DS 320 Algorithms for Data Science	**Lecture #21 Worksheet**
Spring 2024	Prof. Kira Goldner

Zero-Sum Games and the Minimax Theorem

Notation:

- $m \times n$ payoff matrix $\mathbf{A}$—a_{ij} is the row player's payoff for outcome (i, j) when row player plays strategy i and column player plays strategy j
- mixed row strategy $\mathbf{x}$ (a distribution over rows)
- mixed column strategy $\mathbf{y}$ (a distribution over columns)

Expected payoff of the row player:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \Pr[\text{outcome } (i,j)]\, a_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} \underbrace{\Pr[\text{row } i \text{ chosen}]}_{=x_i}\, \underbrace{\Pr[\text{column } j \text{ chosen}]}_{=y_j}\, a_{ij}$$
$$= \mathbf{x}^T \mathbf{A}\mathbf{y}$$

Theorem 1 (Minimax Theorem)**.** *For every two-player zero-sum game* $\mathbf{A}$*,*

$$\max_{\mathbf{x}} \left(\min_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y} \right) = \min_{\mathbf{y}} \left(\max_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{y} \right). \tag{1}$$

Or in English, the expected payoff of the row play is the same whether the row player goes first or second. This is called the *value of the game.*

From LP Duality to Minimax

$$\max_{\mathbf{x}} \left(\min_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y} \right) = \max_{\mathbf{x}} \left(\min_{j=1}^{n} \mathbf{x}^T \mathbf{A} \mathbf{e}_j \right) \tag{2}$$
$$= \max_{\mathbf{x}} \left(\min_{j=1}^{n} \sum_{i=1}^{m} a_{ij} x_i \right) \tag{3}$$

where $\mathbf{e}_j$ is the j^{th} standard basis vector:

$$(\mathbf{e}_j)_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

$$\max v$$

subject to

$$\begin{aligned} v - \sum_{i=1}^{m} a_{ij} x_i &\leq 0 \quad \text{for all } j = 1, \ldots, n \\ \sum_{i=1}^{m} x_i &= 1 \\ x_1, \ldots, x_m &\geq 0 \quad \text{and} \quad v \in \mathbb{R}. \end{aligned}$$

$$\min w$$

subject to

$$\begin{aligned} w - \sum_{j=1}^{n} a_{ij} y_j &\geq 0 \quad \text{for all } i = 1, \ldots, m \\ \sum_{j=1}^{n} y_j &= 1 \\ y_1, \ldots, y_n &\geq 0 \quad \text{and} \quad w \in \mathbb{R}. \end{aligned}$$

Online Learning

So far: we assumed we could see the future (e.g., scheduling, caching).

What if we can't see the future? This is called an *online* setting, not like the internet, but as if the input is waiting *on line*.

An Online Problem

1. The input arrives "one piece at a time."
2. An algorithm makes an irrevocable decision each time it receives a new piece of the input.

Online Decision Making

Choose from expert predictions every time step. An adversary decides who predicts well/poorly in response to your strategy.

Online Decision-Making

At each time step $t = 1, 2, \ldots, T$:

a decision-maker picks a probability distribution $\mathbf{p}^t$ over her experts or actions $i = 1, \ldots, N$

an adversary picks a **loss** vector $\ell^t : A \to [-1, 1]$

an action i^t is chosen according to the distribution $\mathbf{p}^t$, and the decision-maker receives loss ℓ_i^t

the decision-maker learns ℓ^t, the entire **loss** vector

The input arrives "one piece at a time."

What should we compare to?