Modern Algorithmic Game Theory

Martin Schmid
Department of Applied Mathematics
Charles University in Prague

October 1, 2025

Understand These!

(a) AlphaZero

(b) AlphaStar

(c) DeepStack

Organizational Info

- Web: `https://sites.google.com/view/agtg-101`
- Discord: You'll get an invite link after the first lecture
- GitHub: `https://github.com/lifrordi/algorithmic_game_theory`
- Lecturer: Martin Schmid
- Teaching assistants: Rado Haluška, David Sychrovský

About the Course

Class

- Simultaneous and sequential decision making
- Solution concepts and optimal policies
- Practical tabular algorithms for finding optimal policies in relatively small games

Exercises

- No regular exercise classes; let us know if you want consultations
- Homework almost every week in which you get to implement games and algorithms
- One strict deadline for the first half of the homework in the fifth or sixth week

Exam

- Demonstration and discussion of your homework solutions

If you have any questions or feedback during the semester, we ask you to use our Discord server; you'll receive an invite after this lecture.

Course Schedule

The preliminary schedule with the list of topics is as follows:

- Introduction, normal-form games and solution concepts
- Minimax theorem and the fictitious play learning dynamics
- Linear programming approaches and correlated equilibria
- The regret minimization framework and regret matching
- Invited talk, **deadline for the first four homework assignments**
- Sequential decision making, extensive-form games and best responses
- Strategy averaging and extensive-form fictitious play
- Sequence-form linear programming and subgame-perfect equilibria
- Counterfactual regret minimization (CFR) and CFR+
- CFR-BR, Discounted CFR and Monte Carlo CFR
- History of AI in games

Other Game Theory Classes

There are two other game theory classes being taught by people from our group.

Algorithmic Game Theory (NDMI098)

- Taught by Martin Balko, David Sychrovský and Tomáš Čížek
- Runs simultaneously with this class in the winter term
- Takes a more principled and theory-oriented approach (very important!)
- Covers many of the same topics as this class, plus some more

Advanced Modern Algorithmic Game Theory (NOPT022) (awfully long name)

- A brand-new course in the summer term taught by Martin Schmid
- A continuation of this class that focuses on approximation methods and large games

Reinforcement Learning vs. Game Theory

Some of you may be familiar with reinforcement learning and its terminology.

Reinforcement Learning

- Focuses mostly on single-agent settings
- The goal is to maximize a cumulative reward
- Scalable and practical algorithms (PPO, SAC, etc.)

Game Theory

- Focuses on multi-agent settings
- Analyzes agent interactions and their incentives
- Mostly interested in optimal solution concepts
- Algorithms (historically) tabular and not scalable

Terminology

- Environment vs. game
- Agent vs. player
- Policy vs. strategy
- Reward vs. utility

We will use these terms interchangeably!

Normal-Form Games

Normal-Form Games

Normal-form games are a model of simultaneous decision making, in which each player chooses their strategy (action), and subsequently all players execute their strategies simultaneously. The outcome depends on the actions chosen by all the players, not just each player's own action.

Definition: Normal-Form Game

A normal-form game is a tuple $\langle \mathcal{N}, (\mathcal{A}_i), (u_i) \rangle$, where

- $\mathcal{N}$ is a **finite** set of players, denoted $1, \dots, N$
- $\mathcal{A}_i$ is a non-empty set of actions available to player i
- u_i is a **payoff/utility** function of player i, defined as $u_i \colon \mathcal{A} \to \mathbb{R}$, where $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}_i$.

Normal-Form Games

- When there are only two players, we can describe the game using a matrix/table
- Rows and columns correspond to actions of the two players
- The cell (i, j) contains players' payoffs $u_1(i, j)$ and $u_2(i, j)$
- If $u_1 + u_2 = c$ in all cells, we call it a **constant-sum** game
- Furthermore, if $c = 0$ ($u_1 = -u_2$), we call it a **zero-sum** game

	rock	paper	scissors
rock	(0, 0)	(-1, 1)	(1, -1)
paper	(1, -1)	(0, 0)	(-1, 1)
scissors	(-1, 1)	(1, -1)	(0, 0)

Table: Rock Paper Scissors

	cooperate	defect
cooperate	(-1, -1)	(-3, 0)
defect	(0, -3)	(-2, -2)

Table: Prisoner's Dilemma

Zero-Sum Games

Zero-sum games carry critical implications, which we will revisit multiple times!

	rock	paper	scissors
rock	(0, 0)	(-1, 1)	(1, -1)
paper	(1, -1)	(0, 0)	(-1, 1)
scissors	(-1, 1)	(1, -1)	(0, 0)

Table: Rock Paper Scissors

	rock	paper	scissors
rock	0	-1	1
paper	1	0	-1
scissors	-1	1	0

Table: Payoff matrix for Rock Paper Scissors for the row player

Strategies I

Pure strategy

- Strategy $a_i \in \mathcal{A}_i$ is player i's pure strategy, sometimes also referred to as an *action*
- This strategy is referred to as pure, because there's no probability involved
- You can think of this as a one-hot vector with 1 at the position of the selected action
- For example, a player can always play `scissors` in Rock Paper Scissors

Mixed strategy

- Mixed strategy is a probability measure over player's pure strategies
- The set of player i's mixed strategies is denoted by Π_i
- Given $\pi_i \in \Pi_i$, we denote the probability of choosing action $a_j \in \mathcal{A}_i$ as $\pi_i(a_j)$
- Mixed strategies allow players to choose actions probabilistically

Strategies II

Strategy profile

- A strategy profile is the set of all players' strategies, denoted by $\pi = (\pi_0, \pi_1, \ldots, \pi_n)$
- We use π_{-i} to refer to strategies of all players in π except the strategy π_i of player i

Support

- The support of strategy π_i is the set of all actions with non-zero probability $\{a_j \in \mathcal{A}_j \mid \pi_i(a_j) > 0\}$.

Evaluating Strategies

- When all players play a pure strategy, we can easily compute the utilities; it suffices to find the corresponding entry $u_i(a)$ in the payoff matrix
- When players use mixed strategies, we need to compute the expected value (the expected utility) given the strategies $u_i(\pi) = \sum_{a \in \mathcal{A}} \pi(a) u_i(a)$
- Since players choose their actions simultaneously, the events are independent and consequently $\pi(a) = \prod \pi_i(a_i)$
- Now we can derive the following:

$$u_i(\pi) = \sum_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i, a_{-i}) \pi_i(a_i) \prod_{j \neq i} \pi_j(a_j) = \sum_{a_i \in \mathcal{A}_i} \pi_i(a_i) u_i(a_i, \pi_{-i})$$

Computing Expected Payoffs

Compute expected payoffs of both players for the following games and strategy profiles:

	rock	paper	scissors
rock	(0, 0)	(-1, 1)	(1, -1)
paper	(1, -1)	(0, 0)	(-1, 1)
scissors	(-1, 1)	(1, -1)	(0, 0)

- $\pi_1 = (0.2, 0.2, 0.6)$
  $\pi_2 = (0.2, 0.2, 0.6)$
- $\pi_1 = (0.6, 0.2, 0.2)$
  $\pi_2 = (0.2, 0.2, 0.6)$

	cooperate	defect
cooperate	(-1, -1)	(-3, 0)
defect	(0, -3)	(-2, -2)

- $\pi_1 = (0.4, 0.6)$
  $\pi_2 = (0.4, 0.6)$
- $\pi_1 = (0.6, 0.4)$
  $\pi_2 = (0.4, 0.6)$

Best Response

- Informally, a **best response** is a strategy that maximizes the expected utility of a player against a fixed strategy profile of their opponents
- Formally, a best response is a set-valued function b that, for a given strategy profile π_{-i} of the opponents computes a strategy π_i that maximizes the utility of player i
- Mathematically,
$$b(\pi_{-i}) = \arg\max_{\pi_i \in \Pi_i} u_i(\pi_i, \pi_{-i})$$
- We will use $\mathbb{BR}(\pi_{-i})$ to denote the set of best-response policies against policy π_{-i}
- This is one of the key concepts that you will see in this class!

Best Response

Lemma: Best Response Condition

For any best response strategy $\pi_i \in \mathbb{BR}(\pi_{-i})$, all actions in its support have the same expected utility.

- To give you an intuition for this lemma, assume that one action in the support gave a strictly worse payoff, the player could reduce its probability and shift it to another action and thus strictly improve its utility
- We will use this property when we formulate finding NEs as a linear program

Lemma: Convexity of the Set of Best Responses

The set $\mathbb{BR}(\pi_{-i})$ of best responses is convex, i.e. for any two strategies from $\mathbb{BR}(\pi_{-i})$, their convex combination is also in $\mathbb{BR}(\pi_{-i})$.

Best Response in Two-Player Zero-Sum Games

- For zero-sum games, it holds that an opponent maximizing their reward is equivalent to an opponent minimizing our reward, as $u_i = -u_{-i}$. Mathematically,

$$b(\pi_i) = \arg\max_{\pi_{-i} \in \Pi_{-i}} u_{-i}(\pi_i, \pi_{-i}) = \arg\min_{\pi_{-i} \in \Pi_{-i}} u_i(\pi_i, \pi_{-i})$$

- This implies that the player's value against any best response strategy is always the same; we denote this unique value as $BRV(\pi_i)$. Mathematically,

$$BRV(\pi_i) = \max_{\pi_{-i}} u_{-i}(\pi_i, \pi_{-i}) = -\min_{\pi_{-i}} u_i(\pi_i, \pi_{-i})$$

Dominated Strategies

Some actions can clearly be poor choices, and it makes no sense for a rational player to play such actions. We call these actions **dominated strategies**.

- Strategy π_i^a **strictly dominates** π_i^b iff for any π_{-i} $u_i(\pi_i^a, \pi_{-i}) > u_i(\pi_i^b, \pi_{-i})$
- Strategy π_i^a **weakly dominates** π_i^b iff for any π_{-i} $u_i(\pi_i^a, \pi_{-i}) \geq u_i(\pi_i^b, \pi_{-i})$ and there is at least one strategy profile π_{-i} for which $u_i(\pi_i^a, \pi_{-i}) > u_i(\pi_i^b, \pi_{-i})$
- A strategy is **strictly/weakly** dominated if there's another strategy that strictly/weakly dominates it
- Two strategies π_i^a and π_i^b are **intransitive** iff one neither dominates nor is dominated by the other

Iterated Removal of Dominated Strategies

- An algorithm that iteratively discards strictly dominated strategies and as a result simplifies payoff matrices for future analysis/other algorithms
- The process reflects the assumption that all players are rational and that this rationality is **common knowledge**: each player knows others are rational, knows that others know they are rational, and so on
- Rationality in this case means that no player will play a dominated strategy
- In specific instances, running this algorithm will result in a 1x1 matrix which effectively finds a stable solution to the game

	left	center	right
top	(13, 3)	(1, 4)	(7, 3)
middle	(4, 1)	(3, 3)	(6, 2)
up	(-1, 9)	(2, 8)	(8, -1)

Iterated Removal of Dominated Strategies

	left	center	right
top	(13, 3)	(1, 4)	(7, 3)
middle	(4, 1)	(3, 3)	(6, 2)
up	(-1, 9)	(2, 8)	(8, -1)

	left	center	right
top	(13, 3)	(1, 4)	(7, 3)
middle	(4, 1)	(3, 3)	(6, 2)
up	(-1, 9)	(2, 8)	(8, -1)

Iterated Removal of Dominated Strategies

	left	center	right
top	(13, 3)	(1, 4)	(7, 3)
middle	(4, 1)	(3, 3)	(6, 2)
up	(-1, 9)	(2, 8)	(8, -1)

	left	center	right
top	(13, 3)	(1, 4)	(7, 3)
middle	(4, 1)	(3, 3)	(6, 2)
up	(-1, 9)	(2, 8)	(8, -1)

Dominated Strategies and Best Response

Examples

Can a strictly/weakly dominated strategy be a best response?

No, if it is a strictly dominated strategy as there's another strategy that is strictly better.
If it is a weakly dominated action, it may or may not be a best response.
Consider the following game:

	L	R
U	1	1
D	1	0

Action `D` is weakly dominated by `U`, but if the column player plays action `L`, then both `D` and `U` give the same payoff.

Dominated Strategies and Best Response

Examples

Can a strictly/weakly dominated strategy that we found when running Iterated Removal of Dominated Strategies be a best response in the original game?

No, if it is a strictly dominated strategy, and yes, if it is a weakly dominated strategy. Consequently, running Iterated Removal of Dominated Strategies with weak dominance can lead to discarding best responses and destroying Nash equilibria.

Solution Concepts

Solution Concepts

(a) Maximin

(b) Nash Equilibrium

Maximin

Definition: Maximin Strategy

A Maximin strategy of player i is a strategy that guarantees the highest possible expected utility against the worst-case opponent. We define it as:

$$\arg\max_{\pi_i \in \Pi_i} \min_{\pi_{-i} \in \Pi_{-i}} u_i(\pi_i, \pi_{-i}) = \arg\max_{\pi_i \in \Pi_i} BRV_i(\pi_i)$$

- We assume everyone else is *"out there to get us"*
- A maximin policy maximizes our expected utility assuming the worst-case scenario
- We also use $v_i = \max_{\pi_i} \min_{\pi_{-i}} u_i(\pi_i, \pi_{-i})$ to denote the Maximin value of player i

Nash Equilibrium

Definition: Nash Equilibrium

Strategy profile (π_i, π_{-i}) is a Nash equilibrium if none of the players can benefit from unilaterally deviating from their policy. Mathematically,

$$\forall i \in \mathcal{N}, \forall \pi_i' \in \Pi_i : u_i(\pi_i, \pi_{-i}) \geq u_i(\pi_i', \pi_{-i})$$

- The central solution concept in game theory with extremely diverse applications
- Represents a stable solution where players cannot individually improve their utility
- We can also look at this as each player playing a best response against other players
- It has been proven that every game with a finite number of players, each having a finite number of actions, always has a mixed-strategy Nash equilibrium

Maximin vs. Nash Equilibrium

- Maximin is defined as a strategy, whereas Nash equilibrium as a strategy profile
- We will see interesting differences between the two concepts in the next lecture
- However, we will also see that the two can sometimes represent the same thing

Homework Organization

- Each week (except for one), you will be assigned homework where you are expected to implement an algorithm or a part of it that was presented in the lecture
- You will get a simple template that defines the public API (function names and signatures) that you need to follow so we can easily check your solutions
- The templates are written in Python using NumPy and sometimes Matplotlib
- You can also implement homework tasks in JAX, if you want to learn it ;)
- You will also get a series of tests in the form of a Python script or a Jupyter notebook that we will use during exams to verify that your solution works as expected
- In the case of any questions or bug reports, please reach out to us on Discord

Week 1 Homework

You can find more detailed descriptions of homework tasks in the GitHub repository.

1. Strategy profile evaluation
2. Best response calculation
3. Iterated removal of dominated strategies