Learning in Games Seminar

Introduction to Topics – 29. March 2023

Chair of Decision Sciences & Systems

Agenda

- Seminar format
- Background information
  - Recap: Games and Nash equilibria
  - A brief taxonomy of games
  - What is “learning in games” anyway?
  - A short primer on (multi-agent) reinforcement learning
- Topic assignments
- Q&A

Agenda

- Seminar format
- Background information
  - Recap: Games and Nash equilibria
  - A brief taxonomy of games
  - What is “learning in games” anyway?
  - A short primer on (multi-agent) reinforcement learning
- Topic assignments
- Q&A

Format

Each student will be assigned an individual topic and prepare a presentation as well as a short summary paper

Presentation

45 min presentation
of topic

Handout

should be
shared at
least 3 days
before
presentation

Short handout (extended abstract) of
maximal 2 pages

Quiz

reading
handout
should
suffice for
full score

Mini-quiz for the audience: < 1 min
with MC or single-word responses

- We will have **biweekly meetings** throughout the semester with two topics being presented in each meeting
- Attendance of all meetings is mandatory
  - interaction with the other students' work is expected: E.g., answering prepared mini-quizzes

Agenda

- Seminar format
- Background information
  - Recap: Games and Nash equilibria
  - A brief taxonomy of games
  - What is “learning in games” anyway?
  - A short primer on (multi-agent) reinforcement learning
- Topic assignments
- Q&A

What is a game?

- $G = (N, A, u)$
- Set of players $\mathcal{I} = \{1, \dots, N\}$
- Each player has a set of actions: A_i
- Utility function: $u_i : A_1 \times \cdots \times A_N \to \mathbb{R}$

Battle of the Sexes Game

B \ A	B Fight	B Ballet
A Fight	2 / 3	0 / 0
A Ballet	0 / 0	3 / 2

Outcome $u_i(a_1, \dots, a_N)$ for player i depends on **all** players‘ actions.

Strategic behavior

- How should a player i choose his or her action a_i?
- Goal: maximize expected utility u_i.
  - A player that achieves this given all the information that is available to them is called *rational.*
  - Important: Information about other players, and information about other players‘ information about other players, ... („common knowledge“)
- Optimal play often involves randomization. A *mixed strategy* π is a probability distribution (i.e., a vector) over all available actions.
  - For the resulting space of mixed-strategies, we write Π or ΔA.

Notational conventions

- a_i, π_i, u_i, ... describe the action/strategy/utility/... of a single player i.
- a, π, u describe vectors over all players' actions/strategies/utilities. Such vectors are called action-/strategy-/... *profiles*.
- a_{-i}, π_{-i}, u_{-i} describe the partial profiles for all players, except player i.
- Utility of a strategy profile, means the expected utility of resulting action profiles:

$$u_i(\pi_i, \pi_{-i}) \coloneqq \mathbb{E}_{a \sim \pi}[u_i(a_i, a_{-i})]$$

Nash equilibrium

- A strategy profile π^* is a Nash equilibrium (NE), if and only if no single player can improve his or her expected utility by unilaterally changing the strategy:

$$\forall i \text{ and } \forall \pi_i \in \Pi_i: \quad u_i(\pi_i, \pi^*_{-i}) \leq u_i(\pi^*_i, \pi^*_{-i})$$

- At least one Nash equilibrium exists in every finite game.

- Nevertheless, NE are generally hard to compute.

Are Nash equilibria always the goal?

Prisoner's dilemma payoff matrix

B / A	B silent	B betrays
A silent	-1 (B) / -1 (A)	0 (B) / -3 (A)
A betrays	-3 (B) / 0 (A)	-2 (B) / -2 (A)

- NE are often hard to compute, but there are other equilibrium notions, like (coarse) correlated equilibria, that are easier to attain.
- Besides being hard to compute, NE may not always lead to desirable outcomes.
  → "Social Dilemma"
- Sometimes, these problems can be circumvented in repeated games.

Types of games – qualitative dimensions

Number of players

- Special case: 2
- “few”
- Many/infinite/continuum

Stateless vs. stateful

- Normal form
- Extensive form
- Repeated game
- Simultaneous moves vs. sequential moves

Types of utility functions

- Zero-Sum
- Nonzero-sum

Number of actions

- Finite
- Countably infinite
- Continuum

Observability of information

- **Complete vs. incomplete**
  - *Complete information:* Players know the rules of the game and others’ payoff functions (but may possibly not know about past moves of other players or outcomes of chance events).
  - *Complete information:* The structure of the game and the players’ utility functions are commonly known (but players may not see all the moves made by other players).
- **Perfect vs. imperfect information**
  - *Perfect information:* Players can observe all events and the full ‘state’ of the game (but may not know about opponents’ goals).
  - *Imperfect information:* Games where some aspect of play is hidden from players (e.g., poker).

Special Case 1

Matrix games

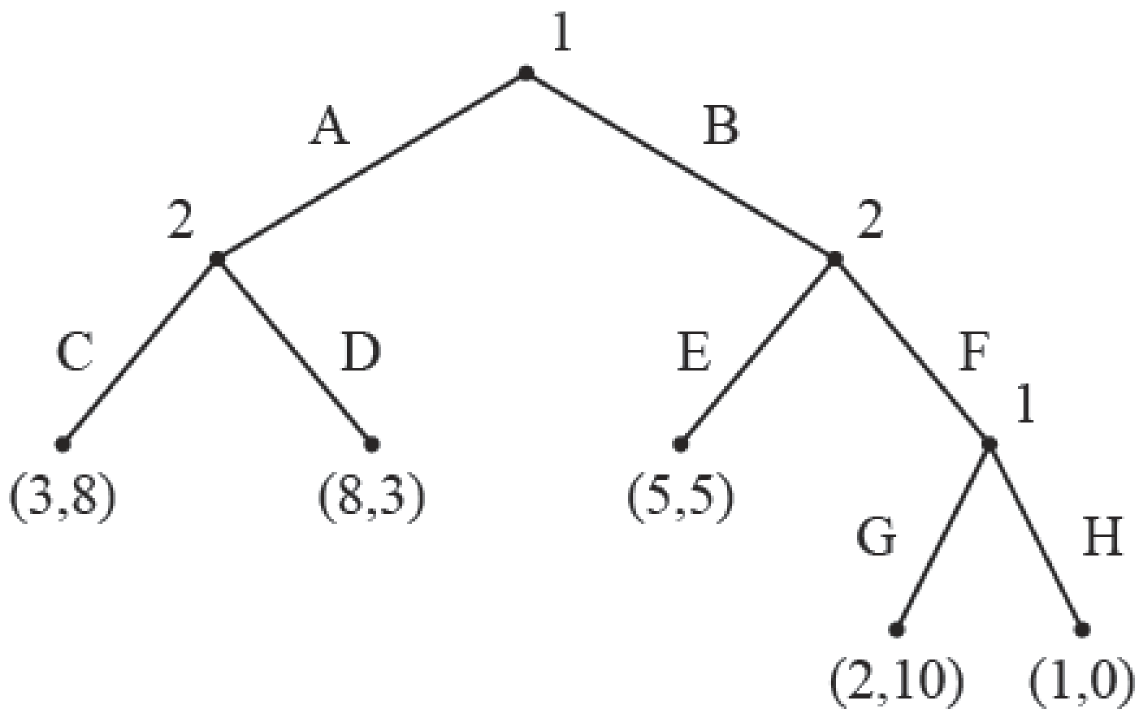
- Finitely many players
- Finitely many actions
- A single, simultaneous move
- Fixed outcomes, no randomness in utility functions
- (Both, complete and perfect information)

B A	B Fight	B Ballet
A Fight	2 3	0 0
A Ballet	0 0	3 2

Special Case 2

Extensive form games

- Sequential actions by players, leading to a "game tree"
- Can have perfect or imperfect information



Special Case 3

Two-player zero-sum games

- Everything that's good for P1 is bad for P2, can be written as

$$u_1 = u \qquad\qquad u_2 = -u$$

- The *minimax* theorem applies:

$$\max_{\pi_1} \min_{\pi_2} u(\pi_1, \pi_2) = \min_{\pi_2} \max_{\pi_1} u(\pi_1, \pi_2)$$

- This makes it easier (in theory) to find and understand NE
- Examples: Two-player board games (Chess, Go, …)
- However: Game can still be extremely large and hard to solve in practice!

Image: James Wainscoat, https://unsplash.com photos/b7MZ6iGIoSI

Special Case 4

Mean field games

- Games with very large populations (e.g., 100s, 1000s, millions of players).
- Each individual agent will have a negligible impact on others.
- Idea: model others as a “continuum” rather than individual actors.
- Name inspired by mean-field theory in physics.
- Examples:
  - Choosing a route on your commute to avoid a traffic jam.
  - Behavior in saturated markets.

Learning in games

> Most of non-cooperative game theory has focused on equilibrium in games […]. This raises the question of when and why we might expect that observed play in a game will correspond to one of these equilibria. One traditional explanation of equilibrium is that it results from analysis and introspection by the players in a situation where the rules of the game, the rationality of the players, and the players' payoff functions are all common knowledge. […]
>
> This book develops the alternative explanation that equilibrium arises as the longrun outcome of a process in which less than fully rational players grope for optimality over time.

- Fudenberg & Levine, The Theory of Learning in Games (1999), page 1

Learning in games

- Most common setting: Learning agents play the game against each other iteratively, in *self-play.*
- Agents *update* their strategies over time, dependent on observed outcomes, in order to improve their own expected utility.
- Update strategies:
  - "Best-response dynamics",
  - improvement updates, often based on (regularized) gradient dynamics, or regret minimization,
  - solving a "meta game", of strategies encountered before (EGTA/PSRO),
  - reinforcement-learning-based updates.

Supervised machine learning

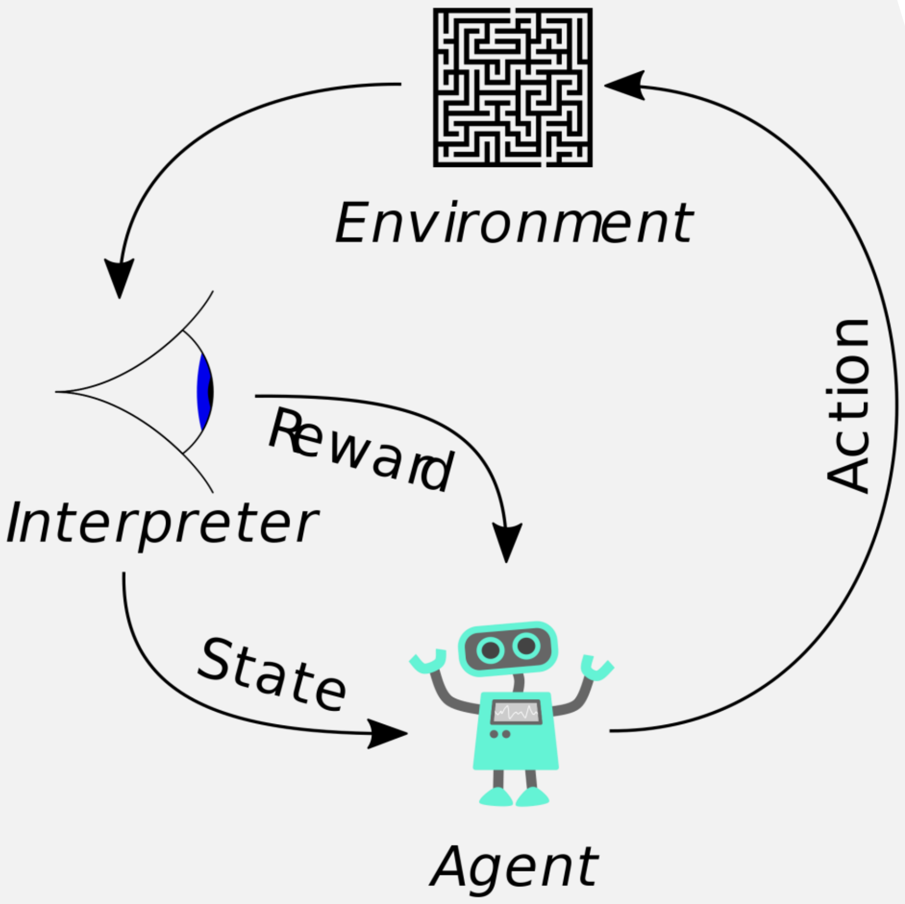
Input: Data set of observations (x_i, y_i)

x_i: features y_i: labels

Goal: Find a *model* $\hat{f}$ such that

$$\hat{f}(x_i) \approx y_i$$

(Single-agent) reinforcement learning



At time *t:*

- agent observes **state** s_t
- Agent chooses **action** $a_t = \pi(s_t)$

Based on a_t, agent receives a **reward** r_{t+1}

Goal: Choose policy π to maximize expected future **return**

$$R_t = \sum_{\tau=1}^{T} \gamma^\tau \, r_{t+\tau}$$

(with $\gamma \in (0,1)$: „discount rate“)

Note: future state depends on current actions!

In practice: randomized actions: $\pi(a_t, s_t)$ = probability of playing a_t when observing s_t

Image: Wikipedia

Formal model for (SA)RL: Markov decision process (MDP)

At each time step t:

- Agent chooses a_t from a set of actions A_t
- **Stationary** state transition distribution given s_t and a_t: $P(s_{t+1} | s_t, a_t)$
- Rewards r_t associated with state transitions

→ Markov property:
„The future is independent of the past, given the present“

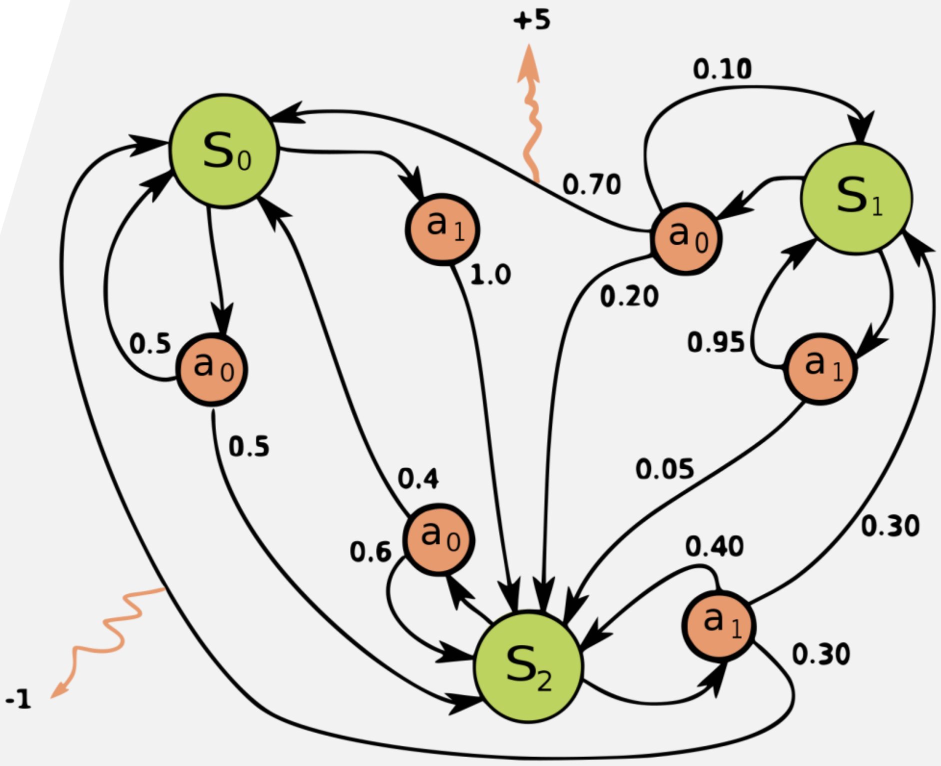


Image: Wikipedia

How to “train” π in (SA)RL?

Data: sampled transitions: $(s_t, a_t, r_{t+1}, s_{t+1})$

Value Iteration

- Estimate the *value* of a state $v(s)$, choose the action that maximizes the expected value of s_{t+1}.
- or directly estimate *state-action values* $q(s, a)$ („Q-Learning“).

Policy Iteration

- Rather than learning values, act directly on the policy function $\pi(s, a)$.
- Most common: „policy gradient“: Estimate $\nabla_\pi \mathbb{E}[R_t]$ from data, then perform gradient ascent.

For MDPs (and some additional details), both converge to optimal policy, in theory.

Practical problem: computational tractability!

(SA)RL: Approximation methods

- Tabular methods:
  - Keep track of q-values for all combinations (s, a), model policies as explicit probability vectors $\pi(s, a) = P(a|s)$.
    → Only feasible for small state and action spaces.
- Approximation Methods:
  - Learn statistical models to represent one or more of v, q, π.
  - Usually using neural networks → „deep reinforcement learning“.
- Examples of deep RL:
  - Deep Q-learning (DQN): Model $q(s, a)$ via a neural net (+ extra tricks).
  - DDPG (deep deterministic policy gradient): Model $q(s, a)$ as one neural net, then use policy iteration on second neural net $\pi(s, a)$ based on *predicted* improvement $\nabla_\pi \, q(s, a)$ → „actor-critic method“.

Multi-agent RL (MARL)

- *Markov game* as a generalization of MDP
- Main Challenge: environment loses its stationarity: state transitions now depend on other agents‘ actions!
  → convergence results from SA-RL break
- Cooperative MARL:
  - Agents share a common utility function, need to learn to work together
- Competitive MARL:
  - Agents have individual utility functions, only interested in their own rewards
- Mixed settings (e.g., team games) are also possible

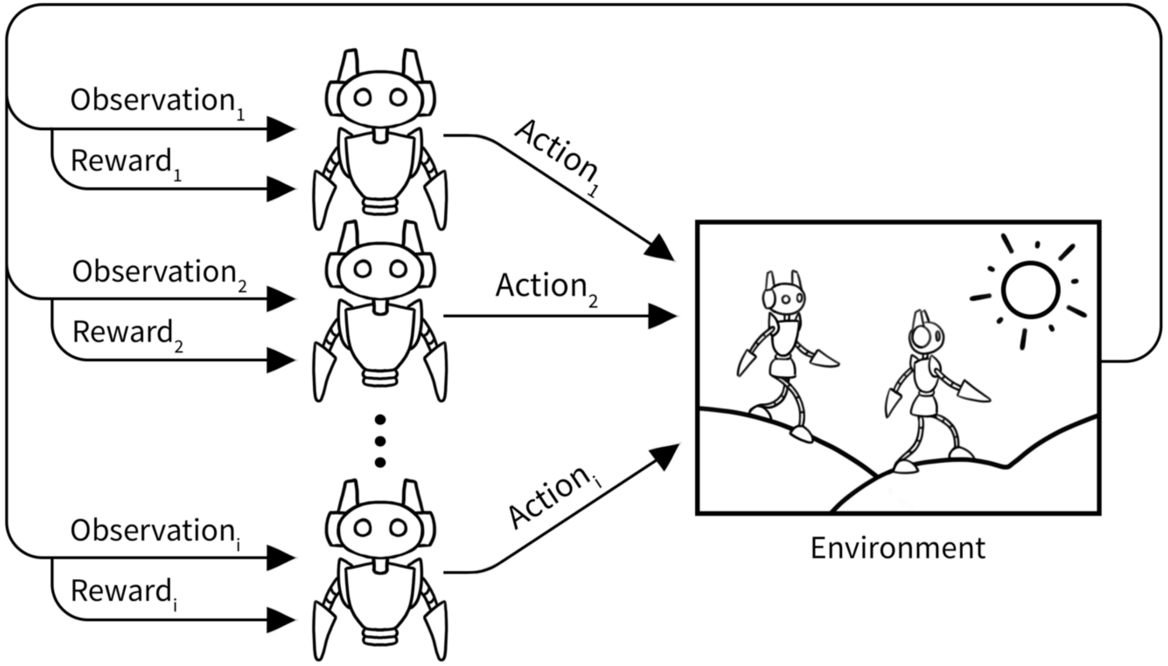


Image: Justin Terry

What's the goal in competitive MARL?

- Equilibrium?
  - For two-player zero-sum games: Yes!
  - Otherwise…?
- Super-human performance?
- Social welfare?

Further Reading: Shoham et al. (2007): If multi-agent learning is the answer, what is the question?

General literature recommendations

- Fudenberg, et al. (1998): *The theory of learning in games*
  - Foundation for equilibrium theory, replicator dynamics and fictious play, normal and extensive form games
  - Can be borrowed at chair, TUM library has a few hard-copies
- Young (2004): *Strategic learning and its limits*
  - Regret and no-regret learning, equilibrium concepts, fictious play
  - Can be borrowed at chair
- Shalev-Shwartz (2011): *Online learning and online convex optimization*
  - Survey paper on online learning that introduces , among other, FTL and FTRL, …
  - Available online
- Nisan, Roughgarden, Tardos, Vazirani (2007): *Algorithmic Game Theory*
  - Collection of guest chapters on different topics in AGT, e.g., complexity, learning, applications, …
- Sutton & Barto (2018): *Reinforcement Learning. An Introduction (2nd edition)*
  - The standard reference for (single-agent) reinforcement learning.
  - Latest edition available for free at http://www.incompleteideas.net/book/the-book.html

Practical Hint:

Accessing paywalled papers

- eAccess is a service provided by the TUM university library. Using eAccess, you can access many scientific papers online that would usually be behind a paywall.
  See https://login.eaccess.tum.edu/login

- If you still cannot access a paper/book you need:
  1. Check the TUM university library catalog.
     Some books are only available as hard copies
  2. Ask your advisor for help.

Agenda

- Seminar format
- Background information
  - Recap: Games and Nash equilibria
  - A brief taxonomy of games
  - What is “learning in games” anyway?
  - A short primer on (multi-agent) reinforcement learning
- Topic assignments
- Q&A

Course Content

Available Topics

01	Normal Form Games & Equilibria
02	Regret Matching & Convergence
03	Zero-Sum, Fictitious Play, MiniMax
04	MDP + Q-Learning
05	Policy Gradients
06	Stochastic Games + Solution Concepts + Shapley Algorithm
07	Lemke-Howson Algorithm
08	Multiplicative Weights & Replicator Dynamics
09	Regret Policy Gradients
10	Collusion
11	Policy Space Response Oracles
12	Sophisticated/Bayesian Learning
13	Inverse Reinforcement Learning
14	Neural Equilbirum Solvers

Topic 01

Normal Form Games & Equilibria

- Introduce basic concepts from Game Theory
  - Normal Form Game
  - Strategy, Dominance, Optimality
  - Solution Concepts:
    - Nash Equilibrium
    - Correlated Equilibrium
    - Coarse Correlated Equilibrium
    - ... others ?
- Illustrate concepts with examples
  - Prisoners’ Dilemma, Matching Pennies, Chicken, etc.

Nils
Fabian
Matthias
Markus
Dima

Topic 02

Regret Matching & Convergence

Nils
Fabian
Matthias
Markus
Dima

- The regret matching algorithm (Hart & Mas-Colell, 2000) seeks to minimize regret about its actions.
  - Regret of not having chosen an action: Difference between the utility of that action and the utility of the action we actually chose, with respect to the fixed choices of other players.
  - It learns from past behavior by favoring actions that have resulted in positive outcomes and avoiding actions that have resulted in negative ones.
- Is the foundation for modern poker bots
- Considers simple normal-form games
- The averaged strategy then converges to a (correlated) equilibrium

Zero-Sum, Fictitious Play, MiniMax

- MiniMax Value Iteration:
  - Determine the best worst-case scenario
  - Solution corresponds to a Nash Equilibrium in 2-Player Zero-Sum games
- Fictitious Play:
  - Agents assume that opponents play stationary strategies
  - Play the best-response against the frequency of opponents play
  - Convergence to the Nash Equilibrium in 2-Player Zero-Sum games
  - Variety of adaptions of this learning procedure

Nils
Fabian
Matthias
Markus
Dima

Literature: Fudenberg and Levine (1998)

Topic 04

MDP + Q-Learning

Nils
Fabian
Matthias
Markus
Dima

- A Markov decision process is the mathematical framework of many fields, such as
  - Reinforcement learning
  - Planning
  - Control theory
  - … and many more
- Introduce MDP mathematics and concepts
  - State, action, transition, reward, policy, etc.
  - What does it mean to “solve” an MDP?
- Discuss Q-learning as one example of an approximate MDP solver
  - $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \left[R(s_t, a_t) + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$

Topic 05

Policy Gradients

- Prerequisite: MDPs
- Policy gradient theorem (PGT):

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log\big(\pi_\theta(s,a)\big) \cdot Q^{\pi_\theta}(\mathrm{s},\mathrm{a})\right]$$

- Many solution methods build upon PGT
- Your task:
  - Understand, explain, illustrate
  - Discuss advantages and disadvantages of policy gradient methods

Nils

Fabian

Matthias

Markus

Dima

Stochastic Games + Solution Concepts + Shapley Algorithm

- Stochastic games (Shapley, 1953) are a natural extension of MDPs to include multiple agents. We want to look at the basic definitions and examples:
  - Stochastic games contain both MDPs and matrix games as subsets of the framework
  - Similar to normal-form games, there are different types (e.g., zero-sum stochastic games)
  - Only for some settings equilibria solutions are known to exist
- The Shapley algorithm can be used to compute equilibria in zero-sum stochastic games
  - The algorithm is based on the proof of the equilibrium existence (Shapley, 1953)
  - The method is similar to value iteration for MDPs (Bowling & Veloso, 2000)

Topic 07

Lemke-Howson Algorithm

- The Lemke-Howson algorithm (Lemke & Howson, 1964) is an algorithm that computes a Nash equilibrium of two-player matrix games.
- It doesn't scale well but does solve the game exactly.
  - Worst case: Number of operations may be exponential in the number of pure strategies.
- It resembles the simplex algorithm (from linear programming).

Nils
Fabian
Matthias
Markus
Dima

Multiplicative Weights & Replicator Dynamics

Nils
Fabian
Matthias
Markus
Dima

- Replicator Dynamics
  - Motivated by the evolution of biological processes
  - Does not necessarily converge to a stable state (i.e., a Nash equilibrium)
- Multiplicative Weights Update:
  - Discrete-time variant of Replicator Dynamics
  - Assign each action a weight
  - Update the weight multiplicatively in each iteration
  - Converges to the Nash Equilibrium in time-average strategies of 2-Player Zero-Sum games
  - Actual strategies diverge from the equilibrium

- Literature: Fudenberg and Levine (1998), Bailey and Piliouras (2018)

Topic 09

Regret Policy Gradients

- Learning parameterized policies (e.g., neural networks) in RL.
- Introduced by Srinivasan et al. (2018) from Deepmind.
- Combination of
  - standard gradient-based learning (following the policy gradient uphill) and
  - regret minimization.
- They consider partially-observable multiagent environments and the algorithm is implemented in Openspiel.

Nils
Fabian
Matthias
Markus
Dima

Topic 10

Collusion

- In general, learning algorithms do not need to converge to Nash equilibria
- Sometimes the process can even lead to undesirable outcomes such as collusion, e.g., firms charge supracompetitive prices (Calvano et al., 2020)
- On the other hand, the choice of algorithms can change the game.
  And collusive outcomes might actually just be equilibrium strategies in the new game (den Boer et al, 2022)
- Therefore, choosing the right learning algorithm and model is crucial.

Topic 11

Policy Space Response Oracles

- Considers the meta-game, where agents are trained via RL in an inner loop and we apply game-theoretic reasoning in the outer loop
- Goal is to arrive at more robust strategies by training a collection of low-level controllers
- Subsumes classical approaches such as independent learning, iterated best response, double oracle, and fictitious play.
- Applying this is costly, but opens up the path to reason about more practical approaches
- „A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning", Lanctot et al., 2017 in Advances in neural information processing systems, 30

Nils
Fabian
Matthias
Markus
Dima

Sophisticated/Bayesian Learning

- Sophisticated learning
  - Many learning algorithms fail to identify patterns (like cycles)
  - Sophisticated Learning explicitly attempts to find those patterns
- Bayesian learning
  - Agents have prior knowledge about the game (opponents' strategies, random state)
  - Represent different beliefs as different models
  - If priors are consistent with the true model, the strategies converge to the true model, which implies convergence to an equilibrium

Literature: Fudenberg and Levine (1998), Wu et al. (2022)

Topic 13

Inverse Reinforcement Learning

Nils
Fabian
Matthias
Markus
Dima

- Idea
  - Assume that optimal actions are already observed by agents
  - Find the reward function that leads to these actions
- Three classes of algorithms
  - Max Margin – Max the margin between optimal policy and value function
  - Bayesian – Use prior knowledge to model the optimal reward
  - Maximum Entropy
- Extensions
  - Multi-Agent Inverse Reinforcement Learning
  - Suboptimal policy demonstrations
  - Feedback-Types

Literature: Adams et al. (2022)

Topic 14

Neural Equilbirum Solvers

- Finding NEs and (C)CEs for normal-form games is hard
- Existing algorithms either may iterate large parts of the joint-action space or are iterative – which fail to converge in some cases
- E.g., meta-learning solvers solve for certain solution concepts in an inner loop, so that fast and constant time calculations to an approximate solution are sufficient
- This work trains a neural network that outputs approximate solutions with a parametrized objective
- „Turbocharging Solution Concepts: Solving NEs, CEs and CCEs with Neural Equilibrium Solvers", Marris et al., 2012 in Advances in neural information processing systems, 35

Nils

Fabian

Matthias

Markus

Dima

Matching of Remaining Topics

Topics ‹› Students

Agenda

- Seminar format
- Background information
  - Recap: Games and Nash equilibria
  - A brief taxonomy of games
  - What is “learning in games” anyway?
  - A short primer on (multi-agent) reinforcement learning
- Topic assignments
- Q&A

Q&A

Contact

nils.kohring@cit.tum.de