DDA4230 Reinforcement learning Markov decision processes

Lecture 2

Lecturer: Baoxiang Wang *Scribe: Baoxiang Wang*

1 Goal of this lecture

In this lecture we will formulate Markov decision processes and introduce different variants of reinforcement learning methods.
Suggested reading: Chapter 1 and 3 of *Reinforcement learning: An introduction.*

2 Reinforcement learning

In Reinforcement Learning (RL), we consider the problem of learning to act through trial and error. In general, we do not assume an explicit teacher or an explicit knowledge of the world model. A reinforcement learning agent interacts with its world and from that learns how to maximize some cumulative reward over time.

Reinforcement learning has become increasingly more popular over recent years with some milestone results. Examples are, Deep Q-Networks and Atari games, series on AlphaGo, AlphaStar, OpenAI Five for Dota 2, DeepStack and Libratus/Pluribus for poker, and AlphaFold. Many other areas borrow concepts and algorithms from RL as well.

2.1 Connection with other learning problems

In supervised learning we are given a dataset, which consists of examples and labels. In the training set, for each sample, we are provided the correct prediction (label in classification problems or correct output in regression problems). In contrast, when no labels are provided in a dataset, unsupervised learning refers to methods that find underlying, latent structure in the data, when no label is given.

However, in reinforcement learning, we are dealing with making decisions and comparing actions that could be taken, rather than making predictions. A Reinforcement Learning agent may interact with the world, and receive some immediate, partial feedback signal — commonly called a reward — for each interaction. However, the agent is given little indication if the action it took was actually the best it could have chosen, and the agent must somehow learn to pick actions that will maximize a long term cumulative reward. Therefore, because of the weak/incomplete feedback provided by the reward signal we could consider Reinforcement Learning to lie somewhere between Supervised Learning, which gives strong feedback with labeled data, and Unsupervised Learning, with no feedback or labels.

2.2 Challenges in reinforcement learning

Reinforcement learning introduces a number of challenges that we need to overcome, and potentially make trade-offs between. The agent must be able to optimize its actions to maximize the reward signal it receives. However, as the agent needs to learn by interacting with its environment, exploration is required. This leads to a natural trade-off between exploration and exploitation, where the agent needs to decide between potentially finding new, better strategies at the risk of receiving a lower reward, or, if it should exploit what it already knows. Another question we face is, can the agent generalize its experience? That is, can it learn whether some actions are good/bad in previously unseen states? And finally, we also need to consider delayed consequences of the agents' actions, that is, if it receives a high reward, was it because of an action it just took, or because of an action taken much earlier?

3 Markov decision processes

We consider the discrete-time Markov decision process (MDP) setting, denoted as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho_0, \gamma)$.

- $\mathcal{S}$ the state space;
- $\mathcal{A}$ the action space;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ the environment transition probability function;
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ the reward function;
- $\rho_0 \in \Delta(\mathcal{S})$ the initial state distribution;
- $\gamma \in [0, 1]$ the unnormalized discount factor.

Note that $\Delta(\mathcal{X})$ denotes the set of all distributions over set $\mathcal{X}$.

A stationary MDP follows for $t = 0, 1, \ldots$ as below, starting with $s_0 \sim \rho_0$.

- The agent observes the current status s_t;
- The agent chooses an action a_t;
- The agent receives the reward $r_t \sim \mathcal{R}(s_t, a_t)$;
- The environment transitions to a subsequent state according to the Markovian dynamics $s_{t+1} \sim \mathcal{T}(s_t, a_t)$.

This process generates the sequence $s_0, a_0, r_0, s_1, \ldots$ indefinitely. The sequence up to time t is defined as the trajectory indexed by t, as $\tau_t = (s_0, a_0, r_0, s_1, \ldots, r_t)$.

The full Markov chain amounts to the dynamics of the second bullet (that the agent chooses an action a_t). We will consider the problem of making a sequence of good decisions. By the formulation of stationary MDPs, the optimal choice of action a_t need only to depend on the state s_t. We call this mapping from $\mathcal{S}$ to $\mathcal{A}$ the policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ of the agent.

We therefore denote $a_t = \pi(s_t) \in \Delta(S)$. As a notational convention, when the policy is stochastic we write $a_t \sim \pi(a_t \mid s_t)$. As there exists at least one deterministic policy to be optimal, it is an alternative formulation to restrict the policy $\pi : \mathcal{S} \to \mathcal{A}$ to be deterministic.

3.1 Important variables in MDPs

The return is defined as the discounted cumulative reward as a random variable

$$R_t = \sum_{t=0}^{\infty} \gamma^t r_t.$$

The expectation of the return is the objective to be maximized by the agent

$$J = \mathbb{E}_{s_t, a_t, r_t, t \geq 0}[\sum_{t=0}^{\infty} \gamma^t r_t].$$

Define the action-state value function of a given policy π

$$Q^\pi(s, a) = \mathbb{E}_{s_t, a_t, r_t, t \geq 0}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$$

to be the expected return of policy π at state s after taking action a. Also define the state-value function

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(s)}[Q^\pi(s, a)]$$

as the expected return given the initial state only, and the advantage function

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

as the difference between the action-value function and the state-value function. When the context is clear we omit the superscript π and write $Q(s, a)$, $V(s, a)$, and $A(s, a)$. Based on the value functions, define the temporal-difference error

$$\delta_t = r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t).$$

By its definition, this error should be zero if π is one of the optimal policies.

4 Taxonomy of reinforcement learning settings

4.1 Stationarity of MDPs and agents

An above-defined MDP is stationary, where the Markovian dynamics of s_{t+1} depends only on s_t and a_t, as $s_{t+1} \sim \mathcal{T}(s_t, a_t)$. An MDP is non-stationary if this transition also depends on the time t. A policy is stationary if the action depends only on the state. A policy is Markovian but non-stationary if the action also depends on the time t.

The above-defined MDPs are stationary MDPs where at least one of the optimal policies is also stationary. Therefore, it is safe to restrict the policy to the set of stationary Markovian policies, as we did above. However, if the process is not indefinite and stops at some fixed horizon T, then there does not necessarily exist a stationary Markovian policy to be one of the optimal policies.

Note that some MDPs have a set $\mathcal{S}_T$ of terminal states, where the process $t = 0, 1, \ldots$ stops $s_t \in \mathcal{S}_T$. Despite that the process is no longer indefinite, there still exists at least one stationary Markovian policy to be optimal. Also note that many papers will restrict the policy to be stationary despite having a horizon T.

4.2 State and action spaces

The state space and the action space can be arbitrary sets, but two common settings are used

- $\mathcal{S} \in \mathbb{R}^n$ the n dimensional state space, $\mathcal{A} \in \mathbb{R}^m$ the m dimensional action space;
- $\mathcal{S} \in [n]$ the size-n discrete state space, $\mathcal{A} \in \mathbb{R}^m$ the size-m discrete action space;

Note that $[x] = \{0, 1, \ldots, x-1\}$ for an integer x. The study on continuous and discrete RL settings can be quite different.

4.3 Stochasticity of MDPs and agents

The stochasticity of a Markov chain given the MDP and the policy can come from three components: stochastic Markovian dynamics, stochastic rewards, and stochastic policies.

In general, any component being stochastic will make the Markov chain stochastic. We therefore need to optimize the objectives who are in forms of expectations and use samples drawn from unknown distributions. Stochasticity leads to important topics in RL, such as the overestimation problem.

4.4 Discount of rewards

The discount factor $\gamma \in [0, 1]$ balances the short-term and long-term rewards. With the discounted objective

$$R_t = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots,$$

with larger γ the agents favors the long-term goal and with smaller γ the agents favors the immediate rewards in some near future. Two extreme cases are $\gamma = 0$ and $\gamma = 1$, where the former corresponds to $R_t = r_0$ as an one-step MDP and the latter corresponds to $R_t = r_0 + r_1 + r_2 + \ldots$.

Note that the original, unbiased objective of an MDP should be $R_t = r_0 + r_1 + r_2 + \ldots$ under $\gamma = 1$. This objective, however, is hard to optimize in many settings. A line of research focuses on choosing the right γ and recovers the original objective.

4.5 Agents of RL

We can classify our agents in a number of ways, as can be seen in Table 1, and each type of agent isn't necessarily unique. For example, an actor critic agent could also be a model free agent. An overview of ways that we can classify agents can also be seen in Figure 1.

4.6 More on Markov structure

The classification of the Markov structure is made by looking at (i) whether a agent can observe the state of a world and (ii) whether the agent can influence the state transition. An overview of different Markov Structures is presented in Figure 2.

Agent type	Policy	Value Function	Model
Value Based	Implicit	✓	?
Policy Based	✓	X	?
Actor Critic	✓	✓	?
Model Based	?	?	✓
Model Free	?	?	X

Table 1: An overview of properties of different types of reinforcement learning agent. Where a check-mark indicates that the agent has the component, a cross indicates that it must not have the component, and a question mark indicates that the agent may have that component, but isn't required to have it.

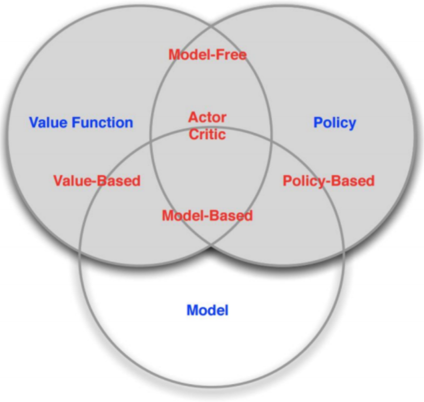
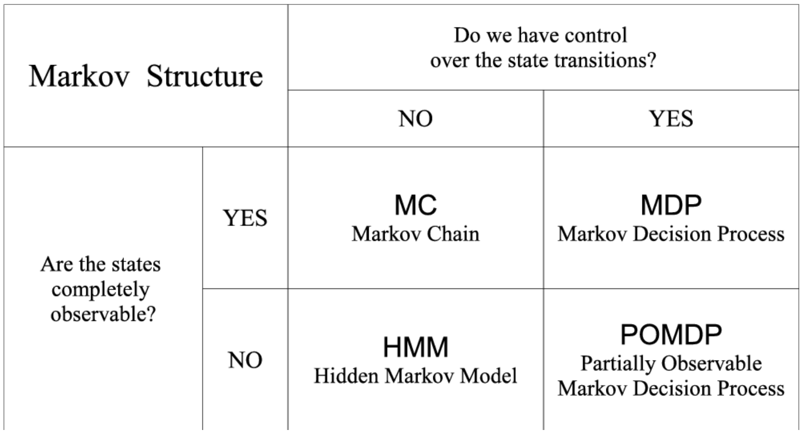


Figure 1: Classification of different reinforcement learning agents.

Acknowledgement

This lecture notes partially use material from *Reinforcement learning: An introduction* and *CS234: Reinforcement Learning* from Stanford. Figure 1 is from David Silver's course *COMPGI13: Reinforcement Learning* at UCL. Figure 2 is from Seong-Hoon Kim. Proofread by Shaokui Wei.



Markov Structure		Do we have control over the state transitions?	
		NO	YES
Are the states completely observable?	YES	MC Markov Chain	MDP Markov Decision Process
	NO	HMM Hidden Markov Model	POMDP Partially Observable Markov Decision Process

Figure 2: Classification of different Markov Structures.