Reinforcement Learning Algorithms for Autonomous Adaptive Agents

Doctoral Consortium

Sindhu Padakandla
Indian Institute of Science
Bangalore, India
sindhupr@iisc.ac.in

ABSTRACT

Intelligent agents are being designed to automate many tasks - for e.g., traffic signal control, vehicle driving, inventory control and are also being used in improving lives of people - like in healthcare, agriculture, wildlife protection etc. The widespread deployment of intelligent agents requires that we minimize the bottlenecks which affect their performance and utility. Motivated by this challenge, my thesis proposes new algorithms and methods which helps the agent in efficiently operating in the real-world and also during interaction with humans. My work has shown significant improvements in the performance of deployed agents, when operating in real world.

KEYWORDS

Reinforcement learning, non-stationary environments, continual learning, deep reinforcement learning, data efficiency

ACM Reference Format:
Sindhu Padakandla. 2020. Reinforcement Learning Algorithms for Autonomous Adaptive Agents. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Autonomous agents driven by learning techniques are increasingly being designed for sequential decision-making applications ([7, 12]) as well as for data intensive applications [8]. Our objective is to solve problems in autonomous systems that can be tackled using data-driven sequential decision-making learning techniques. For e.g., in vehicular traffic signal control [7], an autonomous agent can pick an appropriate green signal duration for all lanes at a traffic junction based on input from visual sensors as well as physical sensors embedded in lanes to measure traffic density. Hence the data collected by a network of sensors combined with apt signal duration (agent decision) can lead to better traffic management. Similarly, a robot senses the surrounding terrain and alters its limb trajectories [12]. However, there are several issues which arise when we think of intelligent agents controlling autonomous systems, the first being *efficiency of the learning techniques* - the agent needs to work with minimal amount of real-time data and computational power. The second aspect is that of *continual* learning - when deployed in real-world, the agent has to deal with dynamically changing operating conditions (see [11]) of the autonomous system, in which case, the agent needs to adapt quickly to the changing conditions and additionally should not forget previously learnt models of the system. The third issue is when *multiple* rational agents drive an autonomous system by mutual interaction [15]. The agents are rational and take individual decisions, but each agent's decision affects the other agents. Thus, stabilizing the autonomous system and efficiently controlling it requires that these agents co-operate with each other in the absence of perfect information. This stabilization is hard to attain when incomplete information is known.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2 BACKGROUND AND PRIOR WORK

Autonomous systems controlled by intelligent agents involve sequential decision making under uncertainty. The natural mathematical framework to model these autonomous systems is a Markov decision process (MDP) [14]. A MDP is characterized by the tuple $\langle S, A, P, R\rangle$, where S is the set of operating states of the system, A is the set of controls/actions available to the agent, $P : S \times A \times S \to [0, 1]$ is the transition probability function and $R : S \times A \times S \to \mathbb{R}$ is the reward function. If the analytical form of the transition probability and reward functions are known, then dynamic programming techniques [14] can be used by the agent to control the autonomous system in an efficient or optimal manner. However, more often autonomous systems are complex to model and generally the agent does not have access to the form of these functions. In this case, the agent can *learn* to control the autonomous system using reinforcement learning (RL) techniques [14]. Then, we say that the agent is a "RL agent". Suppose the initial state of the autonomous system is s_0. The RL agent deterministically chooses an action a_t according to a policy $\pi(s_t)$ and the system evolves to the next state s_{t+1} based on the conditional distribution $p(\cdot|s_t, a_t)$. Simultaneously the RL agent receives a reward $r_t = R(s_t, a_t, s_{t+1})$. This interaction between the RL agent and the autonomous system gives us a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, \ldots)$. The goal of the RL agent is to find a policy π^* that maximizes the expected sum of discounted rewards, i.e. maximizes $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma \in [0, 1)$ is the discount factor and actions are picked according to the policy π^*. This is a long-term performance criterion.

The issues highlighted in Section 1 should be viewed from the lens of the RL paradigm and MDP modeling, because ultimately, any issues which arise in the system need to be tackled at the mathematical model level. Every RL agent has access only to a finite collection of trajectories $\{\tau^i = (s_0^i, a_0^i, r_0^i, s_1^i, a_1^i, r_1^i, s_2^i, \ldots, s_N^i, a_N^i, r_N^i) : i = M < \infty, N < \infty\}$ and it must find an appropriate policy which maximizes the above long-term discounted rewards criterion. A learning technique is efficient if it utilizes minimal amount of trajectory information to learn a good policy. Also, the data obtained can be utilized to infer more about the system conditions as well as the system dynamics. Prior work [2, 4] have explored data efficiency for single-agent settings as well as multi-agent settings but lack theoretical justification. These works are empirical and lack any justification of the results.

Continual learning capability is needed when the transition probability and reward functions P and R vary with time. The system dynamics at every decision instant constitutes an environment *context*, and the context variation is hidden from the agent. The different hidden contexts are formalized as MDPs $M_{\theta_0}, M_{\theta_1}, \ldots, M_{\theta_n}$, where $M_{\theta_i} = \langle S, A, P_{\theta_i}, R_{\theta_i} \rangle,\ 0 \le i \le n$. At time t, if agent takes action a_t in state $s_t \sim p_{\theta_{t+1}}(\cdot|s_t, a_t)$. The RL agent is unaware of the underlying dynamics and the policy it learns using the trajectory τ is not optimal when environment context changes. Some prior works [1, 5] focus on continual learning for different environment contexts. [1] proposes a model-based RL algorithm which estimates transition probability and reward functions from trajectories corresponding to all environment contexts. This algorithm is highly data-inefficient and heuristic. [5] proposes cascade neural networks for recording policies at multiple timescales. This architecture does not deal with environment contexts, but only remembers some previously learnt policies. Additionally, in cases where previous context is again active, this method re-learns the policy and hence is prone to *catastrophic forgetting* [6].

Multi-agent RL [15] techniques are designed for sequential-decision making problems with more than one RL agent involved. The evolution of the autonomous system state and the reward received by each agent are influenced by the joint actions of all agents. Further, each agent has its own long-term reward to optimize, which becomes a function of the policies of all other agents. Prior works [10] have achieved empirical success, but lack theoretical analysis [3]. Scalability issues also present difficult challenges.

3 CONTRIBUTIONS

In my thesis, I focus on developing algorithms which provide practical solutions to the issues described in Sections 1 and 2.

Continual Learning: My work [11] proposes a continual RL algorithm known as *Context Q-learning* (abbrv. CQL). A RL agent employs CQL to search and learn an approximately optimal non-stationary policy and this search over the space of non-stationary policies is extremely efficient. This policy maximizes the long-term performance criterion (see Section 2) when contexts vary and the information about environment contexts is not available. Our algorithm collects experience tuples, where each tuple consists of the current and the next states as well as the reward. These tuples are analyzed to detect changes in the model using a novel changepoint algorithm. Additionally the algorithm is designed so that it resists catastrophic forgetting.

Efficiency in Learning: Efficient learning techniques need to use the trajectory information for multiple purposes. This is even more important in *partially observable* environments, where the state s_t is not observable. Instead, the RL agent has access to a measurement o_t of the state s_t and $o_t \sim O(s_t, a_{t-1})$. Here $O(s_t, a_{t-1})$ is the probability distribution over observations when state is s_t and previous action is a_{t-1}. Unlike the perfect trajectory information, defined in Section 2, in case of partial observability, an RL agent has access to the (modified) trajectory information $\{o_0, a_0, r_o, o_1, a_1, r_1, \ldots, o_N, a_N, r_N\}$, using which it has to control the autonomous system. My work [13] uses the trajectory information obtained in a partially observable environment for multiple purposes. In [13], I focus on enabling a UAV quadrotor equipped with a monocular camera, to autonomously avoid collisions with obstacles in unstructured and unknown indoor environments. The monocular camera images are generally used for surveillance applications. These images are the observations available to the agent and can be utilized for obstacle avoidance as well, as proposed in my work [13]. Our proposed method does not require any additional equipment like stereo camera, optical flow camera etc. to be mounted on a UAV. The obstacle avoidance UAV RL agent utilizes multiple relevant images from a monocular camera to build a map of the surrounding physical environment. This map is leveraged to avoid stationary as well as mobile obstacles that are of various shapes and sizes. Compared to prior approaches which do not fully retain and utilize the extensively available information from the monocular camera, our obstacle avoidance method uses the temporal information inferred from a sequence of images to provide a direction in which the UAV needs to move to avoid obstacles. The method modifies the DQN [9] algorithm using LSTM neural network architecture and temporal attention.

4 FUTURE WORK

My work till now has addressed some issues which arise when RL agents control autonomous systems. However, there is scope for more improvements. Further, I aim to focus on tackling the issues of data efficiency and learning in presence of multiple agents. More specific details are as follows:

Efficiency in Learning: My approach in [13] builds a data efficient deep Q network [9]. However, one can also consider data efficiency in policy-based approaches as well. To enhance usability, I would like to extend my work to actor-critic algorithms. Another direction I intend to explore is to make off-policy prediction [14] and control algorithms more data efficient. Additionally, I intend to leverage compressed sensing [16] to ascertain relevance of trajectory samples.

Multiple Agents: I intend to explore theoretical analysis for multi-agent RL algorithms, using off-policy based approaches. Off-policy approaches as highlighted earlier, promise data efficiency, since trajectories of one policy is used to learn about other policies as well as about optimal policies. Though some prior works [10] exist, there is ample scope for more exploration.

REFERENCES

[1] Bruno C. da Silva et al. 2006. Dealing with Non-stationary Environments Using Context Detection. In *Proceedings of the 23rd International Conference on Machine Learning*. 217–224.

[2] Marc Peter Deisenroth and Carl Edward Rasmussen. 2011. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11)*. Omnipress, Madison, WI, USA, 465–472.

[3] Foerster et al. 2018. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[4] Shixiang Gu et al. 2016. Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic. (2016). arXiv:1611.02247

[5] Christos Kaplanis et al. 2019. Policy Consolidation for Continual Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 3242–3251.

[6] Ronald Kemker et al. 2018. Measuring catastrophic forgetting in neural networks. In *Thirty-second AAAI conference on artificial intelligence*.

[7] Prabuchandran K.J, Hemanth Kumar, and Shalabh Bhatnagar. 2014. Multi-agent Reinforcement Learning for Traffic Signal Control. In *2014 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2529–2534.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*. 1097–1105.

[9] V Mnih et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[10] Afshin OroojlooyJadid and Davood Hajinezhad. 2019. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963* (2019).

[11] Sindhu Padakandla, Prabuchandran K J, and Shalabh Bhatnagar. 2019. Reinforcement Learning in Non-Stationary Environments. *CoRR* (2019). http://arxiv.org/abs/1905.03970

[12] A. Singla et al. 2019. Realizing Learned Quadruped Locomotion Behaviors through Kinematic Motion Primitives. In *2019 International Conference on Robotics and Automation (ICRA)*. 7434–7440. https://doi.org/10.1109/ICRA.2019.8794179

[13] Abhik Singla, Sindhu Padakandla, and S. Bhatnagar. 2019. Memory-Based Deep Reinforcement Learning for Obstacle Avoidance in UAV With Limited Environment Knowledge. *IEEE Transactions on Intelligent Transportation Systems* (2019). https://doi.org/10.1109/TITS.2019.2954952

[14] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press, Cambridge, MA, USA.

[15] Ardi Tampuu et al. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE* 12, 4 (04 2017), 1–15. https://doi.org/10.1371/journal.pone.0172395

[16] Borkar V.S et al. 2018. Gradient Estimation with Simultaneous Perturbation and Compressive Sensing. *Journal of Machine Learning Research* 18, 161 (2018), 1–27.