

Automated Reasoning for Reinforcement Learning Agents in Structured Environments

Alessandro Gianola¹, Marco Montali¹ and Matteo Papini²

¹Free University of Bozen-Bolzano, Bolzano, Italy

²Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Designing agents that are both adaptive and trustworthy is a long-standing problem at the intersection of symbolic AI and Machine Learning. In this position paper, we discuss several benefits of combining automated reasoning and reinforcement learning techniques to formally verify agents' behavior in structured environments, both *during* and *after* the learning process. These are systems where agents have access to an explicit structure representing what they know about the world. Since we care both about the verifiability and the efficiency of the learning process, we argue why it is crucial to efficiently integrate complex structures in the learning algorithms themselves.

Keywords

Reinforcement Learning, Automated Reasoning, Formal Methods, Verification, Data-aware Processes

1. Introduction

Reinforcement Learning (RL) [1] has emerged as one of the most important techniques for equipping agents with learning capabilities used to maximize a reward while operating in an unknown environment. In this respect, RL incarnates one of the main distinctive features of Machine Learning (ML) approaches, namely that the complexity of the world is not explicitly represented, but it is instead reflected through a number of implicit soft constraints that emerge from the learning process. This poses a twofold challenge when it comes to *hard constraints* either related to the environment in which the agent operates, or to the behavior that the agent is expected to show during and/or at the end of the learning phase. Conventional RL-based approaches struggle in dealing with this challenge: mainly based on statistical inference, they cannot prevent the learning agents from reaching unsafe/undesired/dangerous configurations [2]. In fact, in each state of a Markov decision process, every action is in principle executable, with an associated probability. This entails that, especially during learning, every possible execution is allowed, inducing a huge state space that contains all possible configurations, including the unsafe ones (or even those that cannot be reached in the real world).

Safety-oriented extensions of the conventional RL framework have consequently been studied, but they typically achieve safety in a probabilistic sense, that is, guaranteeing that there is a high probability for the system to be safe [3, 4, 5, 6], or that safety will hold only in the limit policy

OVERLAY 2021: 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, September 22, 2021, Padova, Italy

✉ gianola@inf.unibz.it (A. Gianola); montali@inf.unibz.it (M. Montali); matteo.papini@upf.edu (M. Papini)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

obtained at the end of the learning process [7, 8]. In fact, existing symbolic approaches do not explicitly distinguish between two different, but equally important, temporal dimensions: the temporal dimension related to the system dynamics with a fixed agent policy, and the temporal evolution of the policy as a result of the learning process. Nevertheless, in the conventional RL literature both the aforementioned dimensions have been considered [9, 10, 11, 12, 13, 14, 15].

The verification of dynamical systems against some property of interest is traditionally investigated in the realm of formal methods, from general, domain-independent properties (such as safety, liveness, and fairness) [16, 17] to domain-specific properties typically expressed using temporal logics [18], possibly also predicating over the strategies agents may adopt [19, 20]. The application of techniques from formal methods and knowledge representation in an RL-based setting has so far typically revolved around guaranteeing that a single agent [7, 21, 8, 22] or multiple agents [23] in the limit ensure temporal specifications of interest. They do so in a generic context where no assumption is made on the structure of the state space.

Our interest is to study systems where agents come with an explicit *structure*, for example relational data representing what the agents know about the world (in the style of [24, 17]), and it is to investigate such systems both *during and after the learning process*. This is for example relevant for applications like business process management, where the states are process states and there are some constraints that must not only be satisfied in the limit but also during learning. This requires to incorporate at once background knowledge on the process itself in the form of (safety) constraints that should always be respected, preconditions and effects dictating how states can evolve into other states (and, conversely, which transitions do not exist at all), and finally also on goals that should be satisfied in the limit.

All in all, our interest is on verifying properties that are significant for learning agents carrying a structure in the aforementioned sense. This is explained in the next section.

2. Research Directions and Challenges

Currently, RL algorithms considered in symbolic approaches (e.g., [21, 8]) do not exploit the structural information on the state space that the agents explore. Moreover, in the conventional RL theory only simple structures are taken in consideration (for example *linear Markov Decision Processes*, where the transition probabilities are linear in some given state features [25, 26]).

Structural information can help improve the efficiency of the learning process, but only if it is correctly incorporated in the learning algorithm. Indeed, learning from complex structured data with general-purpose algorithms can cause an explosion of the number of states. Designing learning algorithms with a notion of structure is then crucial to prune the state space. For example, take a personalized recommendation problem where states are users, actions are items to recommend and rewards encode user satisfaction. If we simply enrich states with user information (age, sex, etc.), the number of possible states explodes. However, if we find a simpler relationship between user *features* and system dynamics, and the learning algorithm is designed to exploit this structure, efficiency can be even better than by considering each user as an atomic state. For instance, in the special case of a linear relationship, the effective dimension of the problem goes from S (the number of states) to the number of features $d \ll S$ [25].

On the other hand, in the last decades a lot of effort has been made in formal methods to

automatically verify structured systems combining processes and data [24, 27]. Specifically, formal techniques based on *automated reasoning* have been successfully employed [28, 29]. The models formalizing these systems are usually not only able to represent the dynamics of evolving (business) processes, but also the structured data storage that those processes interact with. Unfortunately, these models are usually ‘static’ in the sense that, once they have been created by the users, they cannot be modified over time and improved by receiving some feedback from the real environment that they are intended to represent. Moreover, traditional formal frameworks are not able to natively formalize a learning process as the ones studied in ML.

A holistic framework for structured systems combining automated reasoning and reinforcement learning is missing. Obtaining a working combination would bring several benefits: (i) to exploit automatic analysis that is not affected by statistical uncertainty as learning agents are during their exploration; (ii) to employ highly efficient tools like state-of-the-art model checkers or SMT-solvers for performing the automatic analysis; (iii) to leverage well-studied formalisms for expressive structured systems, like those studied for data-aware processes in [28, 30, 31].

In the remainder of the section we focus in more detail on specific problems that arise in the context of structured RL. Before doing so, we remark the common challenges emerging in all such problems. First, the problem of verifying structured systems like data-aware processes is intrinsically difficult [24, 27], even disregarding the learning aspects. Adding probabilistic features can only make it more challenging. Second, adding structure to states is a genuine novel aspect in RL. From the RL perspective, it is still unclear how to equip learning algorithms with the ability of modeling and using complex, structured states and actions (e.g., activities guided by a business process), and which form of transition systems consequently emerge.

Safe Structured Reinforcement Learning. The most natural problem to consider is that of safety verification of a single agent. Safety is particularly important for RL agents that interact with the real world, since they could harm themselves and their environment [32]. Physical agents like robots [33] or autonomous cars represent the most prominent example, since they could directly harm human operators. However, also software agents responsible of critical decisions (e.g., in healthcare [34]) must be subject to strict safety constraints. A literature on safe learning has developed within RL [2]. In verification, this is the problem of determining whether it is possible that an agent reaches some undesired configuration. This becomes particularly challenging in case of structured agents carrying data, as data bring a dimension of infinity that is difficult to tame even in the conventional setting where learning is not considered [24, 27].

In [7, 8, 23], the learning process and the temporal specification are introduced separately and *only then* combined: this creates a new learning problem where the original reward function needs to be maximized together with the probability of satisfying the temporal specification. Differently from these approaches, when dealing with data-aware learning agents, it appears to us more natural to study the case where the logical specification of the structure and the reward function are intrinsically connected: this can happen, for example, when the actions that the agent can perform are guarded by conditions that can query not only the relational database but also the reward obtained so far. Another natural way for equipping agents with structures is to consider not only purely non-deterministic actions with some associated probability, but also temporal/dynamic constraints over their mutual order, as dictated by a business process model.

Incorporating these features in algorithms becomes crucial for efficiency as discussed below.

RL efficiency. As most ML problems, RL is affected by the *curse of dimensionality*: as the number of variables describing states grows, the number of states explodes combinatorially. So, even if the sample complexity (the amount of data required to learn an optimal policy) of classical RL algorithms is polynomial in the number of states, these methods are unpractical for most real problems. The true complexity of any learning problem depends on the *structure* of the state space, i.e., on the similarity of states w.r.t. their associated rewards and transitions. Structure can be exploited to learn with fewer data, recalling that every data sample acts as a representative for (possibly infinitely) many other data that induce the same behavior. However, a RL algorithm can only exploit this feature if it can actually incorporate this structure-related information. The promise of deep RL [35] is to learn the structure along the decision policy, everything from data, by exploiting the representational power of deep neural networks and the efficiency of first-order optimization methods in high-dimensional settings. Unfortunately, the theoretical understanding of deep learning is still limited. Recently, RL theory has focused on simple structures, such as linear ones [25], where it is possible to derive sample complexity guarantees that do not depend on the number of states. The challenge is to obtain similar results for more complex structures, such as the relational databases and other data-storing structures.

RL-specific Properties. A fundamental problem in RL is the exploration-exploitation trade-off. To converge to optimal policy, an agent must sufficiently try the available options (*explore*) before settling with what appears to be the best solution (*exploit*). RL theory largely studies how to make exploration efficient, i.e., to limit the amount of time the agent must spend (equivalently, data it must collect) in exploratory behavior. However, a sufficient amount of exploration must always be guaranteed, and this often means the agent must be able to visit all states [36, 37]. The problem of sufficient exploration must be considered both in the modeling of the learning process (the definition of the state space) and in the design of the learning algorithm (the policies the agent will execute while learning). Formalizing and verifying such conditions requires to go beyond safety, and to appeal to more sophisticated properties, e.g., liveness and ergodicity [38].

Quantifying over learning steps. A recurring theme in what discussed so far is the fact that sometimes properties of interest have to be always enforced/verified, or only satisfied in the limit (i.e., once the agent has completed its learning phase). This is even trickier, as properties may be used to influence the learning process itself (perturbing the system), or instead be checked without any intervention on the original system. This problem has been considered in the literature. E.g., ensuring that an agent is safe not only in the limit (as in [7]), but also during learning (as in [39]), is what Amodei et al. [40] call *safe exploration* in their survey of AI safety problems. However, a systematic exploration of this space of possibilities is missing. We intend to explore this space with two key features. On the one hand, we intend to express properties in a logic that provides two different types of temporal quantification: one quantifying over the system dynamics when the agent employs a fixed policy; the other over the steps of the learning process, which in turn represent different refinements of the agent policy. On the other hand, we want to equip agents with the ability of checking such properties, influencing its behavior.

References

- [1] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
- [2] J. García, F. Fernández, A comprehensive survey on safe reinforcement learning, *J. Mach. Learn. Res.* 16 (2015) 1437–1480.
- [3] G. Anderson, A. Verma, I. Dillig, S. Chaudhuri, Neurosymbolic reinforcement learning with formally verified exploration, in: *Proc. of NeurIPS 2020*, 2020.
- [4] T. M. Moldovan, P. Abbeel, Safe exploration in Markov decision processes, in: *Proc. of ICML 2012*, 2012.
- [5] Y. Chow, O. Nachum, E. A. Duéñez-Guzmán, M. Ghavamzadeh, A Lyapunov-based approach to safe reinforcement learning, in: *Proc. of NeurIPS 2018*, 2018, pp. 8103–8112.
- [6] J. Achiam, D. Held, A. Tamar, P. Abbeel, Constrained policy optimization, in: *Proc. of ICML 2017*, volume 70, PMLR, 2017, pp. 22–31.
- [7] M. Hasanbeig, A. Abate, D. Kroening, Certified reinforcement learning with logic guidance, *CoRR abs/1902.00778* (2019). URL: <http://arxiv.org/abs/1902.00778>.
- [8] M. Hasanbeig, D. Kroening, A. Abate, Towards verifiable and safe model-free reinforcement learning, in: *Proc. of OVERLAY 2019*, volume 2509, CEUR-WS.org, 2019, p. 1.
- [9] V. S. Borkar, An actor-critic algorithm for constrained Markov decision processes, *Syst. Control. Lett.* 54 (2005) 207–213.
- [10] J. E. Moody, M. Saffell, Learning to trade via direct reinforcement, *IEEE Trans. Neural Networks* 12 (2001) 875–889.
- [11] S. M. Kakade, J. Langford, Approximately optimal approximate reinforcement learning, in: *Proc. of ICML 2002*, Morgan Kaufmann, 2002, pp. 267–274.
- [12] P. S. Thomas, G. Theodorou, M. Ghavamzadeh, High confidence policy improvement, in: *Proc. of ICML 2015*, volume 37, JMLR.org, 2015, pp. 2380–2388.
- [13] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, P. Moritz, Trust region policy optimization, in: *ICML*, volume 37, JMLR.org, 2015, pp. 1889–1897.
- [14] F. Berkenkamp, M. Turchetta, A. P. Schoellig, A. Krause, Safe model-based reinforcement learning with stability guarantees, in: *Proc. of NeurIPS 2017*, 2017, pp. 908–918.
- [15] M. Papini, M. Pirodda, M. Restelli, Smoothing policies and safe policy gradients, *CoRR abs/1905.03231* (2019).
- [16] C. Baier, J. Katoen, Principles of model checking, MIT Press, 2008.
- [17] D. Calvanese, S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, Verification of data-aware processes: Challenges and opportunities for automated reasoning, in: *Proc. of ARCADE 2019*, volume 311, EPTCS, 2019.
- [18] A. Pnueli, The temporal logic of programs, in: *Proc. of FOCS '77*, IEEE Computer Society, 1977, pp. 46–57.
- [19] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, *J. ACM* 49 (2002) 672–713.
- [20] K. Chatterjee, T. A. Henzinger, N. Piterman, Strategy logic, *Inf. Comput.* 208 (2010) 677–693.
- [21] G. D. Giacomo, L. Iocchi, M. Favorito, F. Patrizi, Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications, in: *Proc. of ICAPS 2018*, AAAI Press, 2019, pp. 128–136.

- [22] M. Hasanbeig, D. Kroening, A. Abate, Deep reinforcement learning with temporal logics, in: Proc. of FORMATS 2020, volume 12288 of LNCS, Springer, 2020, pp. 1–22.
- [23] L. Hammond, A. Abate, J. Gutierrez, M. J. Wooldridge, Multi-agent reinforcement learning with temporal logic specifications, in: Proc. of AAMAS '21, ACM, 2021, pp. 583–592.
- [24] D. Calvanese, G. D. Giacomo, M. Montali, Foundations of data-aware process analysis: a database theory perspective, in: Proc. of PODS 2013, ACM, 2013, pp. 1–12.
- [25] C. Jin, Z. Yang, Z. Wang, M. I. Jordan, Provably efficient reinforcement learning with linear function approximation, in: Proc. of COLT 2020, volume 125, PMLR, 2020, pp. 2137–2143.
- [26] L. F. Yang, M. Wang, Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound, CoRR abs/1905.10389 (2019).
- [27] A. Deutsch, R. Hull, Y. Li, V. Vianu, Automatic verification of database-centric systems, ACM SIGLOG News 5 (2018) 37–56.
- [28] D. Calvanese, S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, SMT-based verification of data-aware processes: a model-theoretic approach, Math. Struct. Comput. Sci. 30 (2020) 271–313.
- [29] D. Calvanese, S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, Model completeness, covers and superposition, in: Proc. of CADE 27, volume 11716 of LNCS, Springer, 2019, pp. 142–160.
- [30] D. Calvanese, S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, Formal modeling and SMT-based parameterized verification of data-aware BPMN, in: Proc. of BPM 2019, volume 11675 of LNCS, Springer, 2019, pp. 157–175.
- [31] S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, Petri nets with parameterised data, in: Proc. of BPM 2020, volume 12168 of LNCS, Springer, 2020, pp. 55–74.
- [32] G. Dulac-Arnold, D. J. Mankowitz, T. Hester, Challenges of real-world reinforcement learning, CoRR abs/1904.12901 (2019).
- [33] D. Büchler, S. Guist, R. Calandra, V. Berenz, B. Schölkopf, J. Peters, Learning to play table tennis from scratch using muscular robots, CoRR abs/2006.05935 (2020).
- [34] T. Zhu, K. Li, L. Kuang, P. Herrero, P. Georgiou, An insulin bolus advisor for type 1 diabetes using deep reinforcement learning, Sensors 20 (2020) 5058.
- [35] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nat. 518 (2015) 529–533.
- [36] T. Jaksch, R. Ortner, P. Auer, Near-optimal regret bounds for reinforcement learning, J. Mach. Learn. Res. 11 (2010) 1563–1600.
- [37] A. Agarwal, S. M. Kakade, J. D. Lee, G. Mahajan, Optimality and approximation with policy gradient methods in markov decision processes, in: Proc. of COLT 2020, volume 125, PMLR, 2020, pp. 64–66.
- [38] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, 1994.
- [39] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, U. Topcu, Safe reinforcement learning via shielding, in: Proc. of AAAI 2018, AAAI Press, 2018, pp. 2669–2678.
- [40] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, D. Mané, Concrete problems in AI safety, CoRR abs/1606.06565 (2016).