Value Iteration & Policy Iteration

Yuntian Deng

Lecture 14

Readings: RN 17.2. PM 9.5.2, 9.5.3.

Outline

Learning Goals

- Trace the execution of and implement the value iteration algorithm for solving a Markov Decision Process.
- Trace the execution of and implement the policy iteration algorithm for solving a Markov Decision Process.

Value Functions

- $V^\pi(s)$: Value of being in state s following a policy π
- $V^*(s)$: Value of being in state s following optimal policy π^*
- $Q^\pi(s,a)$: Value of taking action a while in state s and then follow π
- $Q^*(s,a)$: Value of taking action a while in state s and then follow π^*
- $\pi(a|s)$: the policy function, converting state into a distribution over actions

Expected Return

Remember that the agent's goal is to find a sequence of actions that will maximize the long-term return. We have defined the long-term return in a discounted format:

$$
\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+2} + \gamma^{T-1} R_T \\
&= R_{t+1} + \gamma G_{t+1}
\end{aligned}
$$

A value function estimates how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state) in terms of return G.

The V-function

More formally, the V-function also referred to as the state-value function, or simply V, measures the goodness of each state.

$$V^{\pi}(s) = E_{\pi}[G_t | s_t = s] = E_{\pi}[\sum_{j=0}^{T} \gamma^j R_{t+j+1} | s = s_t] \qquad (1)$$

It describes the expected value of the total return G, at time step t starting from the state s at time t and then following policy π. We use expectation E in this definition because the Environment transition function might act in a stochastic way.

The Q-function

It defines the value of taking action a in state s under a policy π, denoted by Q, as the expected Return G starting from s, taking the action a, and thereafter following policy π.

A policy can be written as $\pi(a|s)$, where $\sum_a \pi(a|s) = 1$.

$$
\begin{aligned}
Q^{\pi}(s,a) &= E_{\pi}[G_t | s_t = s, a_t = a] && (2) \\
&= E_{\pi}[\sum_{j=0}^{T} \gamma^j R_{t+j+1} | s_t = s, a_t = a] && (3)
\end{aligned}
$$

In this equation again it is used expectation E because the Environment transition function might act in a stochastic way.

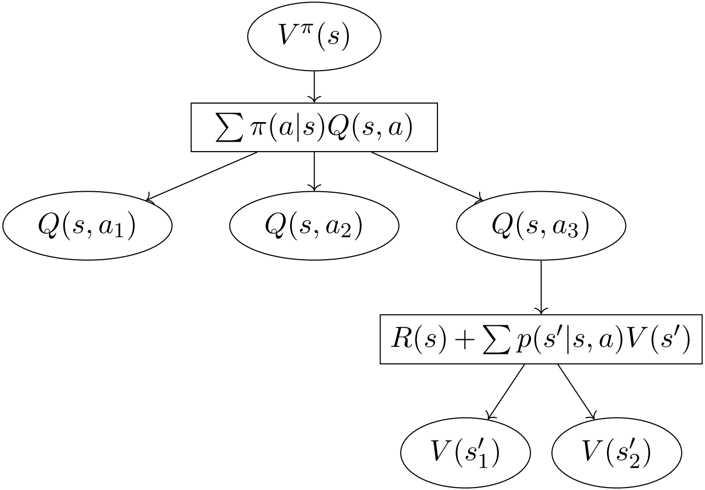
Relation between Q/V function

We can assert the state-value function is equivalent to the sum of action-value functions of all outgoing actions a, multiplied by the policy probability of selecting each action:

$$V^{\pi}(s) = \sum_{a} \pi(a|s) Q^{\pi}(s,a) \tag{4}$$

$$Q^{\pi}(s,a) = r(s) + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s') \tag{5}$$

Graph Relation between Q/V function



Learning Goals

Definition of V/Q-Function

Bellman Equation

Value Iteration

Policy Iteration

Revisiting Learning Goals

Solving for $V^*(s)$

V and Q are defined recursively in terms of each other.

$$V^*(s) = \max_a Q^*(s,a) \tag{6}$$

$$Q^*(s,a) = R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s'). \tag{7}$$

Combining equations 6 and 7, we get the Bellman equations:

$$V^*(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s'). \tag{8}$$

$V^*(s)$ are the unique solutions to the Bellman equations.

Write down $V^*(s_{11})$

Recall the grid environment from Lecture 13.

Write down the Bellman equation for $V^*(s_{11})$.

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	-0.04	$+1$

$\rightarrow$

$$\begin{aligned} V^*(s_{11}) = -0.04 + \gamma \max[&0.8V^*(s_{12}) + 0.1V^*(s_{21}) + 0.1V^*(s_{11}), \\ &0.9V^*(s_{11}) + 0.1V^*(s_{12}), \\ &0.9V^*(s_{11}) + 0.1V^*(s_{21}), \\ &0.8V^*(s_{21}) + 0.1V^*(s_{12}) + 0.1V^*(s_{11})]. \end{aligned}$$

Q: Solve the Bellman equations efficiently

Q #1: Can we solve the system of Bellman equations in polynomial time?

(A) Yes

(B) No

(C) I don't know

The Bellman equation for $V^*(s_{11})$:

$$\begin{aligned} V^*(s_{11}) = -0.04 + \gamma \max[& 0.8V^*(s_{12}) + 0.1V^*(s_{21}) + 0.1V^*(s_{11}), \\ & 0.9V^*(s_{11}) + 0.1V^*(s_{12}), \\ & 0.9V^*(s_{11}) + 0.1V^*(s_{21}), \\ & 0.8V^*(s_{21}) + 0.1V^*(s_{12}) + 0.1V^*(s_{11})]. \end{aligned}$$

Q: Solve the Bellman equations efficiently

Q #1: Can we solve the system of Bellman equations in polynomial time?

(A) Yes

(B) No

(C) I don't know

The Bellman equation for $V^*(s_{11})$:

$$\begin{aligned}V^*(s_{11}) = -0.04 + \gamma \max[&0.8V^*(s_{12}) + 0.1V^*(s_{21}) + 0.1V^*(s_{11}),\\ &0.9V^*(s_{11}) + 0.1V^*(s_{12}),\\ &0.9V^*(s_{11}) + 0.1V^*(s_{21}),\\ &0.8V^*(s_{21}) + 0.1V^*(s_{12}) + 0.1V^*(s_{11})].\end{aligned}$$

→ Correct answer is (B) No. The system of Bellman equations is nonlinear because of "max". There is no general technique to solve a nonlinear system of equations efficiently.

Value Iteration

The Bellman equations:

$$V^*(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s').$$

Let $V_i(s)$ be the values for the i^{th} iteration.

1. Start with arbitrary initial values for $V_0(s)$.
2. At the i^{th} iteration, compute $V_{i+1}(s)$ as follows.

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

3. Terminate when $\max_s |V_i(s) - V_{i+1}(s)|$ is small enough.

If we apply the Bellman update infinitely often, the V_i's are guaranteed to converge to the optimal values.

Apply Value Iteration

Let's apply the value iteration algorithm.

Assume that

- the discount factor $\gamma = 1$.
- $R(s) = -0.04, \forall s \neq s_{24}, s \neq s_{34}$.

Start with $V_0(s) = 0, \forall s \neq s_{24}, s \neq s_{34}$.

Note: for terminal states $s_T \in \{s_{24}, s_{34}\}$, $V(s_T) = R(s_T)$.

Q: Calculating $V_1(s_{23})$

#2: What is $V_1(s_{23})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

(A) $(-\infty, 0)$ (B) $[0, 0.25)$ (C) $[0.25, 0.5)$
(D) $[0.5, 0.75)$ (E) $[0.75, 1]$

$V_0(s)$:

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

Q: Calculating $V_1(s_{23})$

#2: What is $V_1(s_{23})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

(A) $(-\infty, 0)$ (B) $[0, 0.25)$ (C) $[0.25, 0.5)$
(D) $[0.5, 0.75)$ (E) $[0.75, 1]$

$V_0(s)$:

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

$\rightarrow$ Correct answer is (A). $V_1(s_{23}) = -0.04$.

Q: Calculating $V_1(s_{33})$

#3: What is $V_1(s_{33})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

(A) 0.26 (B) 0.36 (C) 0.46
(D) 0.56 (E) 0.76

$V_0(s)$:

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

Q: Calculating $V_1(s_{33})$

#3: What is $V_1(s_{33})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

(A) 0.26 (B) 0.36 (C) 0.46
(D) 0.56 (E) 0.76

$V_0(s)$:

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

$\rightarrow$ Correct answer is (A). $V_1(s_{33}) = 0.76$.

The Values of $V_1(s)$

$V_0(s)$:

	1	2	3	4
1	0	0	0	0
2	0	X	0	-1
3	0	0	0	$+1$

$V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

Q: Calculating $V_2(s_{33})$

Q #4: What is $V_2(s_{33})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.822

(B) 0.832

(C) 0.842

(D) 0.852

(E) 0.862

Q: Calculating $V_2(s_{33})$

Q #4: What is $V_2(s_{33})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.822

(B) 0.832

(C) 0.842

(D) 0.852

(E) 0.862

$\rightarrow$ Correct answer is (B).
$V_2(s_{33}) = 0.832$.

Q: Calculating $V_2(s_{23})$

Q #5: What is $V_2(s_{23})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.464

(B) 0.466

(C) 0.468

(D) 0.470

(E) 0.472

Q: Calculating $V_2(s_{23})$

Q #5: What is $V_2(s_{23})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.464

(B) 0.466

(C) 0.468

(D) 0.470

(E) 0.472

$\rightarrow$ Correct answer is (A).
$V_2(s_{23}) = 0.464.$

Q: Calculating $V_2(s_{32})$

Q #6: What is $V_2(s_{32})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.16

(B) 0.36

(C) 0.56

(D) 0.76

(E) 0.96

Q: Calculating $V_2(s_{32})$

Q #6: What is $V_2(s_{32})$?

$$V_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

Here is $V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

(A) 0.16

(B) 0.36

(C) 0.56

(D) 0.76

(E) 0.96

$\rightarrow$ Correct answer is (C).
$V_2(s_{32}) = 0.56$.

The Values of $V_2(s)$

$V_1(s)$:

	1	2	3	4
1	-0.04	-0.04	-0.04	-0.04
2	-0.04	X	-0.04	-1
3	-0.04	-0.04	0.76	$+1$

$V_2(s)$:

	1	2	3	4
1	-0.08	-0.08	-0.08	-0.08
2	-0.08	X	0.464	-1
3	-0.08	0.56	0.832	$+1$

Observations from Value Iteration

Each state accumulates negative rewards
until the algorithm finds a path to the $+1$ goal state.

How should we update $V^*(s)$ for all states s?

- synchronously: store and use $V_i(s)$ to calculate $V_{i+1}(s)$.
- asynchronously: stores $V_i(s)$ and update the values one at a time, in any order.

Policy Iteration

- Deriving the optimal policy does not require accurate estimates of the utility function $(V^*(s))$.

  → If one action is clearly better than all others, then the exact magnitude of the utilities on the states involved need not be precise.

- **Policy iteration** alternates between two steps.

  1. **Policy evaluation:** Given a policy π_i, calculate $V^{\pi_i}(s)$, which is the utility of each state if π_i were to be executed.
  2. **Policy improvement:** Calculate a new policy π_{i+1} using V^{π_i}.

  Terminates when there is no change in the policy.

  → Must terminate because there are finitely many policies for a finite state space and each iteration yields a better policy.

Policy Iteration

- Policy evaluation:

$$V^{\pi_i}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) V^{\pi_i}(s').$$

- Policy improvement:

$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} P(s'|s, a) V^{\pi_i}(s').$$

Policy Evaluation v.s. Bellman Equations

Policy evaluation:

$$V(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) V(s').$$

Bellman equations:

$$V(s) = R(s) + \gamma \max_{a} \sum_{s'} P(s'|s, a) V(s').$$

Write down both equations for $V(s_{11})$.
Assume that $\pi(s_{11}) =$ down.

Performing Policy Evaluation Exactly

Policy evaluation:

$$V(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) V(s').$$

We could solve the system of linear equations exactly using standard linear algebra techniques.

For n states, this will take $O(n^3)$ time...

Performing Policy Evaluation Iteratively

Policy evaluation:

$$V(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) V(s').$$

Solve the system of linear equations approximately
by performing a number of simplified value iteration steps:

Repeat for $j \in \{1, 2, \ldots, m\}$:

$$V_{j+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) V_j(s').$$

Policy Iteration: An Example

Apply policy iteration for the simple grid environment below. Use iteration for policy evaluation with $m = 1$. s_{12} and s_{22} are terminal states.

-0.04	+1
-0.04	-1

$\mathcal{A} = \{up, right, down, left\}$.

The initial policy is $\pi_1(s) = right$, $\forall s \in \mathcal{S}$.

The agent moves towards, to the right of, or to the left of the intended direction with probabilities 0.8, 0.1, and 0.1 respectively.

Let $\gamma = 1$.

Revisiting Learning Goals

- Trace the execution of and implement the value iteration algorithm for solving a Markov Decision Process.
- Trace the execution of and implement the policy iteration algorithm for solving a Markov Decision Process.