CS861: Algorithmic Game Theory & Learning

Chapter 2: Two Player Zero Sum Games

Kirthevasan Kandasamy

UW-Madison

Outline

1. Introduction and definitions
2. Minimax theorem
3. Case study: Zero sum games in Generative AI

Slides are intended as teaching aids only and do not include all material discussed in class. Students are strongly encouraged to attend lectures and take their own notes.

Ch 2.1: Basic definitions

Example 1. Consider the following example. How is this game different to games we have seen previously?

P2 P1	**C**	**D**
A	$(0, 0)$	$(2, -2)$
B	$(5, -5)$	$(1, -1)$

Let us first compute Nash equilibria in this game. There are no pure NE. So let us assume that $s_1 = (x, 1 - x)$ and $s_2 = (y, 1 - y)$, where $x, y \in (0, 1)$ is a NE.

Applying the indifference principle for P2, *i.e.*, $u_2(x, C) = u_2(x, D)$, we get $x \cdot 0 + (1 - x) \cdot (-5) = x \cdot (-2) + (1 - x) \cdot (-1)$. Hence, $x = 2/3$.

Applying the indifference principle for P1,*i.e.*, $u_1(A, y) = u_1(B, y)$, we get $y \cdot 0 + (1 - y) \cdot 2 = y \cdot 5 + (1 - y) \cdot 1$. Hence, $y = 1/6$.

Hence, $s_1 = (2/3, 1/3)$ and $s_2 = (1/6, 5/6)$, is a NE.

Example 1 (cont'd)

P1 \ P2	**C**	**D**
A	$(0,0)$	$(2,-2)$
B	$(5,-5)$	$(1,-1)$

Recall, safe strategies. Let $g_i(s_i)$ denote the lowest possible utility agent i could achieve over the strategies of the others, *i.e.*, $g_i(s_i) = \min_{s_{-i} \in \mathcal{S}_{-i}} u_i(s_i, s_{-i})$. A strategy $\widetilde{s}_i$ is a safe strategy for agent i if $g_i(\widetilde{s}_i) \geq g_i(s_i)$ for all $s_i \in \mathcal{S}_i$.

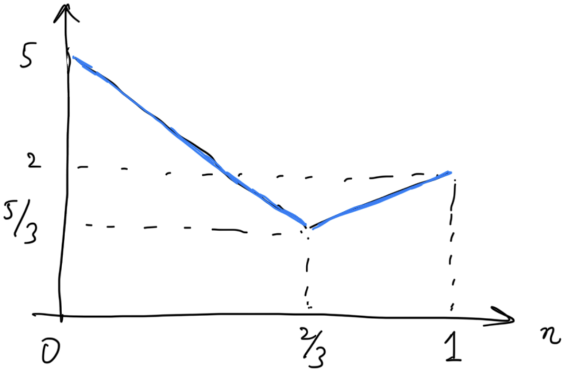
Let us now compute the safe strategy for P1. Under a strategy profile $s_1 = (x, 1-x)$ and $s_2 = (y, 1-y)$, we have,

$$\begin{aligned} u_1(s_1, s_2) &= xy \cdot 0 + x(1-y) \cdot 2 + (1-x)y \cdot 5 + (1-x)(1-y) \cdot 1 \\ &= 1 + x + 4y - 6xy \end{aligned}$$

Therefore,

$$\begin{aligned} g(s_1) = \min_{s_2} u_1(s_1, s_2) &= \min_{y} x + 1 + 6y\left(\frac{2}{3} - x\right) \\ &= \begin{cases} x + 1 & \text{if } \frac{2}{3} - x \geq 0, \\ 5 - 5x & \text{if } \frac{2}{3} - x < 0. \end{cases} \end{aligned}$$

Example 1 (cont'd)



Hence,

$$\underset{x}{\operatorname{argmax}}\, \min_{y} u_1(s_1, s_2) = \frac{2}{3}$$

Example 1 (cont'd)

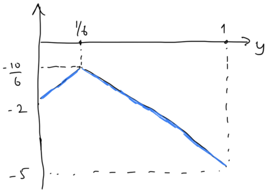
P2 P1	**C**	**D**
A	$(0,0)$	$(2,-2)$
B	$(5,-5)$	$(1,-1)$

Let us now compute the safe strategy for P2. As $u_1 = -u_2$, under a strategy profile $s_1 = (x, 1-x)$ and $s_2 = (y, 1-y)$, we have, $u_2(s_1, s_2) = -1 - x - 4y + 6xy$. Therefore,

$$\min_{s_1} u_1(s_1, s_2) = \min_x -1 - 4y - x\,(6y-1)) = \begin{cases} -1-4y & \text{if } 6y-1 \geq 0, \\ -2+2y & \text{if } 6y-1 < 0. \end{cases}$$

Hence,

$$\operatorname*{argmax}_y \min_x u_2(s_1, s_2) = \frac{1}{6}.$$



Two-player zero sum games

Definition (Two-player zero sum game). A two-player zero sum game is a normal form game with two players and where the sum of utilities is equal for all action profiles, *i.e.*,

$$u_1(a_1, a_2) + u_2(a_1, a_2) = u_1(a_1', a_2') + u_2(a_1', a_2') \quad \text{for all } a_1, a_1' \in \mathcal{A}_1,\ a_2, a_2' \in \mathcal{A}_2.$$

Without loss of generality, we will assume that the sum is 0.

Following convention, we will assume $\mathcal{A}_1 = [m] = \{1, \dots, m\}$ and $\mathcal{A}_2 = [n] = \{1, \dots, n\}$. Then, the utilities can be represented by a payoff matrix $Q \in \mathbb{R}^{m \times n}$ where, $u_1(i,j) = -u_2(i,j) = Q_{i,j}$.

We will also denote the mixed strategies of player 1 and player 2 by x and y respectively (instead of s_1, s_2), where

$$x \in \Delta_m \triangleq \Delta([m]) = \left\{z \in \mathbb{R}^m; z \geq 0; z^\top \mathbf{1}_m = 1\right\},$$

$$y \in \Delta_n \triangleq \Delta([n]) = \left\{z \in \mathbb{R}^n; z \geq 0; z^\top \mathbf{1}_n = 1\right\}.$$

Two-player zero sum games: Nash strategies

We can therefore write the utilities as,

$$u_1(x, y) = \mathbb{E}_{i\sim x, j\sim y}\left[u_1(i, j)\right] = \sum_{i,j} x_i y_j Q_{i,j} = x^\top Q y.$$

Similarly, $u_2(x, y) = -x^\top Q y$.

Recall, Nash equilibrium. A strategy profile $s^\star = (s_1^\star, \dots, s_n^\star)$ is a *Nash equilibrium* if $s_i^\star$ is a best response to $s_{-i}^\star$ for each agent i, *i.e.*, $u_i(s_i^\star, s_{-i}^\star) \geq u_i(s_i', s_{-i}^\star) \qquad \text{for all } s_i' \in \mathcal{S}_i,\ i \in [n]$.

Nash equilibrium in a zero sum game. If $x^\star, y^\star$ is a Nash equilibrium, it satisfies

$$\text{for all } x \in \Delta_m,\ y \in \Delta_n, \qquad x^\top Q y^\star \leq {x^\star}^\top Q y^\star \leq {x^\star}^\top Q y$$

Two-player zero sum games: Safe strategies

Recall, safe strategies. Let $g_i(s_i)$ denote the lowest possible utility agent i could achieve over the strategies of the others, *i.e.*, $g_i(s_i) = \min_{s_{-i} \in \mathcal{S}_{-i}} u_i(s_i, s_{-i})$. A strategy $\widetilde{s}_i$ is a safe strategy for agent i if $g_i(\widetilde{s}_i) \geq g_i(s_i)$ for all $s_i \in \mathcal{S}_i$.

Recall, utilities in a zero-sum game. Under a strategy profile $x \in \Delta_m$, $y \in \Delta_n$, we have $u_1(x, y) = \sum_{i,j} x_i y_j Q_{i,j} = x^\top Q y$, and $u_2(x, y) = -x^\top Q y$.

Safe strategies. The safe strategies $\widetilde{x}, \widetilde{y}$ satisfy,

$$\widetilde{x} = \operatorname*{argmax}_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y,$$

$$\widetilde{y} = \operatorname*{argmax}_{y \in \Delta_n} \min_{x \in \Delta_m} -x^\top Q y = \operatorname*{argmin}_{y \in \Delta_n} \max_{x \in \Delta_m} x^\top Q y.$$

Shortly, we will see that safe strategies and Nash strategies coincide in two-player zero sum games.

Ch 2.2: The minimax theorem

Motivation. Consider a two-player zero sum game with payoff matrix Q. Suppose P1 had to *announce* their strategy first, and commit to it. Which strategy would they choose?

We know, if P1 announces x, P2 will choose

$$\overline{y}(x) = \operatorname*{argmax}_{y\in\Delta_n} \underbrace{u_2(x,y)}_{=-x^\top Qy} = \operatorname*{argmin}_{y\in\Delta_n} x^\top Qy.$$

Hence, if P1 had to announce their strategy first, they will choose:

$$\overline{x} = \operatorname*{argmax}_{x\in\Delta_n} x^\top Q\overline{y}(x) = \operatorname*{argmax}_{x\in\Delta_m} \min_{y\in\Delta_n} x^\top Qy = \textit{P1's safe strategy } \widetilde{x}$$

A similar argument shows that if P2 had to choose their strategy first, they will choose their safety strategy $\widetilde{y} = \operatorname{argmin}_{y\in\Delta_n} \operatorname{argmax}_{x\in\Delta_m} x^\top Qy$.

The minimax theorem: motivation (cont'd)

Question: If you are P1, would you prefer to announce first, or would you prefer that P2 announce first?

Ans: It is quite clear that announcing your strategy first does not help. Recall, in HW0, you showed

$$\max_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y \leq \min_{x \in \Delta_m} \max_{y \in \Delta_n} x^\top Q y$$

Hence, it appears that we should announce second.

Surprisingly, it turns out that announcing firstor second does not matter!
The celebrated minimax theorem tells us that we can replace the above inequality with an equality.

The minimax theorem

Theorem (Von Neumann's minimax theorem). In any two player zero sum game, there exists $V \in \mathbb{R}$ such that,

$$V \overset{\Delta}{=} \max_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y = \min_{x \in \Delta_m} \max_{y \in \Delta_n} x^\top Q y$$

We call V the value of the game.

Proof. The classical proof is based on strong duality (see Theorem 2.3.1 in KP). In Chapter 5, we will do a more modern (and simpler) proof using online learning.

The minimax theorem (cont'd)

Corollary. If $\widetilde{x}, \widetilde{y}$ are safe strategies, then $V = \widetilde{x}^\top Q \widetilde{y}$.

Recall, Safe strategies. The safe strategies $\widetilde{x}, \widetilde{y}$ satisfy, $\widetilde{x} = \mathrm{argmax}_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y$, and $\widetilde{y} = \mathrm{argmin}_{y \in \Delta_n} \max_{x \in \Delta_m} x^\top Q y$.

Proof. First, using the minimax theorem, we have

$$V = \max_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y = \min_{y \in \Delta_n} \widetilde{x}^\top Q y = \max_{y \in \Delta_n} x^\top Q \widetilde{y}.$$

The second equality uses the definition of $\widetilde{x} = \mathrm{argmax}_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y$, while the third equality uses the definition of $\widetilde{y}$.

Hence, for all y, $V \leq \widetilde{x}^\top Q y$. In particular, $V \leq \widetilde{x}^\top Q \widetilde{y}$.

Similarly, for all x, $V \geq x^\top Q \widetilde{y}$. In particular, $V \geq \widetilde{x}^\top Q \widetilde{y}$. □

Safe strategies and Nash equilibria in two-player zero sum games

Theorem (Safe strategies are Nash strategies in TPZSGs (Prop 2.5.3 in KP)). In a two player zero sum game, a pair of strategies $\overline{x} \in \Delta_m, \overline{y} \in \Delta_n$ are safe strategies for P1 and P2 respectively iff $(\overline{x}, \overline{y})$ is a Nash equilibrium.

Proof. 1. Safe $\implies$ NE. Let us first recall the minimax theorem and its corollary.

$$\text{value of the game } V = \widetilde{x}^\top Q \widetilde{y} = \max_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Q y = \min_{x \in \Delta_m} \max_{y \in \Delta_n} x^\top Q y \qquad (1)$$

Suppose $\overline{x}, \overline{y}$ are safe for P1 and P2 respectively. As $\overline{x}$ is safe for P1, she is guaranteed a payoff of at least $\max_x \min_y x^\top Q y$ regardless of what y plays, *i.e.*,

$$\text{for all } y \in \Delta_n, \quad \overline{x}^\top Q y \geq \max_x \min_y x^\top Q y = \overline{x}^\top Q \overline{y}.$$

Here, the last equality uses (1). Hence, $\overline{y}$ is a best response to $\overline{x}$ for P2. We can similarly show that $\overline{x}$ is a best response to $\overline{y}$. Hence, $(\overline{x}, \overline{y})$ is a Nash equilibrium. □

Proof of theorem (cont'd)

2. NE $\implies$ Safe. Now suppose that $(\overline{x}, \overline{y})$ is a Nash equilibrium. Then,

$$\min_y \overline{x}^\top Q y \overset{(a)}{=} \underbrace{\overline{x}^\top Q \overline{y}}_{=V' \text{ (say)}} \overset{(b)}{=} \max_x x^\top Q \overline{y}.$$

Here, (a) uses the fact that $\overline{y}$ is a best response to $\overline{x}$ and (b) uses the fact that $\overline{x}$ is a best response to $\overline{y}$.

Now suppose that $\overline{x}$ is not safe for P1. Then, there exists x' such that, $\min_y x'^\top Q y > V'$. In particular, $x'^\top Q \overline{y} > V'$. However, this contradicts (b) in the equation above.

We can similarly show that $\overline{y}$ is safe for P2. □

Why study two-player zero sum games?

- A very natural abstraction as it models pure competition: players' interests are perfectly opposed to each other.
- Interesting from an academic perspective, as they have many nice properties:
  - Minimax theorem, well-defined "value" for a game
  - Equivalence of safe strategies and NE.
- Many real-world applications can be framed as TPZSGs. Some examples:
  - Security: attacker vs defender
  - ML: generative models (next topic)
- Easy to compute NE using linear programs (coming up in chapter 3) and online learning (in chapter 5).
- Interesting connections to convex optimization.

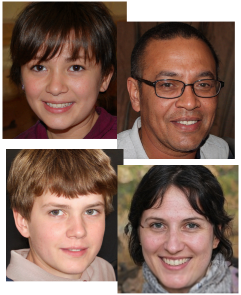
Quiz: state if the following statements are true or false

1. In a two player *general* sum game, it is always better to announce your strategy second.
2. In an n player zero sum game, *i.e.*, where $\sum_{i=1}^{n} u_i(a) = 0$ for all $a \in \mathcal{A}_1 \times \ldots \mathcal{A}_n$, safe strategies constitute a Nash equilibrium.

Ch 2.3: Case study: Zero sum games in Generative AI

Given data from a “true” distribution, we want to be able to generate data that “looks like” the true distribution.

$x \sim p_{\text{true}}$

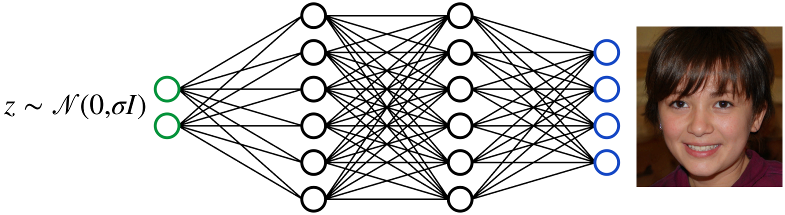


$x \sim p_{\text{learned}}$

We want $p_{\text{true}} \approx p_{\text{learn}}$.

Case study: Zero sum games in Gen AI (cont'd)

How does one generate data? One approach is to generate data from a "simple" distribution P, and pass it through a neural network, to produce "complex" data.



Generator network g_θ with parameters θ

Suppose that the distribution of the data generated by this network is p_{learn}. We wish to learn the parameters θ of the generator network f_θ so that $p_{\text{learn}} \approx p_{\text{true}}$.

Generative Adversarial Networks

Let us try to formalize this (although we will still be somewhat informal).

Suppose the true data is drawn from some distribution p_{true}, and belongs to some space $\mathcal{X}$ (*e.g.*, space of images, text, or just Euclidean data).

We can generate data from a "simple" distribution P (*e.g.*, $P = \mathcal{N}(0, \sigma I_n)$), where the samples belong to some space $\mathcal{Z}$ (*e.g.*, $\mathcal{Z} = \mathbb{R}^n$).

We have a generator network $g_\theta : \mathcal{Z} \to \mathcal{X}$ which maps data in $\mathcal{Z}$ to data in $\mathcal{X}$. Different values for θ produce different mappings.

Let us pretend we know p_{true} for now (in practice, you only have samples from p_{true}). We wish to find θ so that the distribution of $g_\theta(Z)$ where $Z \sim P$, is similar to p_{true}. If we can define a distance ρ between the distributions, then we can choose θ to minimize it, *i.e.*,

$$\theta = \operatorname*{arginf}_{\theta} \rho(p_{\mathrm{true}}, g_\theta(P)).$$

Generative Adversarial Networks (cont'd)

Defining a distance. Consider some function $d : \mathcal{X} \to \mathbb{R}$. Then, $|\mathbb{E}_{X \sim p_{\text{true}}}[d(X)]$ $- \mathbb{E}_{Z \sim P}[d(g_\theta(Z))]|$ is *a* measure of dissimilarity between p_{true} and $g_\theta(P)$.

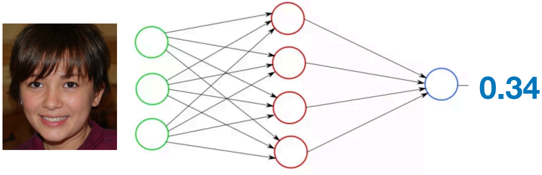
But comparing two distributions using a single function is too weak. We can instead consider a class of functions $\mathcal{D}$ containing many functions which map from $\mathcal{X} \to \mathbb{R}$ and consider the largest possible difference[1]:

$$\begin{aligned} \rho(p_{\text{true}}, g_\theta(P)) &= \sup_{d \in \mathcal{D}} \left| \mathbb{E}_{X \sim p_{\text{true}}}[d(X)] - \mathbb{E}_{Z \sim P}[d(g_\theta(Z))] \right| \\ &= \sup_{d \in \mathcal{D}} \left(\mathbb{E}_{X \sim p_{\text{true}}}[d(X)] - \mathbb{E}_{Z \sim P}[d(g_\theta(Z))] \right) \qquad \texttt{if } \mathcal{D} \texttt{ is symmetric} \end{aligned}$$

[1] You can show that ρ is a distance (*i.e.*, satisfies all 4 axioms). Turns out, when $\mathcal{D}$ is the class of 1-Lipschitz functions, it is the Wasserstein-1 (or earth mover's) distance. This specific formalism for GANs is called Wasserstein-GANs for this reason.

Generative Adversarial Networks (cont'd)

It is usually difficult to work with an abstract function class, so we can consider a function class parametrized by the weights of a different neural network (called the discriminator network), *i.e.*, $\mathcal{D} = \{d_\phi\}_\phi$.

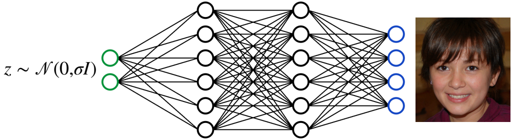


Discriminator network d_ϕ with parameters ϕ

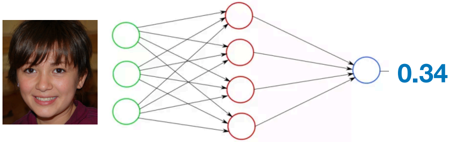
Hence, our metric is,

$$\rho(p_{\mathrm{true}}, g_\theta(P)) = \sup_\phi \left(\mathbb{E}_{X \sim p_{\mathrm{true}}}[d_\phi(X)] - \mathbb{E}_{Z \sim P}[d_\phi(g_\theta(Z))]\right)$$

Generative Adversarial Networks (cont'd)



Generator network g_θ with parameters θ



Discriminator network d_ϕ with parameters ϕ

Let us now put everything together. We initially set our goal as finding parameters θ of a generator network so that, $\theta = \mathrm{argmin}_\theta\, \rho(p_{\mathrm{true}}, g_\theta(P))$.

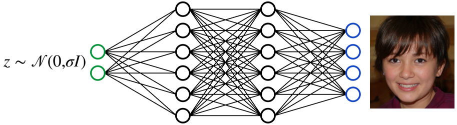
We then defined the following distance,

$$\rho(p_{\mathrm{true}}, g_\theta(P)) = \sup_\phi \left(\mathbb{E}_{X\sim p_{\mathrm{true}}}[d_\phi(X)] - \mathbb{E}_{Z\sim P}[d_\phi(g_\theta(Z))]\right)$$

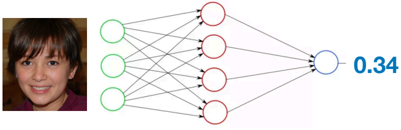
Therefore, we can frame our problem as,

$$\inf_\theta \sup_\phi \underbrace{\left(\mathbb{E}_{X\sim p_{\mathrm{true}}}[d_\phi(X)] - \mathbb{E}_{Z\sim P}[d_\phi(g_\theta(Z))]\right)}_{\triangleq (f(\phi,\theta))}$$

Generative Adversarial Networks (cont'd)



Generator network g_θ with parameters θ



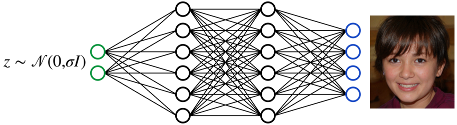
Discriminator network d_ϕ with parameters ϕ

$$\inf_\theta \sup_\phi f(\phi,\theta), \qquad \text{where } f(\phi,\theta) = \mathbb{E}_{X\sim p_{\mathrm{true}}}[d_\phi(X)] - \mathbb{E}_{Z\sim P}[d_\phi(g_\theta(Z))]$$

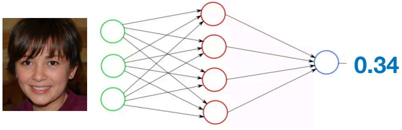
This can be viewed as a zero sum game between two players. P1 wishes to maximize $f(\phi,\theta)$ and her action set is the weights ϕ for the discriminator network. P2 wishes to minimize $f(\phi,\theta)$ and her action set is the weights θ for the generator network.

Intuition. The discriminator network is trying to differentiate between true images and the images generated by the generator, by assigning high scores to true images and low scores to fake images. The generator is trying to deceive the discriminator by making it very hard to distinguish between the two.

Generative Adversarial Networks (cont'd)



Generator network g_θ with parameters θ



Discriminator network d_ϕ with parameters ϕ

$$\inf_{\theta} \sup_{\phi} f(\phi, \theta), \qquad \text{where } f(\phi, \theta) = \mathbb{E}_{X \sim p_{\mathrm{true}}}[d_\phi(X)] - \mathbb{E}_{Z \sim P}[d_\phi(g_\theta(Z))]$$

Some remarks, which are beyond the scope of this course:

- In reality, we will not know p_{true}. So replace the expectations with empirical expectations over true and generated data respectively, *i.e.*,

$$\widehat{f}(\phi, \theta) = \frac{1}{n} \sum_{i=1}^{n} d_\phi(X_i) - \frac{1}{m} \sum_{i=1}^{m} d_\phi(g_\theta(Z_i)).$$

- To find the minimax solution, we can run gradient descent and ascent on $\widehat{f}$ with respect to ϕ and θ respectively.