CSCI E-89C Deep Reinforcement Learning

Harvard Summer School

Dmitry Kurochkin

Summer 2020
Lecture 3

Contents

Contents

State, Action, and Reward Sets

State set:

$\mathcal{S}$ = set of all possible states S_t of the environment, excluding the terminal state if the problem is episodic

$\mathcal{S}^+ = \mathcal{S} \cup \{\text{terminal state}\}$

Action set:

$\mathcal{A}(s)$ = set of all admissible actions A_t the agent can take, given $S_t = s$

Reward set:

$\mathcal{R}$ = set of all possible rewards R_t the agent can receive

Contents

Transition Probabilities

Transition Probabilities:

$$p(s', r|s, a) \doteq P\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}$$

Contents

Value Functions

Value Functions:

State-value: $v_\pi(s) \doteq E_\pi\left[G_t | S_t = s\right]$

Action-value: $q_\pi(s,a) \doteq E_\pi\left[G_t | S_t = s, A_t = a\right]$

Value Functions

Value Functions:
State-value: $v_\pi(s) \doteq E_\pi\left[G_t | S_t = s\right]$
Action-value: $q_\pi(s, a) \doteq E_\pi\left[G_t | S_t = s, A_t = a\right]$

Optimal Value Functions:
take $\max_\pi$ - denote results by $v_*(\cdot)$, $q_*(\cdot)$

Methods for finding $\pi_*(a|s)$

Model-based approach:
If we know $p(s', r|s, a)$ then we can solve the Bellman equation: Dynamic Programming

Model-free approaches:
If we do not know $p(s', r|s, a)$: Monte Carlo Method etc.

Contents

Bellman Equation for $v_\pi(s)$

v_π from q_π:

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

q_π from v_π:

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

Bellman Equation for $v_\pi(s)$

v_π from q_π:

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$$

q_π from v_π:

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

Bellman equation:

$$v_\pi(s) = \sum_a \pi(a|s) \underbrace{\sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right]}_{q_\pi(s,a)}$$

Bellman Equation for $v_*(s)$

v_* from q_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s, a)$$

q_* from v_*:

$$q_*(s, a) = E\left[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\right]$$

Bellman Equation for $v_*(s)$

v_* from q_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s, a)$$

q_* from v_*:

$$q_*(s, a) = E\left[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\right]$$

Bellman equation:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \underbrace{\sum_{s', r} p(s', r | s, a) \left[r + \gamma v_*(s')\right]}_{q_*(s,a)}$$

Contents

Fixed point iteration

Bellman equation for v_π:

$$v_\pi(s) = \sum_a \pi(a|s) \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]}_{q_\pi(s,a)}$$

Bellman equation for v_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]}_{q_*(s,a)}$$

Fixed point iteration to solve $x = f(x)$:

1. initialize x_0
2. for $k \geq 0$ compute $x_{k+1} \doteq f(x_k)$

Policy Evaluation: Solving for $v_\pi(s)$

Bellman equation for v_π:

$$v_\pi(s) = \sum_a \pi(a|s) \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]}_{q_\pi(s,a)}$$

Bellman equation for v_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]}_{q_*(s,a)}$$

Fixed point iteration to solve $x = f(x)$:

1. initialize x_0
2. for $k \geq 0$ compute $x_{k+1} \doteq f(x_k)$

Policy Evaluation: Solving for $v_\pi(s)$

Let $v_k(s)$, $k = 0, 1, 2, \ldots$ denote estimates of $v_\pi(s)$. The fixed-point iteration can be written as follows:

$$v_{k+1}(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

Policy Evaluation: Solving for $v_\pi(s)$

Let $v_k(s)$, $k = 0, 1, 2, \ldots$ denote estimates of $v_\pi(s)$. The fixed-point iteration can be written as follows:

$$v_{k+1}(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r\,|\,s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Source: *Reinforcement Learning: An Introduction* by R. Sutton and A. Barto

Example

The Environment has three states: 1, 2, and 3. Possible transitions are:
(1) $1 \mapsto 1$, $1 \mapsto 2$;
(2) $2 \mapsto 1$, $2 \mapsto 2$, $2 \mapsto 3$;
(3) $3 \mapsto 2$, $3 \mapsto 3$.

Actions of the Agent are decoded by -1, 0, and $+1$, which correspond to its intention to move left, stay, and move right, respectively.

The Environment, however, does not always respond to these intentions exactly, and there is 10% chance that action 0 will result in moving to the left (if moving to the left is admissible), and action $+1$ will result in staying - in other words, there is an "east wind."

If the process enters state 3, the Environment generates reward = 1. In all other cases the reward is 0. Discounting $\gamma = 0.9$.

Example

Under $\pi(0|s) = 1$ for all $s \in \mathcal{S}$,
the expected cumulative discounted reward starting in states 3 would be:

$$\begin{aligned} v_\pi(3) &= E_\pi\left[G_0 | S_0 = 3\right] \\ &= 0.9 \cdot (1 + \gamma \cdot 0.9 \cdot (1 + \gamma \cdot 0.9 \cdot (1 + ...))) \\ &= 0.9 \cdot 1 + \gamma \cdot 0.9^2 \cdot 1 + \gamma^2 \cdot 0.9^3 \cdot 1 + \dots \\ &= \frac{0.9}{1 - \gamma \cdot 0.9} = 4.74. \end{aligned}$$

Example

Under $\pi_*(+1|1) = 1$, $\pi_*(+1|2) = 1$, $\pi_*(0|3) = 1$,
the expected cumulative discounted reward starting in states 3 would be:

$$\begin{aligned} v_*(3) &= E_{\pi_*}\left[G_0 | S_0 = 3\right] \\ &= 0.9 \cdot \frac{1}{1-0.9} = 9. \end{aligned}$$

Here, $\frac{1}{1-0.9}$ is the expected cumulative reward if there is no wind.

Example

Solving the Bellman equation explicitly for this policy $\pi_*(a|s)$: plug

$$\pi_*(-1|1) = 0, \pi_*(0|1) = 0, \pi_*(+1|1) = 1,$$
$$\pi_*(-1|2) = 0, \pi_*(0|2) = 0, \pi_*(+1|2) = 1,$$
$$\pi_*(-1|3) = 0, \pi_*(0|3) = 1, \pi_*(+1|3) = 0.$$

$$p(s' = 1, r = 0|s = 1, a = 0) = 1,$$
$$p(s' = 1, r = 0|s = 1, a = +1) = 0.1, p(s' = 2, r = 0|s = 1, a = +1) = 0.9,$$
$$p(s' = 1, r = 0|s = 2, a = -1) = 1,$$
$$p(s' = 1, r = 0|s = 2, a = 0) = 0.1, p(s' = 2, r = 0|s = 2, a = 0) = 0.9,$$
$$p(s' = 2, r = 0|s = 2, a = +1) = 0.1, p(s' = 3, r = 1|s = 2, a = +1) = 0.9,$$
$$p(s' = 2, r = 0|s = 3, a = -1) = 1,$$
$$p(s' = 2, r = 0|s = 3, a = 0) = 0.1, p(s' = 3, r = 1|s = 3, a = 0) = 0.9.$$

to the Bellman eq.: $v_{\pi_*}(s) = \sum_a \pi_*(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_{\pi_*}(s')\right]$ and get a linear system of 3 equations.

Example

For $s = 1$ and $a \in \{-1, 0, +1\}$, we have:

$\pi_*(-1|1) = 0, \pi_*(0|1) = 0, \pi_*(+1|1) = 1$

$p(s' = 1, r = 0|s = 1, a = 0) = 1,$
$p(s' = 1, r = 0|s = 1, a = +1) = 0.1, p(s' = 2, r = 0|s = 1, a = +1) = 0.9$

and the Bellman eq. $v_{\pi_*}(s) = \sum_a \pi_*(a|s) \sum_{s',r} p(s', r|s, a) [r + \gamma v_{\pi_*}(s')]$ becomes:

$$
\begin{aligned}
v_*(1) &= \underbrace{\pi_*(-1|1)}_{0} \cdot \left(\sum_{s',r} \dots \right) + \underbrace{\pi_*(0|1)}_{0} \cdot \left(\sum_{s',r} \dots \right) + \underbrace{\pi_*(+1|1)}_{1} \cdot \left(\sum_{s',r} \dots \right) \\
&= p(s' = 1, r = 0|s = 1, a = +1) [0 + \gamma v_*(1)] \\
&+ p(s' = 2, r = 0|s = 1, a = +1) [0 + \gamma v_*(2)] \\
&= 0.1 \cdot [0 + \gamma v_*(1)] + 0.9 \cdot [0 + \gamma v_*(2)]
\end{aligned}
$$

Example

Similarly, for $s = 2$ and $s = 3$ we arrive at the following system of equations:

$$\begin{cases} v_*(1) = & 0.1 \cdot [0 + \gamma v_*(1)] + 0.9 \cdot [0 + \gamma v_*(2)] \\ v_*(2) = & 0.1 \cdot [0 + \gamma v_*(2)] + 0.9 \cdot [1 + \gamma v_*(3)] \\ v_*(3) = & 0.1 \cdot [0 + \gamma v_*(2)] + 0.9 \cdot [1 + \gamma v_*(3)] \end{cases}$$

Example

Similarly, for $s = 2$ and $s = 3$ we arrive at the following system of equations:

$$\begin{cases} v_*(1) = & 0.1 \cdot [0 + \gamma v_*(1)] + 0.9 \cdot [0 + \gamma v_*(2)] \\ v_*(2) = & 0.1 \cdot [0 + \gamma v_*(2)] + 0.9 \cdot [1 + \gamma v_*(3)] \\ v_*(3) = & 0.1 \cdot [0 + \gamma v_*(2)] + 0.9 \cdot [1 + \gamma v_*(3)] \end{cases}$$

The iterative policy iteration in this example would be:
pick some $v_0(1)$, $v_0(2)$, $v_0(3)$ and run for $k \geq 0$ until converges:

$$\begin{cases} v_{k+1}(1) = & 0.1 \cdot [0 + \gamma v_k(1)] + 0.9 \cdot [0 + \gamma v_k(2)] \\ v_{k+1}(2) = & 0.1 \cdot [0 + \gamma v_k(2)] + 0.9 \cdot [1 + \gamma v_k(3)] \\ v_{k+1}(3) = & 0.1 \cdot [0 + \gamma v_k(2)] + 0.9 \cdot [1 + \gamma v_k(3)] \end{cases}$$

Policy Improvement

Suppose we know $v_\pi(s)$ for a given policy π.
Can this policy π be improved?

Policy Improvement

Suppose we know $v_\pi(s)$ for a given policy π.
Can this policy π be improved?

We can find the action-value function as follows

$$\begin{aligned} q_\pi(s,a) &\doteq E\left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a\right], \\ &= \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right] \end{aligned}$$

and what if there an action, $\pi'(s)$, such that $q_\pi(s, \pi'(s)) \geq v_\pi(s)$?

If we <u>always</u> follow π', will this new policy π' be at least as good as π?

Policy Improvement Theorem

Theorem

If $q_\pi(s, \pi'(s)) \geq v_\pi(s)$ then $v_{\pi'}(s) \geq v_\pi(s)$.

Policy Improvement Theorem

Theorem

If $q_\pi(s, \pi'(s)) \geq v_\pi(s)$ then $v_{\pi'}(s) \geq v_\pi(s)$.

Proof:

$$
\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= E\left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = \pi'(s)\right] \\
&= E_{\pi'}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\right] \\
&\leq E_{\pi'}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) | S_t = s\right] \\
&= E_{\pi'}\left[R_{t+1} + \gamma E_{\pi'}\left[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})\right] | S_t = s\right] \\
&= E_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \Big| S_t = s\right] \\
&\leq E_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \Big| S_t = s\right] \\
&\ \ \vdots \\
&\leq E_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \Big| S_t = s\right] \\
&= v_{\pi'}(s)
\end{aligned}
$$

Convergence to $v_*(s)$

Suppose π' does not further improve π, i.e. $v_{\pi'}(s) = v_\pi(s)$ for all $s \in \mathcal{S}$, then

$$
\begin{aligned}
v_{\pi'}(s) &= \max_a q_\pi(s,a) \\
&= \max_a q_{\pi'}(s,a) \\
&= \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_{\pi'}(s')\right]
\end{aligned}
$$

i.e. $v_{\pi'}(s)$ solves the Bellman optimality equation.
Then π' is the optimal policies!

Policy Iteration: Solving for $v_*(s)$

Bellman equation for v_π:

$$v_\pi(s) = \sum_a \pi(a|s) \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]}_{q_\pi(s,a)}$$

Bellman equation for v_*:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \underbrace{\sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right]}_{q_*(s,a)}$$

Fixed point iteration to solve $x = f(x)$:

1. initialize x_0
2. for $k \geq 0$ compute $x_{k+1} \doteq f(x_k)$

Policy Iteration: Solving for $v_*(s)$

Let $v_k(s)$, $k = 0, 1, 2, \ldots$ denote an estimate of $v_*(s)$. The fixed-point iteration can be written as follows:

$$v_{k+1}(s) \doteq \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

Policy Iteration: Solving for $v_*(s)$

Let $v_k(s)$, $k = 0, 1, 2, \ldots$ denote an estimate of $v_*(s)$. The fixed-point iteration can be written as follows:

$$v_{k+1}(s) \doteq \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s', r | s, a) \left[r + \gamma v_k(s') \right]$$

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\left[r + \gamma V(s')\right]$

Source: *Reinforcement Learning: An Introduction* by R. Sutton and A. Barto