3- Markov Decision Processes

Melih Kandemir

University of Southern Denmark
Department of Mathematics and Computer Science (IMADA)
kandemir@imada.sdu.dk

Fall 2022

Value Functions in MDPs

State-value function (a.k.a. value function) is the expected return of starting from state s and following policy π

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

(Intuitively, a measure of how good it is to be in a state)

Action-value function is the expected return of starting from state s, taking action a, and following policy π

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

(Intuitively, a measure of how good it is to take a certain action in a certain state)

Relationship between state and action value functions

Integrating out the action variable in the action-value function wrt the policy distribution gives the state-value function

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(A_t = a | S_t = s) q_\pi(s, a).$$

Bellman *Expectation* Equation

For the state-value function

$$\begin{aligned} v_\pi(s) &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s], \\ &= \sum_{a \in \mathcal{A}} \pi(A_t = a | S_t = s) \Big(\langle r_s^a \rangle + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s') \Big) \end{aligned}$$

where $\langle r_s^a \rangle = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$.

For the action-value function

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \\ &= \langle r_s^a \rangle + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s') \\ &= \langle r_s^a \rangle + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(A_{t+1} = a' | S_{t+1} = s') q_\pi(s', a'). \end{aligned}$$

Bellman *Expectation* Equation

$$\mathbf{v} = (\mathbf{I} - \gamma \mathbf{P}^{\pi})^{-1} \langle \mathbf{r}^{\pi} \rangle$$

Remark: The solution is a function of $\pi(\cdot|\cdot)$!

Optimal Value Functions and Policies

The optimal state-value function is defined as

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

and the optimal action-value function as

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

In order to maximize subject to policies, we need to define an ordering between them

$$\pi \geq \pi', \;\; \text{if} \;\; v_\pi(s) \geq v_{\pi'}(s), \; \forall s.$$

Existence of an Optimal Policy

Theorem. For any MDP,

- there exists $\pi_* \geq \pi$, $\forall \pi$.
- optimal policy achieves the optimal value functions: $v_{\pi_*}(s) = v_*(s)$ and $q_{\pi_*}(s, a) = q_*(s, a)$.

Finding an Optimal Policy

Given $q_*(s, a)$, we can find a *deterministic* optimal policy by

$$\pi_*(A_t = a | S_t = s) = \begin{cases} 1, & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s, a), \\ 0, & \text{otherwise.} \end{cases}$$

That is why the action-value function exists.

Bellman *Optimality* Equation

For the state-value function

$$\begin{aligned} v_*(s) &= \max_a \; q_*(s,a) \\ &= \max_a \left[\langle r_s^a \rangle + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_*(s') \right], \end{aligned}$$

and for the action-value function

$$q_*(s,a) = \langle r_s^a \rangle + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \; \max_{a'} \; q_*(s',a').$$

Note the cyclic dependency between $v_*(\cdot)$ and $q_*(\cdot,\cdot)$. We will exploit this fact later.

Finding the optimal policy for an MDP

Highly non-linear problem without closed-form solution. Iterative alternatives include

- Value iteration
- Policy iteration
- Q-learning
- SARSA

Partially Observable MDPs

Known as POMDPs. The case when the environment cannot be accurately observed.

Defined as a tuple of six entities $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where

- $\mathcal{S}$ is the set of environment states: $S_t = s$ with $s \in \mathcal{S},\ \forall t$.
- $\mathcal{A}$ is the set of actions: $A_t = a$ with $a \in \mathcal{A},\ \forall a$.
- $\mathcal{R}$ is the set of rewards: $R_t = r$ with $r \in \mathcal{R},\ \forall r$.
- $\mathcal{O}$ is the set of observations: $O_t = o$ with $o \in \mathcal{O},\ \forall o$.
- $\gamma \in [0, 1]$ is the **discount factor**.
- $\mathcal{P} = P(R_{t+1}, O_{t+1}, S_{t+1} | S_t, A_t)$ is the **environment dynamics model** that naturally decomposes according to the chain rule as

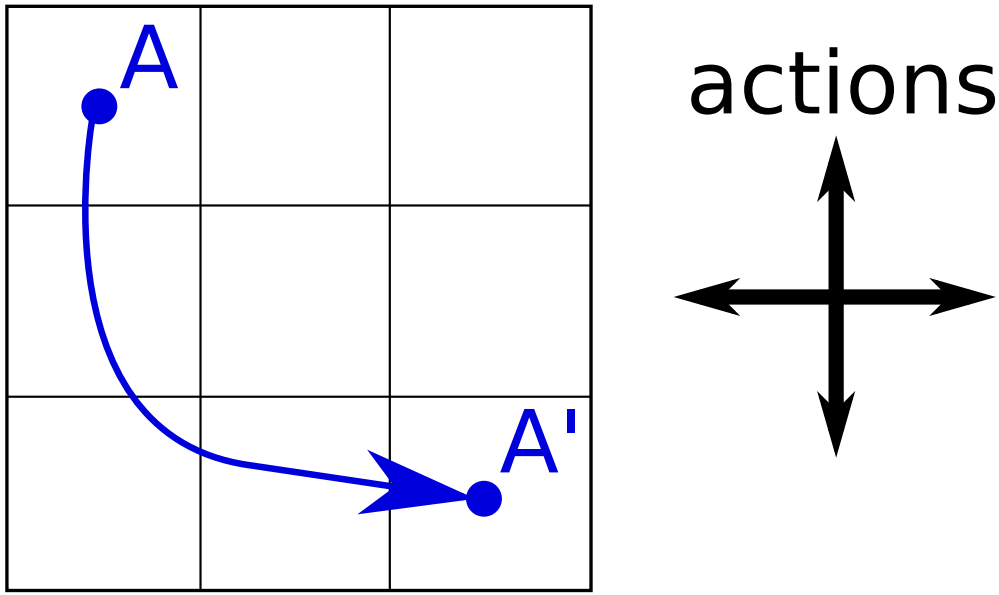
$$P(R_{t+1}, O_{t+1}, S_{t+1} | S_t, A_t) = \underbrace{P(O_{t+1} | S_t)}_{Observation\ model} \underbrace{P(R_{t+1} | S_t, A_t)}_{Reward\ model} \underbrace{P(S_{t+1} | S_t, A_t)}_{transition\ model}.$$

No free lunch theorem for RL

Approximating solutions to MDPs with RL means

- Focus on more frequent cases (state-action pairs)
- at the expense of *very bad* performance on rare cases.

Example: Gridworld



- -1 reward for attempt to go off the grid.
- +10 reward for arriving at A'.
- 0 reward otherwise.

Example: Gridworld

$\langle r_s^a \rangle$	**L**	**U**	**R**	**D**
00	-1	-1	0	0
01	0	-1	0	0
02	0	-1	-1	0
10	-1	0	0	0
11	0	0	0	0
12	0	0	-1	+10
20	-1	0	0	-1
21	0	0	+10	-1
22	0	0	-1	-1

Choose a good policy

Always go right or down.

$\pi(s\|a)$	**L**	**U**	**R**	**D**
00	0	0	1/2	1/2
01	0	0	1/2	1/2
02	0	0	0	1
10	0	0	1/2	1/2
11	0	0	1/2	1/2
12	0	0	0	1
20	0	0	1	0
21	0	0	1	1
22	1/2	1/2	0	0

The resultant value function becomes

0	0	0
0	+49.5	+10
0	+100	-

Now choose a bad policy

Go to a random direction.

$\pi(s\|a)$	**L**	**U**	**R**	**D**
00	0	0	1/2	1/2
01	1/3	0	1/3	1/3
02	1/2	0	0	1/2
10	0	1/3	1/3	1/3
11	1/4	1/4	1/4	1/4
12	1/3	1/3	0	1/3
20	0	1/2	1/2	0
21	0	1/3	1/3	1/3
22	1/2	1/2	0	0

The resultant value function becomes

3.35	3.09	2.98
4.34	4.08	3.53
5.19	7.18	4.28