

Reinforcement Learning

Lecture 2: Markov Decision Processes

Chris G. Willcocks

Durham University



Lecture covers Chapter 3 in Sutton & Barto [3] and uses David Silver's examples [2]

1 Markov Chains

- markov property
- state transition matrix
- definition and example

2 Markov Reward Process

- definition and example
- the return
- state value function
- the Bellman equation

3 Markov Decision Process

- definition and example
- policies
- state and action value functions
- the Bellman equation
- optimal state and action value functions
- the Bellman optimality equations



With the **Markov property**, we can throw away the history and just use the agents state:

Definition: Markov property

A state S_t is **Markov** if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, S_2, \dots, S_t]$$

- For example, a **chess board**
 - We don't need to know how the game was played up to this point
- The state fully characterises the distribution over future events:

$$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$



The probability of transitioning from state s to s' for a Markov state is:

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

where the **state transition probability** for all states to all successor states can be expressed as a large matrix:

$$\mathcal{P} = \begin{matrix} & \overbrace{\hspace{1.5cm}}^{\text{to}} \\ \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \end{matrix},$$

and each row sums to 1.

Click [↗](#) to try a demo [1]



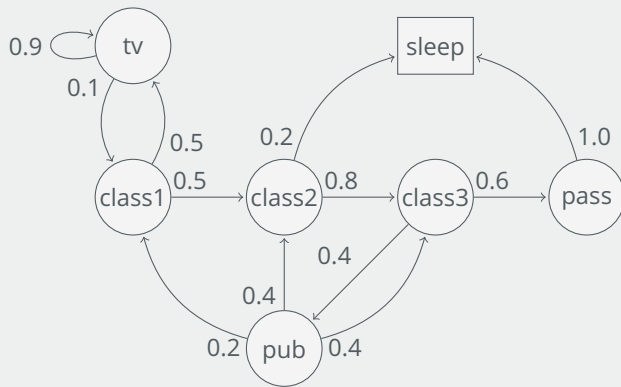
A **Markov chain** (also called Markov Process) is a set of states and a state-transition matrix

Definition: Markov chain

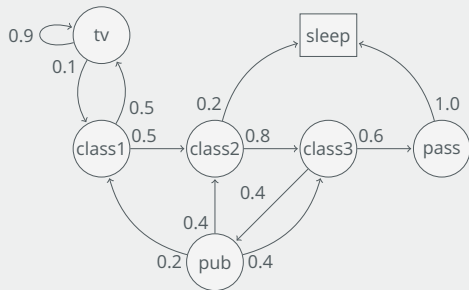
A **Markov chain** is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- \mathcal{S} is a finite set of states
- \mathcal{P} is the state-transition matrix where $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$

Example: Markov Chain



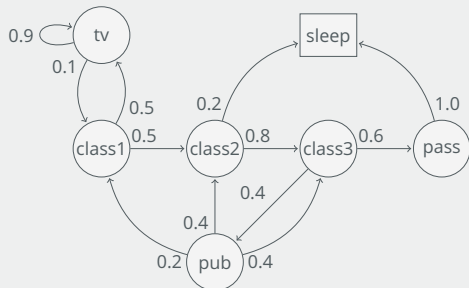
Example: Markov Chain



State Transition Matrix

$$P = \begin{matrix} & \begin{matrix} c1 & c2 & c3 & pass & pub & tv & sleep \end{matrix} \\ \begin{matrix} c1 \\ c2 \\ c3 \\ pass \\ pub \\ tv \\ sleep \end{matrix} & \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ 0.2 & 0.4 & 0.4 & & & & 1.0 \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1.0 \end{bmatrix} \end{matrix}$$

Example: Markov Chain



Episode

An episode is a varying-length sample of a Markov chain:

$$S_1, S_2, \dots, S_T,$$

for example starting from $S_1 = \text{class1}$:

Episode samples

c1,c2,c3,pass,sleep

c1,tv,tv,tv,c1,c2,c3,pub,c2,sleep



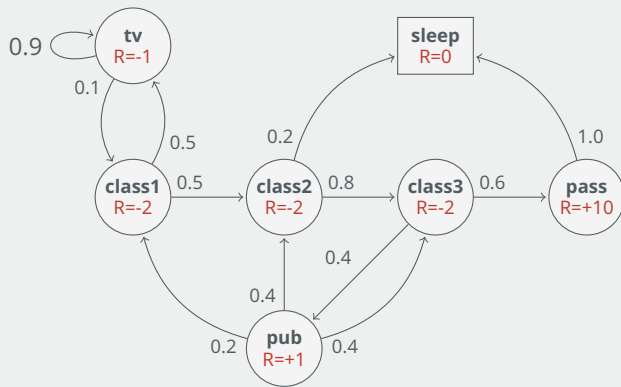
A Markov **reward** process is a Markov Chain with a **reward** function

Definition: Markov reward process

A **Markov reward process** is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- \mathcal{S} is a finite set of states
- \mathcal{P} is the state-transition matrix where $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$
- \mathcal{R} is a **reward** function where $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ is the **discount** rate $\gamma \in [0, 1]$

Example: Markov Reward Process





The **return** G_t , in the simplest case, is the total future reward:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T$$

In practice, we discount rewards into the future by the *discount rate* $\gamma \in [0, 1]$.

Definition: The return

The return G_t is the discounted total future reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Definition: The state value function

The **state value function** $v(s)$ in an MRP is the long-term value of a state:

$$v(s) = \mathbb{E}[G_t \mid S_t = s],$$

for example calculated by sampling episodes...

Sample episodes

c1,c2,c3,pass,sleep

c1,tv,tv,tv,c1,c2,c3,pub,c2,sleep

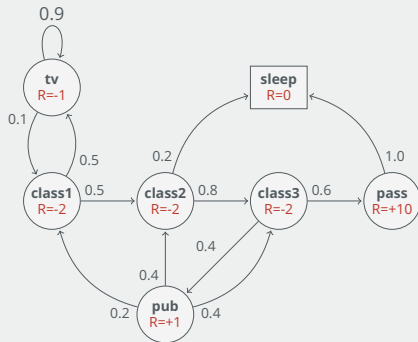
c1,c2,sleep

...

Example: Puppy



Example: MRP



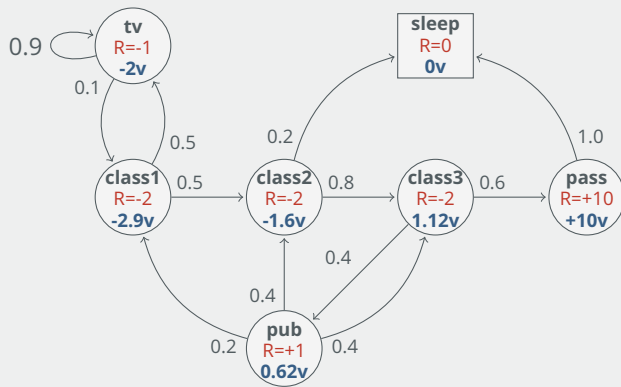
Example: The state value function

This is an example $v(s)$ with $s = \text{'class1'}$ and $\gamma = \frac{1}{2}$:

$$\begin{aligned}
 G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
 &= R_{t+1} + \frac{1}{2} R_{t+2} + \frac{1}{4} R_{t+3} + \dots
 \end{aligned}$$

Episode samples	Value function
c1,c2,c3,pass,sleep	$v_1 = -2 - \frac{1}{2} \cdot 2 - \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 10 = -2.25$
c1,tv,tv,c1,c2,c3,pub,c2,sleep	$v_1 = -2 - \frac{1}{2} \cdot 1 - \frac{1}{4} \cdot 1 + \frac{1}{8} \cdot \dots = -3.125$
c1,c2,sleep	$v_1 = -2 - \frac{1}{2} \cdot 2 - \frac{1}{4} \cdot 0 + \frac{1}{8} \cdot \dots = -3$
...	$\dots = -2.9$

Example: Markov Reward Process for $\gamma = 0.5$





Through a series of identities, we can decompose the value function into the **immediate reward** R_{t+1} and the discounted **value of the next state** $\gamma v(S_{t+1})$.

Definition: Bellman equation for MRP

The Bellman equation is:

$$\begin{aligned} v(s) &= \mathbb{E}[G_t \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots) \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s], \end{aligned}$$

which is equivalent to:

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



The Bellman equation can be expressed with matrices:

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix},$$

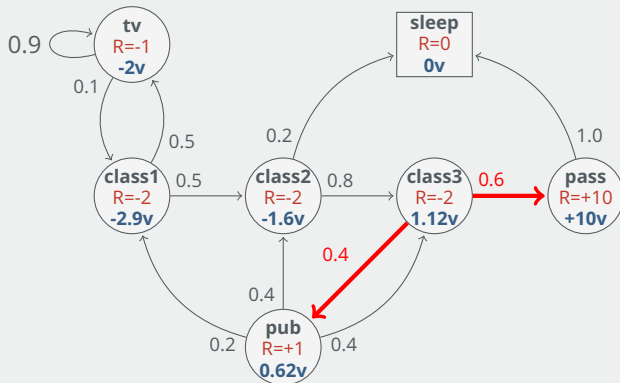
which is a linear equation that can be solved:

$$\begin{aligned} v &= \mathcal{R} + \gamma \mathcal{P} v \\ (\mathbf{I} - \gamma \mathcal{P}) v &= \mathcal{R} \\ v &= (\mathbf{I} - \gamma \mathcal{P})^{-1} \mathcal{R}, \end{aligned}$$

where \mathbf{I} is the identity matrix. Unfortunately this matrix inversion is too slow, except for small MDPs, so we use iterative methods for larger MDP (MC evaluation and TD learning).

Verification: MRP for $\gamma = 0.5$

$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s') = -2 + 0.5 * (0.6 * 10 + 0.4 * 0.62) = 1.12$$





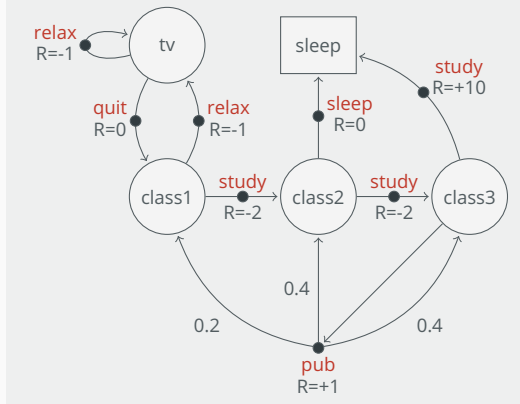
A Markov **decision** process adds 'actions' so the transition probability matrix now depends on which action the agent takes.

Definition: Markov decision process

A **Markov decision process** 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 the state-transition matrix where $\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- \mathcal{R} is a **reward** function where $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- γ is the **discount** rate $\gamma \in [0, 1]$

Example: Markov Decision Process





A policy is a distribution over actions which determines how agents should behave in the environment.

- A lazy agent will sample relaxing actions more than frequently than studying
- A high-performing agent will study at all classes, then study more at home!

Definition: Policy

A policy π is a distribution over actions given a state:

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

Definition: The state-value function

The **state-value function** $v_{\pi}(s)$ is the same, but its the return when following a given policy π :

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

Definition: The action-value function

The **action-value function** is the long term-value of a state when choosing an action with policy π :

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

Example: Arizona trail





Similarly to MRPs, the state-value function can be decomposed into the immediate reward and the discounted value of the next state:

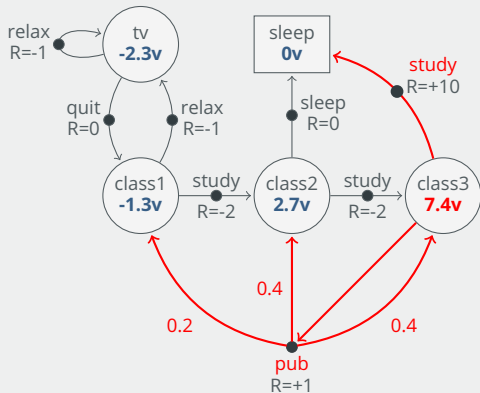
$$\begin{aligned}v_{\pi}(s) &= \mathbb{E}_{\pi}[G_t \mid S_t = s] \\&= \mathbb{E}_{\pi}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s] \\&= \sum_{a \in \mathcal{A}} \pi(a|s) q_{\pi}(s, a),\end{aligned}$$

which is also the case for the action-value function, where:

$$\begin{aligned}q_{\pi}(s, a) &= \mathbb{E}_{\pi}[G_t \mid S_t = s, A_t = a] \\&= \mathbb{E}_{\pi}[R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \\&= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi}(s').\end{aligned}$$



Verification: MDP with average policy



Verification

Under the policy π where we do everything $\{\text{study, pub}\}$ with 50% probability and $\gamma = 1$:

$$\begin{aligned} v_{\pi}(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) q_{\pi}(s, a) \\ &= \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi}(s') \right) \\ &= \frac{1}{2} * 10 \\ &\quad + \frac{1}{2} (1 + 0.2(-1.3v) + 0.4(2.7v) + 0.4(7.4v)) \\ &= 7.4v \end{aligned}$$

Definition: The optimal state-value function

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

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

Definition: The optimal action-value function

The **optimal action-value function** is the maximum action value function over all policies:

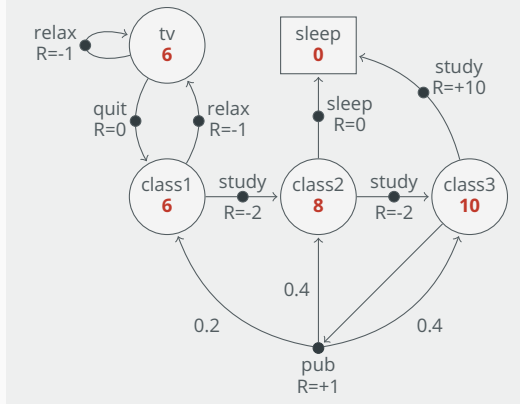
$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

Example: Mo Farah



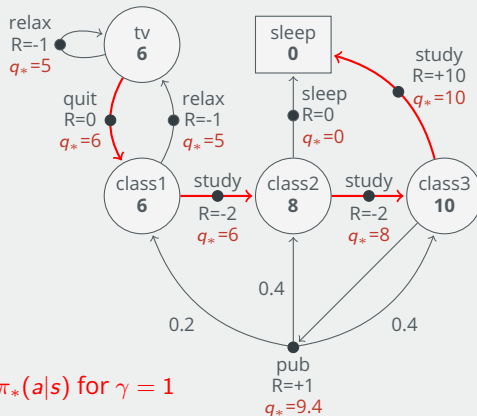


Example: $v_*(s)$ for $\gamma = 1$





Example: $q_*(s, a)$ for $\gamma = 1$



$\pi_*(a|s)$ for $\gamma = 1$



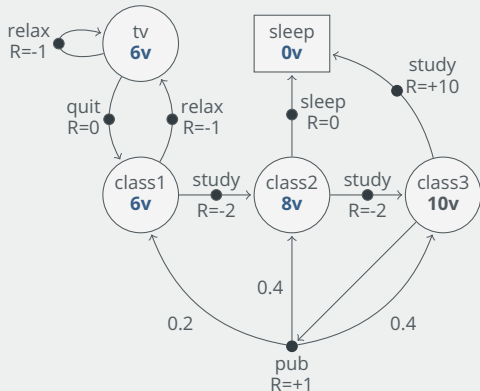
The optimal value functions are similarly recursively related by the Bellman optimality equations, where:

$$\begin{aligned} v_*(s) &= \max_{\pi} v_{\pi}(s) \\ &= \max_a q_*(s, a), \end{aligned}$$

and the optimal action-value function:

$$\begin{aligned} q_*(s, a) &= \max_{\pi} q_{\pi}(s, a) \\ &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s'). \end{aligned}$$

Verification: MDP with average policy




Verification

The optimal state-value for class3 following $\gamma = 1$ requires q_* for the pub action:

$$\begin{aligned}
 v_*(s) &= \max_a q_*(s, a) \\
 &= \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s') \\
 &= \max \left\{ 10 + 1 * (0v), \right. \\
 &\quad \left. (1 + 0.2(6v) + 0.4(8v) + 0.4(10v)) \right\} \\
 &= \max \{ q_* = 10, q_* = 9.4 \} \\
 &= \mathbf{10v}
 \end{aligned}$$



- [1] V. Powell and L. Lehe.
Markov chains explained visually.
<https://setosa.io/markov/index.html>, 2014.
- [2] D. Silver.
Reinforcement learning lectures.
<https://www.davidsilver.uk/teaching/>, 2015.
- [3] R. S. Sutton and A. G. Barto.
Reinforcement learning: An introduction (second edition).
[Available online](#) , MIT press, 2018.