of $f(m)$.

Proof.

- Take $m = 2$: one expert always predicts "up", the other always predicts "down".
- Choose each outcome opposite to the deterministic algorithm's prediction.
- Our algorithm made T mistakes, and the best expert makes at most $T/2$.
- If $T \gg f(m)$, we have $T > (2 - \epsilon) \cdot \frac{T}{2} + f(m)$. □

- However, if randomness is allowed, we can make multiplicative factor $1 + \epsilon$.

Appendix: 英汉术语对照 (1/1)

English	中文	English	中文
aggregated constraint	聚合约束	approximate LP feasibility	近似线性规划可行性
binary input length	二进制输入长度	column player	列玩家
constraint	约束	deterministic algorithm	确定性算法
distribution	分布	dual solution	对偶解
halving	折半算法	learning with expert advice	专家建议学习
linear constraint	线性约束	linear objective	线性目标函数
linear programming	线性规划	matrix	矩阵
membership test	成员资格判定	multiplicative weight update	乘法权重更新
multiset	多重集	objective function	目标函数
optimal strategy	最优策略	payoff	收益
penalty vector	损失向量	polytope	多胞体
primal solution	原始解	row player	行玩家
uniform distribution	均匀分布	value of the game	博弈值
weighted majority	加权多数算法	width bound	宽度界
zero-sum game	零和博弈		