Lecture 2: Markov Decision Processes

Markov processes

All RL problems can be formalised as MDPs, even partially observable problems. **Bandits** are MDPs with one state.

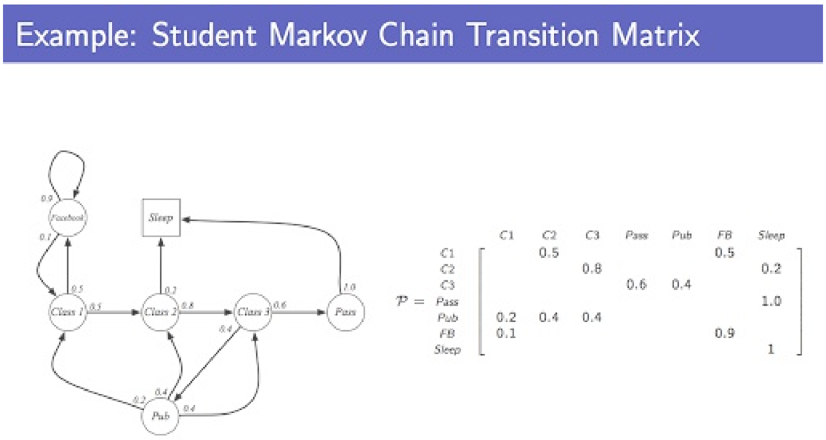
For a Markov state s and successor state s', the **state transition probability** is:

$$P_{ss'} = P[S_{t+1} = s' | S_t = s]$$

A **state transition probability matrix** defines transition probabilities from s to all s', where the $(i, j)^{th}$ element is the probability of state i to get to state j. Then, each row sums to 1.

A **Markov process** (or **Markov Chain**) is a sequence of Markov states, denoted as $< S, P >$.

- S is a finite set of states
- P is a state transition probability matrix



Left: states and probabilities. *Right*: corresponding transition matrix.

Markov reward processes

A **Markov reward process** is a Markov chain denoted $< S, P, R, \gamma >$.

- R is a reward s.t.

$$R_s = \mathbf{E}[R_{t+1} | S_t = s]$$

- γ is a discount factor, $\gamma \in [0, 1]$. It's the *present* value of *future* rewards. If all sequences terminate (that is, $\gamma = 1$), it's an **undiscounted** Markov reward process.

The goal is to optimise the **return** G_t, the total discounted reward fom time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

So, the value of receiving reward R after $k+1$ time-steps is $\gamma^k R$.

(In the student example above, we simply assign an R value to each state. *Ex.* passing has $R = 10$, Facebook has $R = -1$, etc.)

Value functions

The **state-value function** $v(s)$ is the expected return starting from state s (we prefer to be in states of higher value).

$$v(s) = \mathbf{E}[G_t | S_t = s]$$

So if I understand correctly, when $\gamma = 0$, $v(s) = R(s)$.

Bellman equation

The **Bellman equation** can be used to calculate value. $v(s)$ is the sum of the immediate reward R_{t+1} and discounted value of successor state $\gamma v(S_{t+1})$.

$$v(s) = \mathbf{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$$

It can also be expressed in matrices:

$$v = R + \gamma P v$$

Markov decision processes (MDP)

A **Markov decision process** is a reward process denoted $< S, A, P, R, \gamma >$, adding in A (a finite set of actions). So, there are *decisions* rather than/in addition to *probabilities* of states happening.

A **policy** π is how you make decisions. It's a distribution over actions given states (stochastic transition matrix). It's *stationary* (time-independent).

$$\pi(a|s) = P[A_t = a | S_t = s]$$

Given an MDP, notice that

- The state sequence $S_1, S_2, \ldots$ is a Markov process
- The state/reward sequence $S_1, R_2, S_2, \ldots$ is a Markov reward process

State-value functions

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

$$v_\pi(s) = \mathbf{E}_\pi[G_t | S_t = s]$$

Ya boi Bellman decomposed it like this:

$$v_\pi(s) = \mathbf{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

Action-value functions

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

$$q_\pi(s, a) = \mathbf{E}_\pi[G_t | S_t = s, A_t = a]$$

It can be decomposed similarly to the state-value function:

$$q_\pi(s, a) = \mathbf{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

Optimal value functions

The **optimal state-value function** $v_*(s)$ is the maximum value function over all policies:

$$v_*(s) = \max_\pi v_\pi(s)$$

The **optimal action-value function** $q_*(s, a)$ is the maximum action-value function over all policies:

$$q_*(s, a) = \max_\pi q_\pi(s, a)$$

A policy π is an **optimal policy** if $\pi \geq \pi'$ if $v_\pi(s) \geq v_{\pi'}(s), \forall s$.

For any MDP, there is an optimal policy π_*. All optimal policies achieve the optimal value function $v_{\pi_*}(s) = v_*(s)$ and the optimal action-value function $q_{\pi_*}(s, a) = q_*(s, a)$.

Bellman optimality equation for v_*

Basically, the value at each state s is the maximum return from all actions taken at that state:

$$v_*(s) = \max_a q_*(s, a)$$

Then, one-step lookahead:

$$q_*(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

Then, two-step lookahead:

$$v_*(s) = \max_a R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

Notice that it's non-linear. There do exist iterative solution methods.

Reference

- Video lecture
- Slides