Learning in the Presence of Multiple Agents

Giorgia Ramponi

Abstract Reinforcement Learning (RL) has emerged as a powerful tool to solve sequential decision-making problems, where a learning agent interacts with an unknown environment in order to maximize its rewards. Although most RL real-world applications involve multiple agents, the Multi-Agent Reinforcement Learning (MARL) framework is still poorly understood from a theoretical point of view. In this manuscript, we take a step toward solving this problem, providing theoretically sound algorithms for three RL sub-problems with multiple agents: Inverse Reinforcement Learning (IRL), online learning in MARL, and policy optimization in MARL. We start by considering the IRL problem, providing novel algorithms in two different settings: the first considers how to recover and cluster the intentions of a set of agents given demonstrations of near-optimal behavior; the second aims at inferring the reward function optimized by an agent while observing its actual learning process. Then, we consider online learning in MARL. We showed how the presence of other agents can increase the hardness of the problem while proposing statistically efficient algorithms in two settings: Non-cooperative Configurable Markov Decision Processes and Turn-based Markov Games. As the third sub-problem, we study MARL from an optimization viewpoint, showing the difficulties that arise from multiple function optimization problems and providing a novel algorithm for this scenario.

Keywords Multi-agent learning · Reinforcement learning · Inverse reinforcement learning

1 Introduction

Learning is one of the most fascinating open problems of our days. The first thing we can do is to observe the word and try to infer how this process happens in nature. For example, think about how humans learn to perform a task: humans adapt their

G. Ramponi (✉)
Politecnico di Milano, Milano, Italy
e-mail: giorgia.ramponi@polimi.it

C. G. Riva (ed.), *Special Topics in Information Technology*,
PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-031-15374-7_8

behavior to maximize a signal from the environment. Imagine a child who has to learn to ride a bicycle. The child starts by sitting on the seat, and nothing happens. Then she puts her feet on the pedals but rides too slowly, causing the bicycle to lose its balance, and she falls. She thus learned from her experience that she must pedal faster to avoid falling again. This concept is mathematically modeled by *Reinforcement Learning* [20]. However, we need to consider that humans, and animals, do not live alone: we are "social beings", i.e., we act in a social system in which multiple entities interact with each other. Therefore, the actions of each individual can modify the learning process of all the other entities involved. For example, if we decide to buy a stock on the stock exchange, the result of our action will affect not only us but also the entire stock market. It is easy to imagine that these interactions can be very complex, and we can hardly understand how our decisions can affect the world around us. One of the sciences that mathematically model these interactions is called *Game Theory* [8]. From these considerations, we can conclude that in order to create a system that is capable of acting autonomously, we must study how to build an autonomous learning agent (*Reinforcement Learning*) and model how this is influenced by the other entities that surround it (*Game Theory*). *Multi-agent Reinforcement Learning* (MARL) [4, 21] is a bridge between these two worlds. The MARL framework studies the problem of learning by interacting with an unknown system, considering that it is composed of more than one entity.

In this contribution, we provide a summary of the Ph.D. dissertation entitled "Challenges and Opportunities in Multi-Agent Reinforcement Learning" [13], focused on studying the aspects of learning in multi-agent environments. In Sect. 2 we provide the preliminaries and background on RL, MARL, and IRL. Then, we start outline the contribution of the work. For each contribution, we introduce the setting, provide some insights on the proposed algorithm, and discuss the results. In Sect. 3 we analyze the problem of IRL in a multi-agent environment from two viewpoints: first, we consider the problem of learning the intentions of another agent which is learning a new task; second, we propose an algorithm to deal with IRL about Multiple Intentions, i.e., the problem of recovering the reward functions from a set of experts. In Sect. 4 we address the problem of online learning in Stochastic Games. Specifically, we propose an algorithm to deal with the online learning problem in the Configurable Markov Decision Process, where the two entities involved are the configurator of the environment and the RL agent. Then, we introduce a new lower bound on the online learning problem in Stochastic Games, proposing an algorithm that nearly matches this lower bound. Finally, in Sect. 5 we summarize the results and discuss some future research directions. We do not discuss the fourth part "Policy Optimization in Multi-Agent Reinforcement Learning" due to lack of space. In this part, the author reported the result of the paper [17], where they introduced a new optimization algorithm for Continuous Stochastic Games, proving convergence results to equilibrium solutions in general games.

2 Preliminaries

In this section, we provide the necessary background on Reinforcement Learning, Multi-Agent Reinforcement Learning and Inverse Reinforcement Learning.

2.1 (Multi-agent) Reinforcement Learning

Reinforcement Learning (RL) is a framework to learn by trial-and-error in a sequential-decision way: the agent performs an action and receives feedback from the environment. The RL problem involves a learning *agent* (or learner) which interacts with an *environment* during a sequence of discrete-time steps. This interaction is described by three components: the *state* s, the *action* a and the *reward* r (see Fig. 1). The *state* describes the actual configuration of the environment perceived by the agent, which can be a subset of the environment state's characteristics. The *action* consists of the decision taken by the RL agent. The environment responds to every performed action changing the state and giving the agent a *reward*, where the *reward* is numeric feedback of the agent's performances. The interactions are formally described by a Markov Decision Process (MDP) [3, 12] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu, H)$, where $\mathcal{S}$ and $\mathcal{A}$ are respectively the set of states and actions, $\mathcal{P}$ is the probability of transitioning from a state to another taking an action, $\mathcal{R}$ describes the immediate reward obtained in a state taking an action, γ is the discount factor, μ the initial states distribution and H is the horizon (the number of sequential interactions between the agent and the environment). An agent's behavior is described by a policy π which represents the probability of choosing an action a in a state s at a particular timestep h. The performance index of an RL agent's policy is the expected discounted sum of the rewards collected during the interaction with the environment:

$$J(\pi) := \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t \mathcal{R}(s_h, a_h)\right], \tag{1}$$

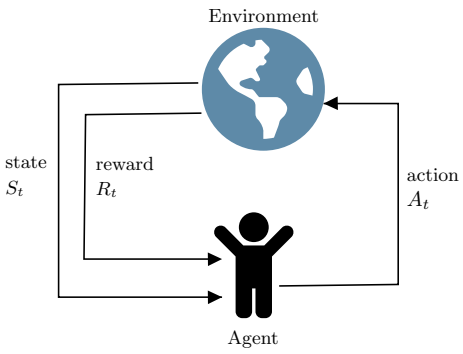

Fig. 1 The agent-environment interaction in a Markov decision process

where the expectation is taken with respect to $s_0 \sim \mu$, $s_{h+1} \sim \mathcal{P}(\cdot|s_h, a_h)$, $a_h \sim \pi(\cdot|s_h)$. When multiple agents are involved the MDP framework is extended to the Markov Game (or Stochastic game) framework. A Markov Game $\mathcal{MG} = (\mathcal{S}_1, \ldots, \mathcal{S}_n, \mathcal{A}_1, \ldots, \mathcal{A}_n, \mathcal{P}, \mathcal{R}_1, \ldots, \mathcal{R}_n, \gamma_1, \ldots, \gamma_n, \mu, H)$ is an MDP where the transition distribution over the next state depends on the actions of all the agents and each agent i has its own reward function $\mathcal{R}_i$. In this case the performances of all the systems are described by equilibrium concepts. The most famous equilibrium is the Nash Equilibrium (NE), where a joint policy $\pi^* = \{\pi_i^*\}_{i=1}^n$ is an NE if:

$$J_i(\pi^*) \geq J_i(\pi_i, \pi_{-i}^*) \quad \forall \pi_i \in \Pi_i, \quad i \in [n]. \tag{2}$$

The idea behind the NE is that each agent cannot improve its performance, when the other agents' policies are fixed. Also other equilibrium concepts are relevant as Stackelberg Equilibrium and Coarse Correlated Equilibrium.

2.2 Inverse Reinforcement Learning

As we wrote in the previous section, RL is a framework to learn how to perform a task described by a reward function. However, in some cases, it is extremely difficult to design a suitable reward function. On the other hand, for many tasks, we already have experts (for example, humans) who know how to accomplish the same task. The Imitation Learning (IL) [10] paradigm aims to exploit the expert information to clone the experts' behavior or formalize the expert's intentions. The IL framework is divided into two main subareas: Behavioral Cloning (BC) and Inverse Reinforcement Learning (IRL). BC, as the name says, aims to clone the expert's behavior in order to use it as a policy. IRL, instead, can recover the expert's reward function to understand its intentions and use this reward function to learn an optimal policy in any environment, even different from the one in which the expert acts. The IRL problem is composed of two agents: an expert who shows how to perform a task and an observer who watches the expert's demonstrations and learns from them the reward function (see Fig. 2). The framework used to model the problem is known

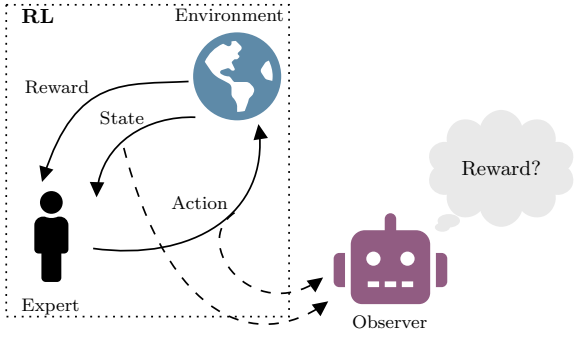

Fig. 2 The expert-observer interaction in the Inverse Reinforcement Learning framework

as Markov Decision Process without Rewards (MDP$\backslash\mathcal{R}$). An MDP$\backslash\mathcal{R}$ is defined as a tuple $\mathcal{M}\backslash\mathcal{R} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mu, H)$ which is the same as an MDP, but without the notion of a reward function. The expert plays a policy π^E which is (nearly) optimal for some unknown reward function $\mathcal{R}$, and we are given a dataset $D = \{\tau_1, \ldots, \tau_m\}$ of trajectories from π^E. The IRL problem consists in recovering the reward function $\mathcal{R}$ that the agent is optimizing.

3 Inverse Reinforcement Learning in Multi-agent Environments

The IRL [9] framework aims at recovering the reward function of an optimal agent. In the classical setting, an expert, i.e., an agent that has already learned a task, makes available a dataset of its interactions with the environment. From this, the IRL algorithm recovers the reward function that the expert is optimizing. When there are multiple agents in the environment, the IRL framework changes its objective too. For example, there could be multiple experts who show their possible different behaviors, leading to an increase in available data, but a necessity to cluster them by their intentions. Or, an agent can be interested in learning the other agents' reward functions, to use it to compute their strategy or to cooperate; however, it has to discover it without waiting for the other agent's convergence to an optimal policy. In this section we briefly describe two algorithms we designed to recover the reward function when there are multiple experts (Sect. 3.1) and when we can observe a learning agent (Sect. 3.2).

3.1 Multiple-intentions Inverse Reinforcement Learning

The first multi-agent framework that we studied is the Multiple-Intentions IRL (MI-IRL) [2] which involves an observer who has access to the demonstrations performed by multiple experts. The observer has to recover the reward functions and use them to cluster the observed agents. Solving this problem is helpful for reasons of explainability since it could be used to understand the similarity between apparently different agents. Moreover, as an immediate benefit, grouping experts who show other behaviors but share the same intent allows for enlarging the set of demonstrations available for the reward recovery process. This has a significant impact on several realistic scenarios, where the only information available is the demonstration dataset, and no further interactions with the environment are allowed.

In [15] we propose a novel batch model-free IRL algorithm, named Σ-*Gradient Inverse Reinforcement Learning* (Σ-GIRL), and then we extend it to the multiple-intention setting. Σ-GIRL, similarly to [11, GIRL], searches for a reward function that makes the estimated *policy gradient* [5] vanish, i.e., a reward function that is a stationary point of the expected return. However, differently from GIRL, Σ-GIRL

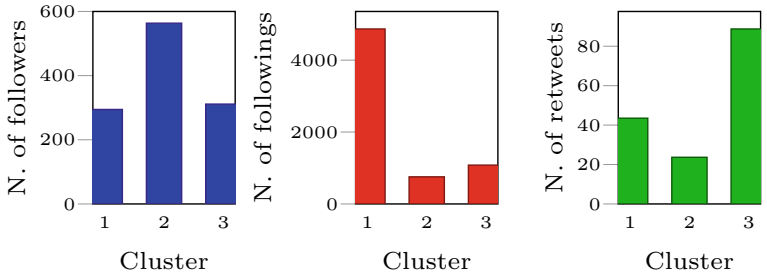

Fig. 3 Twitter clustering statistics. Average number of followers (left), followings (center) and retweets (right) for each cluster

explicitly considers the uncertainty in the gradient estimation process, looking for the reward function that maximizes the likelihood of the estimated policy gradients, under the constraint that such reward is a stationary point of the expected return. Then, we embed Σ-GIRL into the multiple-intention framework by proposing a *clustering algorithm* that, by exploiting the likelihood model of Σ-GIRL, groups the experts according to their intentions. The optimization of the multiple-intention objective is performed in an *expectation-maximization* (EM) fashion, in which the (soft) agent-cluster assignments and the reward functions are obtained through an alternating optimization process. In Fig. 3 we present the result of Σ-GIRL in recovering and clustering the intents of a group of Twitter users. The algorithm divided the 14 Twitter accounts into three clusters. The first cluster is interested in retweeting posts with high popularity. As we can observe from Fig. 3, this cluster represents a *normal* Twitter user: he/she follows many users and has a lower number of followers. In the second cluster, the agents do not retweet often, and the reason could be they have not used the social network much, as they have few retweets and follow a small number of people. In the last cluster (to which a bot, a company, and two HR managers belong) the agents tend to retweet all popular tweets.

3.2 Inverse Reinforcement Learning from a Learning Agent

The second setting that we take into account is the Inverse Reinforcement Learning from a Learner (IRLfL), proposed by [6]. The standard IRL setting assumes that an observer receives the interactions between another agent, the expert, which already knows how to perform the task, and the environment. However, in some cases, the observer can observe the learning process of this other agent, and so it can try to infer the agent's reward function beforehand. In this setting, the observer recovers the reward function from a learning agent and not from an expert. In [6] the authors assume that the learner is learning under an entropy-regularized framework,

Table 1 Reward weights for the autonomous simulate driving scenario

	Recovered weights	Real weights
Time	0.0401	0.0017
Jerk	0.0174	0.0003
Slow	0.0001	0.0000
Crash	0.9424	0.9980

motivated by the assumption that the learner is showing a sequence of constantly improving policies. However, many Reinforcement Learning (RL) algorithms [5] do not satisfy this assumption, and human learning is also characterized by mistakes that may lead to non-monotonic learning process. In our work [14], we proposed an algorithm for this relatively new setting, IRLfL, called Learning Observing a Gradient not-Expert Learner (LOGEL), which is not affected by the violation of the constantly improving assumption. Given that many successful RL algorithms are gradient-based [5] and there is some evidence that the human learning process is similar to a gradient-based method [19], we assume that the learner is following the gradient direction of her expected discounted return. The algorithm learns the reward function that minimizes the distance between the actual policy parameters of the learner and the policy parameters that should be obtained if she were following the policy gradient using that reward function. In [14], we provide a finite-sample analysis that bounds the correctness of the recovered weights. Table 1 reports some results on a simulated autonomous driving task. It is easy to see that, the proposed algorithm succeeds in recovering the correct reward from the driver's trajectories.

4 Online Learning in Multi-agent Reinforcement Learning

In this section we consider the problem of online learning in MARL. In this case, we are not only interested in finding the optimal policies but also in measuring the performance of our algorithm during the learning process. The performance is measured using the *regret*, i.e., comparing the performance of the agent's actual policy with the optimal one. This problem is important in practice where we cannot learn from a simulator or using offline data, but we actually learn interacting with the system. In our work we consider the challenging problem of minimize the regret in a multi-agent game when we can control only one of the agents. We consider two different multi-agent framework. In the first one, called Configurable Markov Decision Process, the two entities are the configurator and the agent. The configurator can partially control the environment, i.e., the transition probability distribution, and the agent is the classical RL agent. The second is a Turn-based Markov Game, where the two agents involved optimize two (possibly different) reward functions.

4.1 Online Learning in Non-cooperative Configurable Markov Decision Processes

In this section, we briefly expose the result published in [16] where we solved an online learning problem in the Configurable Markov Decision process framework. A Configurable Markov Decision process involves two entities, the configurator, and the agent. This framework is motivated by real-world scenarios in which an external supervisor (configurator) can *partially* modify the environment. Previous to [16], the Configurable Markov Decision Processes [7, Conf-MDPs] consists of simultaneously optimizing a set of environmental parameters and the agent's policy to reach the maximum expected return. In many scenarios, however, the configurator does not know the agent's reward and its intention differs from that of the agent. In [16] the authors introduce the Non-Cooperative Configurable Markov Decision Processes (NConf-MDP), a new framework that handles the possibility of having different reward functions for the agent and the configurator. An NConf-MDP allows modeling a more extensive set of scenarios, including all the cases in which the agent and configurator display a non-cooperative behavior, modeled by the individual reward functions. Obviously, this setting cannot be solved with a straightforward application of the algorithms designed for Conf-MDPs that focus on the case in which both entities share the same interests. In this novel setting, the authors consider an online learning problem, where the configurator has to select a configuration, within a finite set of possible configurations, in order to maximize its own return. This framework can be seen as a *leader-follower* game, in which the *follower* (the agent) is selfish and optimizes its own reward function, and the *leader* (the configurator) has to decide the best configuration w.r.t. the best response of the agent. Clearly, to adapt its decisions, the configurator has to receive some form of feedback related to the agent's behavior. The authors analyze two settings based on whether the configurator observes only the agent's actions or also a noisy version of the agent's reward function. For the two settings, they propose algorithms based on the Optimism in the Face of Uncertainty [1] principle, which yield bounded regret.

4.2 Online Learning in General-Sum Stochastic Games

In this setting, we consider the learning problem in two-player general-sum Markov Games when we can control only one agent which is playing against an arbitrary opponent to minimize the regret. Previous works only consider the zero-sum setting, in which the two agents are completely adversarial. However, in some cases, the two agents may have different reward functions without having conflicting objectives. This class of games is called general-sum Markov Games. In our work [18], we show that the regret minimization problem is significantly harder than in standard Markov Decision Processes and zero-sum Markov Games. We derive a lower bound on the expected regret of any "good" learning strategy. The lower

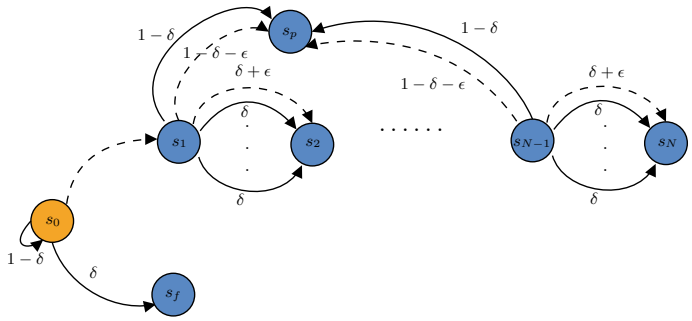

Fig. 4 The Turn-based Stochastic Game for the lower bound. The states belonging to the non-controllable agent are in orange, the ones belonging to the controllable one in blue. The dashed lines corresponds to the transition probabilities taking the optimal action a^*, and the others taking any other action

bound shows the constant dependencies with the number of deterministic policies that the agent can play, which is not present in zero-sum Markov Games and Markov Decision Processes. To have an intuition, we can look at Fig. 4. The controllable agent controls the blue states, while the non-controllable one controls the orange ones. The idea behind the proof is the following: if the controllable agent does not play the optimal policy, the uncontrollable one will always choose the action which leads to the down path, preventing the agent from exploring the environment. After this result, we propose a novel optimistic algorithm that nearly matches the proposed lower bound.

5 Conclusions

The work [13] addressed different problems in MARL, going from IRL to online learning and policy optimization. The MARL framework provides a useful way to model multi-agent decision-making problems such as smart grids, autonomous driving, financial markets, drone delivery, and robotic control problems. The main purpose of the work is to show the flexibility of the MARL framework to model real-world problems compared to the single-agent one, and, on the other hand, how it leads to novel challenges: the learning objective changes, the environment is no more stationary and standard algorithms from single-agent RL cannot be applied. We consider three sub-problems, inspired by the single-agent literature: IRL in Multi-Agent Systems, Online Learning in MARL, and Optimization in MARL. Although we provide novel algorithms for various problems which arise in these contexts, many problems remain unsolved, and also new open questions, practical and theoretical, come out.

Inverse Reinforcement Learning in Multi-Agent systems From the MI-IRL setting, the Σ-GIRL algorithm presents some limitations: we need to specify the number

of existing clusters as hyperparameter, and the algorithm can converge to stationary points which are not local maximizers. In the IRL from a Learner setting, the main challenge arises from the assumption that we know when the agent changes its policy and so a natural extension could be the automatic detection of the policy change. Moreover, if we want to play simultaneously and/or the other agents are not rational, it is necessary to build different algorithms to recover the reward function.

Online Learning in Markov Games There are many future directions in the online learning problem in Stochastic Games. From our work in Non-cooperative Configurable MDPs, a direct follow-up will be extending the approach to deal with continuous state and action spaces using, for example, function approximation. From our findings in general-sum Markov Games, it is clear that the algorithm design and the The resulting performance guarantees heavily depend on any knowledge about the opponents, either known as a priori or obtainable during the learning process. An interesting future direction is to assume to have the possibility to observe other agents' interactions with the environment or having some previous knowledge about the other agents (as having access to a finite set of opponents or considering a larger set of opponents' classes with some regularity assumptions).

Acknowledgements The author acknowledges Marcello Restelli for the support during this work.

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Mach. Learn. **47**(2–3), 235–256 (2002)
2. Babes, M., Marivate, V.N., Subramanian, K., Littman, M.L.: Apprenticeship learning about multiple intentions. In: ICML (2011)
3. Bellman, R.: A markovian decision process. J. Math. Mech. 679–684 (1957)
4. Buşoniu, L., Babuška, R., De Schutter, B.: Multi-agent reinforcement learning: an overview. Innovat. Multi-agent Syst. Appl. **1**, 183–221 (2010)
5. Deisenroth, M.P., Neumann, G., Peters, J., et al.: A survey on policy search for robotics. Found. Trends Robot. **2**(1–2), 388–403 (2013)
6. Jacq, A., Geist, M., Paiva, A., Pietquin, O.: Learning from a learner. In: International Conference on Machine Learning, pp. 2990–2999. PMLR (2019)
7. Metelli, A.M., Mutti, M., Restelli, M.: Configurable Markov decision processes. In: International Conference on Machine Learning, pp. 3491–3500. PMLR (2018)
8. Morgenstern, O., Von Neumann, J.: Theory of Games and Economic Behavior. Princeton University Press (1953)
9. Ng, A.Y., Russell, S.J., et al.: Algorithms for inverse reinforcement learning. In: Icml. vol. 1, p. 2 (2000)
10. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J.: An algorithmic perspective on imitation learning. CoRR (2018). arXiv:abs/1811.06711
11. Pirotta, M., Restelli, M.: Inverse reinforcement learning through policy gradient minimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
12. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley (2014)

13. Ramponi, G.: Challenges and opportunities in multi-agent reinforcement learning (2021)
14. Ramponi, G., Drappo, G., Restelli, M.: Inverse reinforcement learning from a gradient-based learner **33**, 2458–2468 (2020). https://proceedings.neurips.cc/paper/2020/file/19aa6c6fb4ba9fcf39e893ff1fd5b5bd-Paper.pdf
15. Ramponi, G., Likmeta, A., Metelli, A.M., Tirinzoni, A., Restelli, M.: Truly batch model-free inverse reinforcement learning about multiple intentions. In: International Conference on Artificial Intelligence and Statistics, pp. 2359–2369. PMLR (2020)
16. Ramponi, G., Metelli, A.M., Concetti, A., Restelli, M.: Online learning in non-cooperative configurable Markov decision process. In: AAAI Workshop on Reinforcement Learning in Games (2021)
17. Ramponi, G., Restelli, M.: Newton optimization on helmholtz decomposition for continuous games. In: Thirty-Fifth AAAI Conference on Artificial Intelligence (2021)
18. Ramponi, G., Restelli, M.: Learning in markov games: can we exploit a general-sum opponent? In: The 38th Conference on Uncertainty in Artificial Intelligence (2022)
19. Shteingart, H., Loewenstein, Y.: Reinforcement learning and human behavior. Curr. Opin. Neurobiol. **25**, 93–98 (2014)
20. Sutton, R.S., Barto, A.G., et al.: Introduction to Reinforcement Learning, vol. 135. MIT Press Cambridge (1998)
21. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: a selective overview of theories and algorithms (2019). arXiv preprint arXiv:1911.10635