Matrix games

Axel Böhm

December 13, 2021

1 Introduction

2 Algorithms

3 More min-max problems

Introduction

Given

- Player I (rows, Alice)
- Player II (columns, Bob)
- a *payoff* matrix $A \in \mathbb{R}^{m \times n}$

Every round

(i) Alice picks (row) strategy $i \in [m] := \{1, \ldots, m\}$ Bob picks (col) strategy $j \in [n]$

(ii) Bob pays Alice the amount $a_{i,j}$

zero-sum game

Example: penalty game

		kicker L	kicker R
goalkeeper	L	1	-1
	R	-1	1

Figure: penalty game

Example: prisoners dilemma

	Confess A	Stay quiet A
Confess B	6 6	10 0
Stay quiet B	0 10	2 2

Figure: prisoners dilemma

Worst case

- if Alice chooses strategy i she gets (at least): $\min_{j\in[n]} a_{i,j}$
- Alice can ensure payoff $\max_{i\in[m]} \min_{j\in[n]} a_{i,j}$
- Bob pays (at most) $\min_{j\in[n]} \max_{i\in[m]} a_{i,j}$

We claim:

$$\max_i \min_j a_{i,j} \leq \min_j \max_i a_{i,j}$$

"Tallest dwarf is not as tall as the smallest giant."

But: **No equality in general!**

Worst case

- if Alice chooses strategy i she gets (at least): $\min_{j\in[n]} a_{i,j}$
- Alice can ensure payoff $\max_{i\in[m]} \min_{j\in[n]} a_{i,j}$
- Bob pays (at most) $\min_{j\in[n]} \max_{i\in[m]} a_{i,j}$

We claim:

$$\max_i \min_j a_{i,j} \leq \min_j \max_i a_{i,j}$$

"Tallest dwarf is not as tall as the smallest giant."

But: **No equality in general!**

Proof of the min-max theorem

$$
\begin{aligned}
a_{ij} &\leq a_{ij} && \forall i, j \\
a_{ij} &\leq \max_i a_{ij} && \forall i, j \\
\min_j a_{ij} &\leq \min_j \max_i a_{ij} && \forall i
\end{aligned}
$$

Definition

We call (i^*, j^*) a saddle point (or *Nash equilibrium*) if

$$\max_i a_{ij^*} = a_{i^*j^*} = \min_j a_{i^*j}.$$

These are called *pure strategies*.

Proof of the min-max theorem

$$
\begin{aligned}
a_{ij} &\leq a_{ij} && \forall i, j \\
a_{ij} &\leq \max_i a_{ij} && \forall i, j \\
\min_j a_{ij} &\leq \min_j \max_i a_{ij} && \forall i
\end{aligned}
$$

Definition

We call (i^*, j^*) a saddle point (or *Nash equilibrium*) if

$$\max_i a_{ij^*} = a_{i^*j^*} = \min_j a_{i^*j}.$$

These are called *pure strategies*.

Rock paper scissors

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

Mixed Strategies

With pure strategies we do not always have a saddle point.

von Neumann (1928) — Mixed strategies

- Alice picks strategies $1, \ldots, m$ *with probabilities* $x \in \Delta_m$
- Bob picks strategies $1, \ldots, n$ *with probabilities* $x \in \Delta_n$

Expected gain of Alice is

$$\langle x, Ay \rangle = \sum_{i,j} a_{ij} x_i y_j$$

Theorem (Saddle point exists)

Expected gain of Alice = expected loss of Bob

$$\max_{x \in \Delta} \min_{y \in \Delta} \langle x, Ay \rangle = \min_{y \in \Delta} \max_{x \in \Delta} \langle x, Ay \rangle.$$

Mixed Strategies

With pure strategies we do not always have a saddle point.

von Neumann (1928) — Mixed strategies

- Alice picks strategies $1, \ldots, m$ *with probabilities* $x \in \Delta_m$
- Bob picks strategies $1, \ldots, n$ *with probabilities* $x \in \Delta_n$

Expected gain of Alice is

$$\langle x, Ay \rangle = \sum_{i,j} a_{ij} x_i y_j$$

Theorem (Saddle point exists)

Expected gain of Alice = expected loss of Bob

$$\max_{x \in \Delta} \min_{y \in \Delta} \langle x, Ay \rangle = \min_{y \in \Delta} \max_{x \in \Delta} \langle x, Ay \rangle.$$

Stopping criteria

$$\min_{x\in\Delta} \underbrace{\max_{y\in\Delta}\langle x, Ay\rangle}_{f_p(x)} =: \boldsymbol{v} = \max_{y\in\Delta} \underbrace{\min_{x\in\Delta}\langle x, Ay\rangle}_{f_d(y)}$$

stopping criterion

$$f_p(x) - f_p(x^*) = f_p(x) - v \leq \epsilon/2$$
$$f_d(y^*) - f_d(y) = v - f_d(y) \leq \epsilon/2$$
$$\Rightarrow f_p(x) - f_d(y) \leq \epsilon$$

Before we never had the optimal value!

Solution will always be on the boundary:

$$f_p(x) = \max_{y\in\Delta}\langle A^T x, y\rangle = \max_j \langle A^T x, e_j\rangle$$

Stopping criteria

$$\min_{x\in\Delta} \underbrace{\max_{y\in\Delta}\langle x, Ay\rangle}_{f_p(x)} =: \boldsymbol{v} = \max_{y\in\Delta} \underbrace{\min_{x\in\Delta}\langle x, Ay\rangle}_{f_d(y)}$$

stopping criterion

$$f_p(x) - f_p(x^*) = f_p(x) - v \leq \epsilon/2$$
$$f_d(y^*) - f_d(y) = v - f_d(y) \leq \epsilon/2$$
$$\Rightarrow f_p(x) - f_d(y) \leq \epsilon$$

Before we never had the optimal value!

Solution will always be on the boundary:

$$f_p(x) = \max_{y\in\Delta}\langle A^T x, y\rangle = \max_j \langle A^T x, e_j\rangle$$

Consider

$$\min_{x\in\Delta}\max_{y\in\Delta}\langle x, Ay\rangle$$

as a minimization problem

$$\min_{x\in\Delta} f_p(x) = \langle x, Ay^*\rangle.$$

Then, by the **first-order optimality condition**

$$x^* \in \arg\min_{x\in\Delta} f_p(x) \Leftrightarrow \langle \nabla f_p(x^*), x - x^*\rangle \geq 0 \quad \forall x \in \Delta$$

Thus

$$\langle A^T y^*, x - x^*\rangle \geq 0 \quad \forall x \in \Delta$$
$$\langle -Ax^*, y - y^*\rangle \geq 0 \quad \forall y \in \Delta$$

Concatenate the two conditions to get

$$\left\langle \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^* \\ y^* \end{pmatrix} \right\rangle \geq 0.$$

Games as Variational Inequalities

We had:

$$\left\langle \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^* \\ y^* \end{pmatrix} \right\rangle \geq 0.$$

By rewriting $z = (x, y)$ and $F(z) = [A^T y; Ax]$, then

$$\langle F(z^*), z - z^* \rangle \geq 0 \quad \forall z \in \Delta_n \times \Delta_m =: C \tag{VI}$$

Variational inequality

If $F = \nabla\varphi$ then (VI) would be equivalent to

$$\min_{z \in C} \varphi(z)$$

Potential - integrability

Question: Does there exist a potential φ for F, such that $F = \nabla\varphi$

Integrability condition (from calculus)

Is the case if

$$\frac{\partial\varphi}{\partial x\partial y} = \frac{\partial\varphi}{\partial y\partial x}$$

But

$$\frac{\partial F_1}{\partial y} = A^T \neq -A = \frac{\partial F_2}{\partial x}$$

VI as Fixed point equation

$$\langle F(z^*), z - z^* \rangle \geq 0 \quad \forall z \in C$$
$$\Leftrightarrow z^* = P_C(z^* - F(z^*)) \qquad \text{(FP)}$$

Proof.

Applying the property of the projection

$$\langle P_C(x) - x, x' - P_C(x) \rangle \geq 0 \quad \forall x' \in C$$

with (FP), gives

$$\langle z^* - (z^* - F(z^*)), z - z^* \rangle \geq 0 \quad \forall z \in C. \quad \square$$

- should remind us of (projected) gradient descent
- when you see a **fixed point equation: iterate!**

But is it any good?

Consider the **unconstrained case**

$$\langle F(z^*), z - z^* \rangle \geq 0, \forall z \quad \Leftrightarrow \quad F(z^*) = 0.$$

$$z_{k+1} = z_k - \alpha F(z_k)$$

Then

$$\|z_{k+1}\|^2 = \|z_k\|^2 - \underbrace{2\alpha\langle F(z_k), z_k\rangle}_{=0} + \alpha^2\|F(z_k)\|^2$$

$$= \|z_k\|^2 + \alpha^2\|F(z_k)\|^2$$

Resulting in $\|z_{k+1}\| \geq \|z_k\|$.

$\Rightarrow$ **No bueno!**

We can still show

Theorem

Convergence rate for ***averaged iterates*** *in terms of* ***primal-dual gap***

$$f_p(\bar{x}_k) - f_d(\bar{y}_k) \leq \frac{C}{\sqrt{k}}.$$

- with the same analysis as for subgradient descent

Sketch of the proof

With the notation $g_k = F(z_k)$ we get

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &\leq \|x_k - \alpha_k g_k - x^*\|^2 \\ &= \|x_k - x^*\|^2 + 2\alpha_k \langle g_k, x^* - x_k \rangle + \alpha^2 \|g_k\|^2.\end{aligned}$$

But now $\langle g_k, x^* - x_k \rangle = [f_d(y_k) - f_p(x_k)]$. Rest of the proof is left as an exercise.

A better method

Algorithm Extragradient Method [1976]

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad w_{k+1} = z_k - \alpha_k F(z_k)$
3: $\quad z_{k+1} = z_k - \alpha_k F(w_k)$

- more conservative
- works for general min-max problems

$$\min_x \max_y \Phi(x, y) \quad \Rightarrow F(x, y) = (\nabla_x \Phi(x, y), -\nabla_y \Phi(x, y))^T$$

- can even improve behavior of pure minimization ($F = \nabla f$)
- improved rate of $\mathcal{O}(1/k)$ for averaged iterates

Robust Optimization

- uncertainty in the **objective** ($u \in U$ uncertainty set)

$$\min_x f(x, u)$$

- bound the **worst case**

$$\min_x \max_{u \in U} f(x, u)$$

Vulnerability of Neural Networks

Explaining and Harnessing Adversarial Examples

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy
Google Inc., Mountain View, CA
{goodfellow,shlens,szegedy}@google.com

Abstract

Several machine learning models, including neural networks, consistently misclassify *adversarial examples*—inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed in-

Vulnerability in "real life"

(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Adversarial examples can be printed out on normal paper and photographed with a standard resolution smartphone and still cause a classifier to, in this case, label a "washer" as a "safe".

Generation of adversarial examples

Training objective

$$\min_{\theta} J(\theta, x, y)$$

- θ parameters (weights, biases)
- x images
- y corresponding labels

Attack direction ("white box")

$$\nabla_x J(\theta, x, y)$$

Typically only the sign is used.

Adversarial training

Augment training data with adversarial examples

- grab a fresh training example x
- perturb image: $\tilde{x} \leftarrow x - \epsilon \text{sign}(\nabla_x J(\theta, x, y))$
- update weights: $\theta \leftarrow \theta - \nabla_\theta J(\theta, \tilde{x}, y)$

Corresponds to solving the robust min-max problem

$$\min_\theta \max_{\tilde{x}} \; J(\theta, \tilde{x}, y)$$
$$\text{s.t. } \|\tilde{x} - x\|_\infty \leq \epsilon$$

GANs

Generative Adversarial Networks is the most interesting idea in the last 10 years in Machine Learning.

— Yann LeCun, Director of AI Research at Facebook AI

What do they *generate*?

Dank Learning: Generating Memes Using Deep Neural Networks

Abel L. Peirson V
Department of Physics
Stanford University
alpv95@stanford.edu

E. Meltem Tolunay
Department of Electrical Engineering
Stanford University
meltem.tolunay@stanford.edu

Abstract

We introduce a novel meme generation system, which given any image can produce a humorous and relevant caption. Furthermore, the system can be conditioned on not only an image but also a user-defined label relating to the meme template, giving a handle to the user on meme content. The system uses a pre-trained Inception-v3 network to return an image embedding which is passed to an attention-based deep-layer LSTM model producing the caption - inspired by the widely recognized Show and Tell Model. We implement a modified beam search to encourage diversity in the captions. We evaluate the quality of our model using perplexity and human assessment on both the quality of memes generated and whether they can be differentiated from real ones. Our model produces original memes that cannot on the whole be differentiated from real ones. https://github.com/alpv95/MemeProject

How did they do?

Glove Average

Attention

Seen Images

Unseen Images

More serious applications (inpainting)

High Resolution Face Completion with Multiple Controllable Attributes via Fully End-to-End Progressive Generative Adversarial Networks

ZEYUAN CHEN, North Carolina State University
SHAOLIANG NIE, North Carolina State University
TIANFU WU, North Carolina State University
CHRISTOPHER G. HEALEY, North Carolina State University

.07632v1 [cs.CV] 23 Jan 2018

Fig. 1. Face completion results of our method on CelebA-HQ [Karras et al. 2017]. Images in the left most column of each group are masked with gray color, while the rest are synthesized faces. *Top:* our approach can complete face images at high resolution (1024 × 1024). *Bottom:* the attributes of completed faces can be controlled by conditional vectors. Attributes ["Male", "Smiling"] are used in this example. The conditional vectors of column two to five are [0, 0], [1, 0]

Or less serious...

Creating Art

TECH \ ARTIFICIAL INTELLIGENCE \ CULTURE \

Christie's sells its first AI portrait for $432,500, beating estimates of $10,000

The image was created using a machine learning algorithm that scanned historical artwork

By James Vincent | Oct 25, 2018, 1:03pm EDT

Why Adversarial?

Two NNs trained at the same time.

GANs mathematically

Given by the *two-player game*.

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{x \sim p_{data}}[\log(\mathcal{D}(x))] + \mathbb{E}_{z \sim p_{noise}}[\log(1 - \mathcal{D}(\mathcal{G}(z)))]$$

where

- $p_z(z)$ is the input noise variable,
- $\mathcal{G}$ is the *generator*,
- $\mathcal{D}$ is the *discriminator*.

Discriminator gives probability that x came from the data rather than from the generator.