1 Lecture two: Markov Decision Process (MDP)

1.1 Markov Process

The current state characterises the process -¿ we are told the state −¿ environment is fully observable

Almost all RL problems can be formalised as MDPs
−¿ Optimal control primarily deals with continious MDPs
−¿ Any partially observable problems can be converted into MDPs
−¿ Bandits are MDPs with one state

"The future is independent of the past given the present"

$$\mathbb{P}\left[S_{t+1} \mid S_t\right] = \mathbb{P}\left[S_{t+1} \mid S_1, \ldots, S_t\right] \tag{1}$$

What happens next only depends on what happend on the state before - you can throw away anything else.

For a Markov state s and successor state s', the *state transition probability* is defined as

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right] \tag{2}$$

State transition matrix ρ defines transition probabilites from all states s to all successor states s'.

$$\mathcal{P} = from \begin{pmatrix} \mathcal{P}_{11} & \ldots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \ldots & \mathcal{P}_{nn} \end{pmatrix} \tag{3}$$

Each row of the matrix sums to 1!
A Markov process is a memoryless random process, i.e. a sequence of random states, S_1, S_2, ... with the Markov propoerty. It is defined as a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$.

- S is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix

1.2 Markov Reward process (MRP)

The Markov Reward Process is defined as a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$.

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$

The *return* $\mathcal{G}_t$ is the total discounted reward from time-step t.

$$\mathcal{G}_t \;=\; R_{t+1} \;+\; \gamma R_{t+2} \;+\; \ldots \;=\; \sum_{k=0}^{\infty} \gamma^k \, R_{t+k+1} \tag{4}$$

- The discount γ is the present value of future rewards - the closer γ is to zero, the less are later rewards accounted (e.g. more 'short-sighted').
- The value of receiving reward $\mathcal{R}$ after $k+1$ time-steps is $\gamma^k R$.

Why discount?

- Unless you really trust your model and believe that everything turns out as planned, you need to discount in deviations - Uncertainty about the future may not be fully represented
- Mathematically convenient to discount rewards, avoids infinite returns
- If the reward is financial, immediate rewards may earn more interest than delayed rewards
- Animal/human behaviour shows preference for immediate reward
- It is sometimes possible to use *undiscounted* Markov reward process (i.e. $\gamma = 1$), e.g. if all sequences terminate.

The value function

The value function *v(s)* gives the long-term value of state s. It defines the expected return in a MRP starting from state s:

$$v(s) \;=\; \mathbb{E}\left[G_t \mid S_t \;=\; s\right] \tag{5}$$

The value function can be decomposed into two parts:

- immediate reward $R+1$
- discounted value of successor state $\gamma v(S_{t+1})$

This resolves into the **Bellman equation for MRPs**:

$$v(s) \;=\; \mathbb{E}\left[G_t \mid S_t \;=\; s\right] \;=\; \mathbb{E}\left[R_{t+1} \;+\; \gamma v(S_{t+1}) \mid S_t \;=\; s\right] \tag{6}$$

By averaging all possible outcomes we get

$$v(s) \;=\; \mathcal{R}_s \;+\; \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} \, v(s') \tag{7}$$

The Bellman equation can be concisely using matrices, where v is a column vector with one entry per state

$$v = \mathcal{R} + \gamma \mathcal{P} v \quad \rightarrow \quad \begin{pmatrix} v(1) \\ \vdots \\ v(n) \end{pmatrix} = \begin{pmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{pmatrix} + \gamma \begin{pmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{pmatrix} \begin{pmatrix} v(1) \\ \vdots \\ v(n) \end{pmatrix} \tag{8}$$

The Bellman equation is linear. It can be solved directly by $v = (I - \gamma \mathcal{P})^{-1} \mathcal{R}$.

- The Computational complexity is $O(n^3)$ for n states.
- Direct solution only possible for small MRPs
- Iterative methods for large MRPs, e.g. Dynamic programming, Monte-Carlo evaluation, Temporal-Difference learning

1.3 Markov Decision Process (MDP)

A MDP is a MRP with decisions. It is an *environment* in which all states are Markov.
A MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$.

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{A}$ is a (finite) set of actions
- $\mathcal{P}$ is a state transition probability matrix
  $\mathcal{P}^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s,\ A_t = a\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s,\ A_t = a]$
- γ is a discount factor, $\gamma \in [0, 1]$

A policy π is a distribution over actions given states. It fully defines the behaviour of an agent.

$$\pi(a|s) = \mathbb{P}\left[\ S_t = s,\ A_t = a\right] \tag{9}$$

In an MDP, the policies depend on the current state (not the history). Policies are stationary: $A_t = \pi(\dot{|}S_t), \forall t > 0$

Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle andapolicy \pi$:

- The state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- The state and reward sequence $S_1, , R_2, S_2, \ldots$ is a Markov reward process $\langle \mathcal{S},\ \mathcal{P},\ \mathcal{R},\ \gamma \rangle$ where

$$\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s)\, \mathcal{P}^a_{s,s'} \qquad \mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s)\, \mathcal{R}^a_s \tag{10}$$

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π:

$$v_\pi(s) \;=\; \mathbb{E}_\pi[G_t \mid S_t \;=\; s] \tag{11}$$

The *action-value function* $q_\pi(s, a)$ of an MDP is the expected return starting from state s, taking action a, and then following policy π:

$$q_\pi(s, a) \;=\; \mathbb{E}_\pi[G_t \mid S_t \;=\; s,\; A_t \;=\; a] \tag{12}$$

The state-value and action-value functions can again be decomposed into Bellman equations consisting of immediate reward plus discounted value of successor state:

$$v_\pi \;=\; \mathbb{E}_\pi\left[R_{t+1} \;+\; \gamma v_\pi(S_{t+1}) \;\mid S_t \;=\; s\right] \tag{13}$$

$$q_\pi \;=\; \mathbb{E}_\pi\left[R_{t+1} \;+\; \gamma v_\pi(S_{t+1},\; A_{t+1}) \;\mid S_t \;=\; s,\; A_t \;=\; a\right] \tag{14}$$

Basically, the state-value averages over the different actions that can be taken:

$$v_\pi(s) \;=\; \sum_{a \in \mathcal{A}} \pi(a|s)\, q_\pi(s,\; a) \tag{15}$$

The other way around, by using the probabilities of the transition dynamics we can average through t
continue at 58:00 write down q pi