State of the Art on: Configurable Markov Decision Process

Alessandro Concetti, alessandro.concetti@mail.polimi.it

1. Introduction to the research topic

Machine learning (ML) is a subfield of Artificial Intelligence that leverages statistical techniques to develop algorithms able to learn from data. Reinforcement learning (RL) is one of the three basic machine learning paradigms, alongside supervised learning and unsupervised learning. RL studies how software agents ought to take actions in an environment in order to maximize a cumulative reward. In recent years, the interest in Machine Learning has increased more and more both from the academic and industrial world. NeurIPS (Neural Information Processing Systems) and ICML (International Conference on Machine Learning) are the most relevant conferences related to the field of Machine Learning and the attendance in the last year has grown considerably. Other prestigious conferences with a wide spectrum on the AI field, are AAAI (AAAI Conference on Artificial Intelligence) and IJCAI (International Joint Conference on Artificial Intelligence). Relevant journals are the Journal of Machine Learning Research, Journal of Artificial Intelligence Research and Machine Learning (Springer).

1.1. Preliminaries

1.1.1 Markov Decision Processes

Markov Decision Processes (MDPs) [25] are a mathematical framework for modelling sequential decision making used in the Reinforcement Learning field to model the interaction of an agent with the environment. Formally, an MDP is a tuple $(S, A, P, R, \gamma, \mu)$ [27], where S and A represent respectively the set of states and the set of actions that the agent can perform, $P(s'|s, a)$ is the probability distribution representing the probability of ending up in state s' starting from s performing the action a, while $R(s, a)$ is the immediate reward, given the current state s and the performed action a. Moreover $\mu(s)$ is the probability distribution over S, providing the probability of the initial state for each episode, and $\gamma \in (0, 1)$ is the discount factor, which models the interest of the agent in future rewards. In a Reinforcement Learning scenario, the agent performs actions through a policy $\pi(a|s)$, which provides a probability distribution over A given the current state s. The goal of the agent is to find the optimal policy π^* that maximizes the *expected reward* J^π, i.e. the expected discounted sum of the rewards collected over an episode.

$$J^\pi = \frac{1}{1-\gamma} \int_S d^\pi(s) \int_A \pi(a|s) R(s, a) da ds,$$

where $d^\pi(s)$ is the γ-discounted state distribution [28]. Naming $P^\pi(s'|s) = \int_A \pi(a|s)P(s'|s, a)da$, we can recursively define $d^\pi(s)$ as:

$$d^\pi(s) = (1-\gamma)\mu(s) + \gamma \int_S d^\pi(s')P^\pi(s'|s)ds'.$$

1.1.2 Configurable Markov Decision Processes

Configurable Markov Decision Process (Conf-MDP) [18] is a framework that extends MDP in order to deal with configurable environments, i.e. environments characterized by tunable parameters. In practice, this means that there are two entities acting in the environment: an agent who learns the optimal policy and a supervisor whose aim is to configure parameters in order to optimize the agent's learning process. Formally a Conf-MDP is a tuple $(S, A, R, \gamma, \mu, \mathcal{P}, \Pi)$ where (S, A, R, γ, μ) is an MDP without the transition model and $\mathcal{P}$ and Π are the model and policy spaces. The performance of a model-policy pair $(P, \pi) \in \mathcal{P} \times \Pi$ is evaluated through the *expected reward* defined similarly to the MDP case:

$$J^{P,\pi} = \frac{1}{1-\gamma} \int_S d^{P,\pi}(s) \int_A \pi(a|s) R(s,a) da ds$$

where $d^{P,\pi}(s)$ is the γ-discounted state distribution.

$$d^{P,\pi}(s) = (1-\gamma)\mu(s) + \gamma \int_S d^{P,\pi}(s') P^{\pi}(s'|s) ds'$$

1.1.3 Tools

The main programming language used in the field of Machine Learning is Python. It offers several ad-hoc libraries for all possible purposes and it is sustained by an increasingly growing community. The main libraries in scientific fields are SciPy, Matplotlib, Numpy, and Pandas. These libraries offer a variety of algorithms and data structures suitably optimized for mathematical and algebraic purposes. More specific libraries in the field of Machine Learning are Tensorflow [1], Scikit-learn [22], Caffe [15] and Torch [5] which enable researchers and programmers to train machine learning models in very handy and intuitive way. Moreover, there are also many toolkits used in the field of Reinforcement Learning such as OpenAi gym [3], Baselines [7], Garage [9] and MushroomRL [6] which offer several environments and main RL algorithms implemented to ease the development and testing of new RL models.

1.2. Research topic

The Conf-MDP framework proposed in [18] is based on the assumption that the supervisor and the learning agent optimize the same reward function. In some sense, the two entities act in a fully-cooperative scenario. However, a cooperative multi-agent approach is not suitable for solving the problem since the supervisor acts at a different level and could be transparent to the agent who learns in a non-stationary environment. Furthermore, if the supervisor and the learning agent optimized two different reward functions, the Conf-MDP would not be suitable in its canonical formulation. In this case, it would be reasonable to shift to a non-cooperative multi-agent approach where a game between the supervisor and the learning agent is established. The supervisor aims to configure the environment by optimizing its reward function and taking into account the reward function of the learning agent which has to be modeled through one of the main Inverse Reinforcement Learning (IRL) techniques [21]. On the other hand, the learning agent performs actions in the environment by optimizing its reward function and it may be unaware or aware of the presence of a non-cooperative supervisor. In the former case, the learning agent ignores the presence of the supervisor and performs actions in a non-stationary environment induced by the supervisor. In the latter case, the learning agent is aware of being in a non-cooperative game with the supervisor and this could lead to possible strategic behaviors. In practice, this means that the agent could perform some actions only to deceive the supervisor and end up in a new better configuration.

2. Main related works

2.1. Classification of the main related works

The scenario presented in Section 1.2 is the natural extension of the works brought forward in the field of Conf-MDP [18, 16, 17] and it deviates from those for the multi-agent perspective, the usage of IRL and the introduction of a strategic behavior due to the agent's awareness of being in a competitive game. These three differences characterized the three main research areas to which the concerned topic is related: *Multi-agent systems* (MAS), *Inverse Reinforcement Learning* (IRL) and *Game Theory* (GT).

2.2. Brief description of the main related works

This section aims to present the major works concerning the four main research fields related with the presented research topic: Conf-MDP, MAS, IRL and GT. First of all, a briefly description of the Conf-MDP literature will be presented to be aware of the main open issues in this field. Then, the literature concerning MAS will be analyzed to understand what are the main techniques to deal with multi-agent environments. In addition, as discussed in Section 1.2, IRL techniques will be leveraged by the supervisor to model the reward function of the learning agent and they will be explored in this section. Finally, the main GT frameworks will be analyzed to characterize the possible strategic behavior between the supervisor and the agent.

Conf-MDP In [18] it is possible to find the first formulation of Conf-MDP, as briefly described in section 1.1.2. In [16] a new learning algorithm, called Relative Entropy Model Policy Search (REMPS), has been presented to deal with real-world continuous environments. The algorithm takes inspiration from REPS [23] and it is divided in two phases: *optimization* and *projection*. The first phase aims at identifying a new stationary distribution for the Conf-MDP that maximizes the total reward in a neighborhood of the current stationary distribution. This distribution may fall outside the space of representable distributions, given the parametrization of the policy and of the configuration. Hence, the projection phase performs a moment projection in order to find an approximation of this stationary distribution in terms of representable policies and configurations.

In the most recent work [17], Conf-MDP has been leveraged to simplify the identification of the policy of an agent. In this context, configuring the environment can be used to induced some behavior of the agent and better understand its policy. In particular, this approach turned out to be successful to identify the set of policy parameters the agent can control. When the agent reaches the optimal policy, some parameters are set to zero, but there could be two possible reasons for this to happen: the parameter is useless to fulfill the goal or it is not controllable by the agent. Configuring the environment has been used to induce the agent revealing which parameters it can control.

MAS The problem of learning in multi-agent environments needs a paradigm-shift and the classical MDP has to be extended to deal with these contexts. The classical framework used in these scenarios is the Stochastic Game (SG). Multi-Agent Reinforcement Learning (MARL) is a growing research area in the field of Machine Learning. In [4] many of the recent successes in MARL are collected, both in fully-cooperative, fully-competitive and mixed scenarios. Another reason for the growing of interests in MAS is its synergistic compatibility with Deep Learning (DL), leading to a new area of research called Multi-Agent Deep Reinforcement Learning (MADRL) [12], where DL techniques are used for the analysis of emergent behaviors, learning communication, learning cooperation or modelling of other agents. In [11] a new classification of works around this topic is proposed. MAS approaches can be divided into five categories in increasing order of sophistication: Ignore, Forget, Respond to target opponents, Learn opponent models, Theory of mind.

IRL Russell formulates the problem of IRL [26] as follows. *Given 1) measurements of an agent's behavior over time, in a variety of circumstances, 2) measurements of the sensory inputs to that agent; 3) a model of the physical environment (including the agent's body). Determine the reward function that the agent is optimizing.* Hence, the goal of IRL is to recover the unknown reward function from the expert's demonstrations. The main techniques of IRL and Behavioral Cloning (BC) are collected in [21]. The main methods in IRL can be clustered in two main area: *Model-based* and *Model-free*. Model-based approaches, such as [2, 10], require access to the environment and they are relatively simple to implement when system dynamics are known. However, it is challenging to apply these methods to domains with non linear dynamics, which are very hard to estimate. On the other hand, Model-free IRL methods, such as [24, 19], do not require prior knowledge of the system, but they are time consuming and computationally expensive due to the sampling-based approach.

GT Game theory is the study of mathematical models of strategic interaction among rational decision-makers. Adversarial models are becoming more and more popular in the field of Machine Learning because of many recent successes (e.g Generative Adversarial Networks [20] or Generative Adversarial Inverse Reinforcement Learning [13]). Stackelberg leadership model, presented in [29], is a GT framework composed of two agents: a leader and a follower. The leader plays the first move and then the follower plays its rational response. In [8] important convergence results of the learning dynamics in Stackelberg Games are presented, investigating the relationship between Nash and Stackelberg equilibrium in zero-sum games and providing a new gradient-based update for the leader. Another interesting branch of GT research that shares common ground with Conf-MDP is the Mechanism Design Theory, sometimes called Reverse Game Theory since the objectives-first approach to design mechanisms toward the desired objectives. The mechanism design literature [14] models the interaction of individuals using game theoretic tools, where the institutions governing interaction are modeled as mechanisms.

2.3. Discussion

The main open issue in Conf-MDPs is the unexplored possibility of having a non-cooperative relationship between the supervisor and the agent. Braking the cooperation hypothesis can lead to interesting strategic behavior that could be leveraged in many real-world scenarios (e.g. e-commerce). However, in order to deal with those scenarios two main facts have to be taken into account: 1) The configurator could not know the reward function of the agent and has to model it in order to optimize its own reward function. 2) The agent could be aware to act in a game and this leads to a strategic behavior of the agent that could perform some actions only to make the configurator change parameters in a positive way for its own goal.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016), pp. 265–283.

[2] Abbeel, P., and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning* (2004), p. 1.

[3] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

[4] Bu, L., Babu, R., De Schutter, B., et al. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38*, 2 (2008), 156–172.

[5] Collobert, R., Kavukcuoglu, K., and Farabet, C. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop* (2011), no. CONF.

[6] D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M., and Peters, J. Mushroomrl: Simplifying reinforcement learning research. *arXiv preprint arXiv:2001.01102* (2020).

[7] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. `https://github.com/openai/baselines`, 2017.

[8] Fiez, T., Chasnov, B., and Ratliff, L. J. Convergence of learning dynamics in stackelberg games. *arXiv preprint arXiv:1906.01217* (2019).

[9] garage contributors, T. Garage: A toolkit for reproducible reinforcement learning research. `https://github.com/rlworkgroup/garage`, 2019.

[10] Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems* (2016), pp. 3909–3917.

[11] Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183* (2017).

[12] Hernandez-Leal, P., Kartal, B., and Taylor, M. E. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems 33*, 6 (2019), 750–797.

[13] Ho, J., and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems* (2016), pp. 4565–4573.

[14] Jackson, M. O. Mechanism theory. *Available at SSRN 2542983* (2014).

[15] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), pp. 675–678.

[16] Metelli, A. M., Ghelfi, E., and Restelli, M. Reinforcement learning in configurable continuous environments. In *International Conference on Machine Learning* (2019), pp. 4546–4555.

[17] Metelli, A. M., Manneschi, G., and Restelli, M. Policy space identification in configurable environments. *arXiv preprint arXiv:1909.03984* (2019).

[18] Metelli, A. M., Mutti, M., and Restelli, M. Configurable markov decision processes. *arXiv preprint arXiv:1806.05415* (2018).

[19] Metelli, A. M., Pirotta, M., and Restelli, M. Compatible reward inverse reinforcement learning. In *Advances in Neural Information Processing Systems* (2017), pp. 2050–2059.

[20] Mirza, M., and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[21] Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics 7*, 1-2 (2018), 1–179.

[22] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[23] Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010).

[24] Pirotta, M., and Restelli, M. Inverse reinforcement learning through policy gradient minimization. In *Thirtieth AAAI Conference on Artificial Intelligence* (2016).

[25] Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[26] Russell, S. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory* (1998), pp. 101–103.

[27] Sutton, R. S., and Barto, A. G. Reinforcement learning: An introduction.

[28] Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (2000), pp. 1057–1063.

[29] Von Stackelberg, H. *Market structure and equilibrium*. Springer Science & Business Media, 2010.