| **DDA4230 Reinforcement learning** | Markov decision process |
|---|---|
| Lecture 1 | |
| *Lecturer: Guiliang Liu* | *Scribe: Baoxiang Wang* |

# 1 Goal of this lecture

In this lecture we will formulate Markov decision process 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-learning 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 $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma)$.

- $\mathcal{S}$ the state space;
- $\mathcal{A}$ the action space. $\mathcal{A}$ can depend on the state $s$ for $s \in \mathcal{S}$;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ 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 state $s_t$;
- 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 $s_0, a_0, r_0, s_1, \ldots$, until when $s_T$ is a terminal state, or 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

$$R_t = \sum_{t' \geq t}^{\infty} \gamma^{t'} r_{t'} \,.$$

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

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

Define the action value function of a given policy $\pi$

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

to be the expected return of policy $\pi$ at state $s$ after taking action $a$. Also define the state value function

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

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 $\pi$ and write $Q(s, a)$, $V(s, a)$, and $A(s, a)$. Based on the value functions, define the temporal-difference error

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \,.$$

These variables are not guaranteed to exist. For example, when $r_t = 1$ and $\gamma = 1$, $R_t$ does not exist. It is also possible that $R_t$ exists as a random variable but its expectation does not exist (therefore $J$ does not exist). Under several sufficient conditions, these variables will exist. For example, when the reward function $\mathcal{R}$ is bounded and $\gamma < 1$, these variables exist. When $\mathcal{R}$ is nonzero only at terminal states the variables will also exist. When the context is clear, throughout this series of lecture notes, by writing any of these variables we assume their existence.

## 3.2 The Bellman equation

The definition of the state value function insists that

$$V(s_t) = \mathbb{E} \left[ r_t + \gamma V(s_{t+1}) \right]$$

when $s_t$ is not a terminal state. When $s_t$ is instead a terminal state, we have

$$V(s_t) = \mathbb{E} \left[ r_t \right] \,.$$

This recursive property is called the Bellman equation, named after Richard E. Bellman. The equation can be alternatively denoted by the action value function as

$$Q(s_t, a_t) = \mathbb{E} \left[ r_t + \gamma Q(s_{t+1}, a) \mid a \sim \pi(a \mid s_{t+1}) \right] \quad \text{and} \quad Q(s_t, a_t) = \mathbb{E} \left[ r_t \right]$$

for non-terminal and terminal states, respectively.

# 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 $s_{t+1} \sim \mathcal{T}(s_t, a_t)$ and the reward $r_t$ depends only on $s_t$ and $a_t$ as $r_t \sim \mathcal{R}(s_t, a_t)$. An MDP is non-stationary if any of the transition dynamics and the reward also depend 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 and 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, $\mathcal{A} \in [m]$ the size-$m$ discrete action space.

Note that $[x] = \{1, 2, \ldots, x\}$ 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. When the objective is discounted ($\gamma < 1$)

$$R_0 = r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots ,$$

with larger $\gamma$ the agents favor the long-term goal and with smaller $\gamma$ the agents favor the immediate rewards in some near future. Two extreme cases are $\gamma = 0$ and $\gamma = 1$,

where the former corresponds to $R_0 = r_0$ as a one-step MDP and the latter corresponds to $R_0 = r_0 + r_1 + r_2 + \dots$.

Note that the original, unbiased objective of an MDP should be $R_0 = r_0 + r_1 + r_2 + \dots$ under $\gamma = 1$. This objective, however, is hard to optimize in many settings. A line of research focuses on choosing the right $\gamma$ 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.

| Agent type | Policy | Value Function | Model |
|---|---|---|---|
| Value-based | Implicit | ✓ | ? |
| Policy-based | ✓ | ✗ | ? |
| Actor-critic | ✓ | ✓ | ? |
| Model-based | ? | ? | ✓ |
| Model-free | ? | ? | ✗ |

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 is not required to have it.

In the same formulation of sequential decision making, it is possible that the agent knows the world model, including the state transition function and the reward function. In this case we typically categorizes it into the area of control theory. Though, some results in this setting can be useful in analyses of certain RL problems. When the agent does not know the world model, the agent is categorized into model-based agents if it makes some attempt to try to obtain an estimation of the world model. On the contrary, the model is model-free if it learns the task purely through trial and error.

If the agent focuses on learning a optimal policy function then it is policy-based. If the agent focuses on learning an optimal value function then it is value-based. Each type of agent is not necessarily unique. For example, actor-critic methods is both policy-based and value-based. 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. This course focuses on model-free agents (with some presence of model-based methods), which can be policy-based, value-based, or actor-critic-based.

### 4.6 Classification of Markov structures

Apart from the canonical Markov decision processes we introduced in this lecture notes, several other Markov settings can be relevant. The classification of Markov structures is made by looking at whether an agent can observe the state of a world and whether the agent can influence the state transition. An overview of different Markov structures is presented in Table 2.

For fully observable states with state transitions that are independent of the policy, the problem constitute a Markov process (also known as a Markov chain), which is heavily

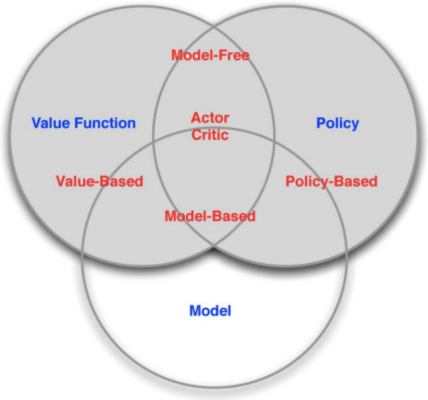


Figure 1: Classification of different reinforcement learning agents.

studied in theory for its various properties. The setting is the same as when the policy is fixed, which is known as the policy evaluation problem. If the state is instead not fully observable, the problem is known as hidden Markov models (HMM), which is heavily studied in machine learning. The inference of the Markov structure in HMM is a classical statistical machine learning approach to investigate sequential data. When in general actions have influence over the state transitions, these two models generate to MDPs and partially observable Markov decision processes (POMDPs).

| Markov structure | | Do actions have influence over the state transitions? | |
|---|---|---|---|
| | | NO | YES |
| Are the states fully observable? | YES | Markov process (Markov chain) | Markov decision process |
| | NO | Hidden Markov model | Partially observable Markov decision process |

Table 2: Classification of different Markov structures.

## 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. Table 2 is drawn by Shaokui Wei. Proofread by Shaokui Wei.