Theoretical Reinforcement Learning Notes

Liangyawei Kuang

11st Feb 2022

Please email me at kriskongloveyou@gmail.com if you find any typos or errors. I do appreciate it!

Hong Kong University of Science and Technology

Theoretical Reinforcement Learning Notes

Liangyawei Kuang

Abstract

A kind reminder: This notes are mainly for reinforcement learning (RL) researchers who want to have a deep understanding in some basic RL concepts and related topics from a theoretical perspective. For those novices in this area, I recommend you read Csaba Szepesvári's RL book (Szepesvári, 2010) (about 100 pages) or the most famous RL book (Sutton & Barto, 2018) (about 500 pages) by Richard S. Sutton and Andrew G. Barto before you read my notes. Of course, you can also take my note as your first RL book and take the two books I mentioned above as references if you have a strong math background.

Updating...

To my mum Jing Liang and my dad Zisheng Kuang with my forever love!

Acknowledgements

First and foremost, I am greatly thankful to my advisor: Fangzhen Lin, who was willing to be my supervisor, and also gave me the freedom to pursue my own research ideas on Multi-Agent Systems and Reinforcement Learning.

I am so grateful to the Guangzhou Government, who stuck to support the development of the Hong Kong University of Science and Technology (Guangzhou) Campus and our research teams, even during a financially difficult time and a global pandemic caused by COVID-19.

I would like to thank my interview committee members: Ming Liu and Michael Yu Wang, who decided to recruit me as a post-graduate researcher at Hong Kong University of Science and Technology.

Thanks to Miss. Xuetong Wang as she donated her monitor to me so that I don't have to be worried about health issues of my eyes and back.

Thanks to Miss. Xiaomeng Chen as she donated her Reinforcement Learning book (Sutton and Barto) to me. I read that book more than five times, which makes me fall in love with this great research area.

Thanks to Miss. Huiwen Yang as she gave me a lot of academic support so that I could have a deep understanding of some pure math topics.

Thanks to Miss. Wei Huo as she sent me tons of course slides so that I could follow so many courses as I want at the same time.

Thanks to Mr. Nachuan Yang as he let me borrow his math books about control and systems, which inspired me the importance of theoretical work during research and shocked me with the beauty of pure math.

Thanks to Mr. Dongwei Xiao as he talked with me many times about research and many other broad topics, and also gave me a lot of covid test boxes during the pandemic. I hope both of us could be great professors who will not only do a great job at research but also do care about students with a warm heart.

Thanks to the Computer Science and Engineering (CSE) department for offering me a quiet working place with a desk, a chair, and a monitor in the CSE Research Postgraduate Hub, so that I can concentrate on my research for days and nights.

Last but not least, I would like to thank my parents, Jing Liang and Zisheng Kuang, who provided me with their endless love and supported me all the time. I am forever grateful to them!

Updating...

Contents

List of Figures

List of Tables

Chapter 1

Reinforcement Learning Fundamentals

Markov Decision Process.
For a reinforcement leaning question, we can formulate it as a infinite-horizon discounted Markov Decision Process (MDP). An MDP is defined by a quintuiple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the set of states; $\mathcal{A}$ is the set of actions; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the set of possibility from a state $s \in \mathcal{S}$ to a state $s' \in \mathcal{S}$, given a action $a \in \mathcal{A}$; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the immediate reward function for agents tranfer from (s, a) to s'; $\gamma \in [0, 1)$ is the discount factor.
At time t, the agent in state s_t executes an action a_t by following the policy $\pi : \pi(a|s)$, which is a mapping from states $\mathcal{S}$ to actions $\mathcal{A}$. The system transit from state s_t to the next state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. For MDPs, the goal is to find the optimal policy π to maximize $a_t \sim \pi(\cdot|s_t)$ and the accumulated rewards

$$\mathbb{E}\left[\sum_{t\geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0\right].$$

Accordingly, given policy π, for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we could define the *action-value function* (the Q-function), which is starting from $(s_0, a_0) = (s, a)$, as

$$Q_\pi(s, a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), a_0 = a, s_0 = s\right],$$

and the *state-value function* (the V-function), starting from $s_0 = s$, as

$$V_\pi(s, a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \middle| a_t \sim \pi(\cdot|s_t), s_0 = s\right]$$

π^* are referred to the optimal policy as the optimal Q-function and V-function respextively. By virtue if the Markov property, the optimal function could be obtained by iteration based on dynamic programming (DP), which is usually required of the complete knowledge of the model.

Value-Based Methods.
The value-based methods are mainly to find the estimate of the optimal Q-function Q^*_π. One famous value-based algorithm is Monte-Carlo tree search (MCTS), which is used under the imcomplete environment knowledge. In this method, a monte carlo simulation

is executed based on a search tree to estimate the optimal value function.

Temporal-Difference (TD) learning method is a combination of MCTS and DP. It learns the estimates partially based on estimates, which is known as *bootstrapping*. As model free methods, TD methods are implemented more naturally than MCTS and DP with a online and fully incremental way.

Q-learning is one of the most important value-based method, which is actuall an off-policy TD control. The optimal policy can be approximated by taking the greedy action of estimation of the Q-value function $\hat{Q}(s,a)$. The Q-function is updated according to

$$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a) + \alpha \left[r + \gamma \max_{a'} \hat{Q}(s',a') - \hat{Q}(s,a) \right]$$

with the loss funciton

$$\mathcal{L}(s,a,r,s') = (r + \gamma \max_{a'} Q(s',a') - Q(s,a))^2.$$

Policy-Based Methods.

The main idea in policy-based method is to update the parameter by following the gradient direction, which is known as policy gradient (PG). The closed-form of PG is given as

$$\nabla J(\theta) = \mathbb{E}_{a \sim \pi_\theta(\cdot|s), s \sim \eta_{\pi_\theta}(\cdot)} \left[\mathcal{Q}_{\pi_\theta}(s,a) \nabla log \pi_\theta(a|s) \right],$$

where $J(\theta)$ and $\mathcal{Q}_\pi$ are the expected reward and Q-function with following policy π_θ, respectively, $\pi_\theta(\cdot|s)$ is the approximation of $\pi(\cdot|s)$, η_{π_θ} is the measurement of state occpancy, and $\nabla log \pi_\theta(a|s)$ is the score of the policy.

Compared with value based methods, policy based one are more powerful with better convergence guarantees with neural networks for function approximation, which is a fashion today with the rise of Deep Learning (DL). And policy-based method are believed to have the ability to handle bigger discrete or even continuous state-action spaces.

Updating...

$$\frac{a}{b} \tag{1.1}$$

Theorem 1 *Let f be a function whose derivative exists in every point, then f is a continuous function.*

Chapter 2

Action-Value Estimation

Updating...

Chapter 3

Policy Gradient

Updating...

Chapter 4

Multi-Agent Reinforcement Learning

Markov Games.
Markov games (MGs), which is also known as stochastic games, is originally a framework for MDP in multi-agent settings. A markov game is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, \cdots, N\}$ denotes the set of N agents, $\mathcal{S}$ denotes the globally observed state space by the whole system, $\mathcal{A}^i$ denotes the action space of agent i, $\mathcal{P} : \mathcal{S} \times \{\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^N\} \to \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ denotes the possibility distribution for mapping from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ via any joint action $a \in \mathcal{A}$, $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the immediate reward received by agent i via a transition from (s, a) to s', $\gamma \in [0, 1)$ denotes the discount factor.
From time t to time $t+1$, agent $i \in \mathcal{N}$ executes action a_t^i, the system will transites from s_t to s_{t+1}, and all agents get immediate reward by $R^i(s_t, a_t, s_{t+1})$. For every individual agent i, the goal is to maximize its own reward in a long finite horizon or infinite horizon by finding the optimal policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ so that $a_t^i \sim \pi^i(\cdot|s_t)$. The joint policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is $\pi(a|s) := \prod_{i\in\mathcal{N}} \pi^i(a^i|s)$. For any state $s \in \mathcal{S}$ and joint policy π,

$$V^i_{\pi^i,\pi^{-i}}(s) := \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R^i(s_t, a_t, s_{t+1}) \middle| a_t^i \sim \pi^i(\cdot|s_t), s_0 = s\right],$$

where $-i$ denotes the indices of all other agents in $\mathcal{N}$ except agent i. A nash equilibrium (NE) of a *markov game* $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \gamma)$ is a joint policy $\pi^* = (\pi^{1,*}, \cdots, \pi^{N,*})$ so that for any $s \in \mathcal{S}$, $i \in \mathcal{N}$, and π^*

$$V^i_{\pi^{i,*},\pi^{-i,*}}(s) \geq V^i_{\pi^i,\pi^{-i,*}}(s)$$

The nash quilibrium point π^* is a fixed point so that all agent won't transite to a better point as there is not any incentive to do so. For any agent $i \in \mathcal{N}$, $\pi^{i,*}$ is the best response to $\pi^{-i,*}$. For MARL settings, finding the NE is a standard learning goal and NE always exists for finite-space infinite-horizon discounted MGs (Filar & Vrieze, 2012).

Cooperative Settings.
In this short paper, we only consider cooperative settings which mean all the agents collaborate with each other to achieve shared goal. In a fully cooperative setting, all agents share one reward function $\mathcal{R}^1 = \mathcal{R}^2 = \cdots = \mathcal{R}^N = \mathcal{R}$. With this model, the Q-function is identical to all agents so that Q-learning updates could be applied with taking the max over the joint action space $a' \in \mathcal{A}$. In a more general cooperative setting, agents have their own reward function and the goal is to optimize the long-term reward for all agents. One common reward model is *team-averager* reward $\bar{R}_(s, a, s') := N^{-1} \cdot \sum_{i\in\mathcal{N}} R^i(s, a, s')$

for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$.

Partial Observability.
In multi-agent settings, we can not ignore the influence from the real environment, the noise and limited sensors may prevent the agent from abserving the state of the environment (Oliehoek & Amato, 2016). However, MGs can only handle the fully observed environment. In this case, *Partially Observable Markov Decision Process* (POMDP) is more suitable to represent such state uncertainty by incorporating observations and their probability of occurrence conditional on the state of the environment (Kaelbling, Littman, & Cassandra, 1998). More generally speaking, this partially observed setting can be modeled by a decentralized POMDP (Dec-POMDP), which shares most of the elements, including the reward function and the trasition model. Based on MGs, a Dec-POMDP is formally defined by the tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i \in \mathcal{N}}, \mathcal{Z}, \mathcal{O}, \gamma)$, where $\mathcal{Z} := \{\mathcal{Z}^1 \times \cdots \times \mathcal{Z}^N\}$ denotes the joint abservations, $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ denotes the observation probabilities, the others denotation remain the same as MGs. Under this setting. At time t, agent i has its *action-observation* history $\mathcal{H}_i := [\mathcal{O}_{i,1}, \mathcal{A}_{i,1}, \cdots \mathcal{O}_{i,t-1}, \mathcal{A}_{i,t-1}]$, $\mathcal{H}_i \in \mathcal{H}$, and a stochastic policy for agent i is $\pi^i(\mathcal{A}^i | \mathcal{H}^i)$. Given the history $\mathcal{H}$, the goal for each agent is to maximize its expected discounted rewards in a long term.

Updating...

Appendix A

Appendix Title

Updating...

References

Filar, J., & Vrieze, K. (2012). *Competitive markov decision processes.* Springer Science & Business Media.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, *101*(1-2), 99–134.

Oliehoek, F. A., & Amato, C. (2016). *A concise introduction to decentralized pomdps.* Springer.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, *4*(1), 1–103.