Polices & Value Functions
Lecture 3

Alina Vereshchaka

CSE4/510 Reinforcement Learning
Fall 2019

avereshc@buffalo.edu

September 3, 2019
Overview

1. Recap
2. Policies
3. Reward and Return
4. Value Functions
5. Bellman Equation
1 Recap

2 Policies

3 Reward and Return

4 Value Functions

5 Bellman Equation
Recap: MDP

A Markov Decision Process (MDP) is a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. It is a sequential decision-making problem where the decision maker (agent) interacts with an environment, taking actions that affect the state of the environment, and receives rewards based on the current state and action taken.

The basic components of an MDP are:
- **States** ($S_t$): The current state of the environment.
- **Actions** ($A_t$): The actions that the agent can take at each time step.
- **Rewards** ($R_t$): The immediate reward received by the agent for taking an action in a particular state.
- **Transitions** ($P_{S_{t+1} | S_t, A_t}$): The probability of moving from one state to another given an action.

The goal of an agent in an MDP is to learn a policy that maximizes the expected cumulative reward over time.
Recap: MDP

At each step $t$ the agent:

- Receives state $S_t$ / observation $O_t$ and reward $R_t$
- Executes action $A_t$

The environment:

- Receives action $A_t$
- Emits state $S_{t+1}$ / observation $O_{t+1}$ and reward $R_{t+1}$
Recap: MDP Definitions

Definition

Markov decision process (MDP) defined by the tuple \( \langle s, a, O, P, r, \rho_0, \gamma \rangle \), where:

- \( s \in S \) denotes states, describing all possible configurations;
- \( a \in A \) denotes actions;
- \( P : S \times A \times S \to \mathbb{R} \) is the states transition probability distribution;
- \( O \) is a set of observations;
- \( r : S \to \mathbb{R} \) is the reward function;
- \( \rho_0 : S \to [0, 1] \) is the distribution of the initial state \( s_0 \);
- \( \gamma \in [0, 1] \) is a discount factor.
Recap
Recap: Return $G_t$

**Definition**

The *return* $G_t$ is the total discounted reward from time-step $t$.

$$G_t = R_t = r_{t+1} + \gamma r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$  \hspace{1cm} (1)

- $\gamma$ is a discount factor ($\gamma \in [0, 1)$)
- $r$ is the immediate reward, $R$ is the cumulative reward
- The value of receiving reward $R$ after $k + 1$ time-steps is $\gamma^k R$
Example: Return $G_t$

**Definition**

The *return* $G_t$ is the total discounted reward from time-step $t$.

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

Suppose $\gamma = 0.5$ and the following sequence of rewards is received $R_1 = 1, R_2 = 8, R_3 = 4$, with $T = 3$. What is the return $G_0, G_1, G_2, G_3$?
A policy $\pi$ is a distribution over actions given states. It defines the agent’s behaviour. It can be either deterministic or stochastic:

- Deterministic: $\pi(s) = a$
- Stochastic: $\pi(a|s) = P_{\pi}[A = a|S = s]$

- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
Notations: Transition Probability

\[ P_{s,s'}^a = P(S_{t+1} = s' | S_t = s, A_t = a) \]

\( P \) is the transition probability. If we start at state \( s \) and take action \( a \) and we end up in state \( s' \) with probability \( P_{ss'}^a \).
Notations: Transition Probability

$$P_{s,s'}^a = P(S_{t+1} = s' | S_t = s, A_t = a)$$

$P$ is the transition probability. If we start at state $s$ and take action $a$ and we end up in state $s'$ with probability $P_{s,s'}^a$.

$$R_{s,s'}^a = \mathbb{E}[R_{t+1} | S_t = s, S_{t+1} = s', A_t = a]$$

$R_{s,s'}^a$ is another way of writing the expected (or mean) reward that we receive when starting in state $s$, taking action $a$, and moving into state $s'$. 
Example: Recycling Robot

| $s$   | $a$    | $s'$   | $p(s'|s,a)$ | $r(s,a,s')$ |
|-------|--------|--------|-------------|-------------|
| high  | search | high   | $\alpha$    | $r_{\text{search}}$ |
| high  | search | low    | $1 - \alpha$| $r_{\text{search}}$ |
| low   | search | high   | $1 - \beta$ | $r_{\text{search}}$ |
| low   | search | low    | $\beta$     | $r_{\text{search}}$ |
| high  | wait   | high   | 1            | $r_{\text{wait}}$ |
| high  | wait   | low    | 0            | -            |
| low   | wait   | high   | 0            | -            |
| low   | wait   | low    | 1            | $r_{\text{wait}}$ |
| low   | recharge | high    | 1            | 0            |
| low   | recharge | low    | 0            | -            |

Diagram:

- States: high, low
- Actions: search, wait, recharge
- Transitions:
  - $p(\text{high}|\text{high}, \text{search}) = \alpha$
  - $p(\text{low}|\text{high}, \text{search}) = 1 - \alpha$
  - $p(\text{high}|\text{low}, \text{search}) = 1 - \beta$
  - $p(\text{low}|\text{low}, \text{search}) = \beta$
  - $p(\text{high}|\text{high}, \text{wait}) = 0$
  - $p(\text{low}|\text{high}, \text{wait}) = 0$
  - $p(\text{low}|\text{low}, \text{wait}) = 1$
  - $p(\text{high}|\text{low}, \text{wait}) = 1$
  - $p(\text{high}|\text{high}, \text{recharge}) = 1$
  - $p(\text{low}|\text{high}, \text{recharge}) = \alpha$
  - $p(\text{low}|\text{low}, \text{recharge}) = 1 - \alpha$
  - $p(\text{low}|\text{low}, \text{recharge}) = 1 - \beta$
  - $p(\text{low}|\text{high}, \text{recharge}) = 1$
  - $p(\text{low}|\text{low}, \text{recharge}) = 0$

Rewards:

- $r_{\text{search}}$ for transitioning to search
- $r_{\text{wait}}$ for transitioning to wait
- $r_{\text{recharge}}$ for transitioning to recharge
- $-3$ for transitioning to a state with no available action
RL agents learn to **maximize discounted cumulative future reward** ($R$).
Cumulative reward:

\[ R_t = r_{t+1} + r_{t+2} + r_{t+3} + r_{t+4} + \cdots = \sum_{k=0}^{\infty} r_{t+k+1} \]
Reward \((r_{t+1})\) and Return \((G_t = R_t)\)

Cumulative reward:

\[
R_t = r_{t+1} + r_{t+2} + r_{t+3} + r_{t+4} + \cdots = \sum_{k=0}^{\infty} r_{t+k+1}
\]

Discounted cumulative reward:

\[
R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}
\]

where \(0 \leq \gamma \leq 1\)
Discounted cumulative reward:

\[ R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \]

where \( 0 \leq \gamma \leq 1 \)

Example. Suppose \( \gamma = 0.5 \) and the following sequence of rewards is received
\( r_1 = 1, r_2 = 8, r_3 = 4, \) with \( T = 3 \). What is the return \( R_0, R_1, R_2, R_3 \)?
## Table of Contents

1. Recap
2. Policies
3. Reward and Return
4. Value Functions
5. Bellman Equation
There are two types of value functions:

- state value function $V(s)$
There are two types of value functions:

- state value function $V(s)$
- action value function $Q(s, a)$
State value function $V(s)$

**Definition**

*State value function* describes the value of a state when following a policy. It is the expected return when starting from state $s$ acting according to our policy $\pi$:

$$V^\pi(s) = \mathbb{E}_{\pi}[R_t | S_t = s]$$
State value function

$V^\pi(s)$ can also be interpreted, as the \textbf{expected cumulative future discounted reward}, where

- "Expected" refers to the "expected value"
- "Cumulative" refers to the summation
- "Future" refers to the fact that it’s an expected value of a future quantity with respect to the present quantity, i.e. $s_t = s$.
- "Discounted" refers to the "gamma" factor, which is a way to adjust the importance of how much we value rewards at future time steps, i.e. starting from $t + 1$.
- "Reward" refers to the main quantity of interested, i.e. the reward received from the environment.
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$$V^*(1, 4) =$$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$$V^*(1, 4) = 1$$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$V^*(1, 4) = 1$

$V^*(1, 3) =$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

\[ V^*(1, 4) = 1 \]
\[ V^*(1, 3) = 1 \]
Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

\begin{align*}
V^*(1, 4) &= 1 \\
V^*(1, 3) &= 1 \\
V^*(1, 2) &=
\end{align*}
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

\[ V^*(1, 4) = 1 \]
\[ V^*(1, 3) = 1 \]
\[ V^*(1, 2) = 1 \]
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$V^*(1, 4) = 1$
$V^*(1, 3) = 1$
$V^*(1, 2) = 1$
$V^*(1, 1) =$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$V^*(1, 4) = 1$
$V^*(1, 3) = 1$
$V^*(1, 2) = 1$
$V^*(1, 1) = 1$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

$V^*(1, 4) = 1$
$V^*(1, 3) = 1$
$V^*(1, 2) = 1$
$V^*(1, 1) = 1$
$V^*(2, 3) =$
Example: State value function

Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

- $V^*(1, 4) = 1$
- $V^*(1, 3) = 1$
- $V^*(1, 2) = 1$
- $V^*(1, 1) = 1$
- $V^*(2, 3) = 1$
Assume: actions deterministically successful. $\gamma = 1$. Agent is following an optimal policy.

- $V^*(1, 4) = 1$
- $V^*(1, 3) = 1$
- $V^*(1, 2) = 1$
- $V^*(1, 1) = 1$
- $V^*(2, 3) = 1$
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$$V^*(1, 4) =$$
Example: State value function

Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

\[ V^*(1, 4) = 1 \]
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$V^*(1, 4) = 1$

$V^*(1, 3) =$
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$V^*(1, 4) = 1$

$V^*(1, 3) = 0.9$
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

\[
\begin{align*}
V^*(1, 4) &= 1 \\
V^*(1, 3) &= 0.9 \\
V^*(1, 2) &= \\
\end{align*}
\]
Assume: actions deterministically successful. \( \gamma = 0.9 \). Agent is following an optimal policy.

\[
\begin{align*}
V^*(1, 4) &= 1 \\
V^*(1, 3) &= 0.9 \\
V^*(1, 2) &= 0.81
\end{align*}
\]
Example: State value function

Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

\[
V^*(1, 4) = 1 \\
V^*(1, 3) = 0.9 \\
V^*(1, 2) = 0.81 \\
V^*(1, 1) =
\]
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$V^*(1, 4) = 1$

$V^*(1, 3) = 0.9$

$V^*(1, 2) = 0.81$

$V^*(1, 1) = 0.729$
Example: State value function

Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$V^*(1, 4) = 1$
$V^*(1, 3) = 0.9$
$V^*(1, 2) = 0.81$
$V^*(1, 1) = 0.729$
$V^*(2, 3) =$
Example: State value function

Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

\[
\begin{align*}
V^*(1, 4) &= 1 \\
V^*(1, 3) &= 0.9 \\
V^*(1, 2) &= 0.81 \\
V^*(1, 1) &= 0.729 \\
V^*(2, 3) &= 0.81
\end{align*}
\]
Assume: actions deterministically successful. $\gamma = 0.9$. Agent is following an optimal policy.

$$V^*(1, 4) = 1$$
$$V^*(1, 3) = 0.9$$
$$V^*(1, 2) = 0.81$$
$$V^*(1, 1) = 0.729$$
$$V^*(2, 3) = 0.81$$
**Action value function**

**Definition**

*Action value function* tells us the value of taking an action $a$ in state $s$ when following a certain policy $\pi$. It is the expected return given the state and action under $\pi$:

$$Q^\pi(s, a) = \mathbb{E}_\pi[R_t | S_t = s, A_t = a]$$
Richard Bellman was an American applied mathematician who derived the following equations which allow us to start solving these MDPs. The Bellman equations are ubiquitous in RL and are necessary to understand how RL algorithms work.
Bellman equation for the state value function

\[ V^\pi(s) = \mathbb{E}_\pi[R_t | S_t = s] \]
Bellman equation for the state value function

\[
V^\pi(s) = \mathbb{E}_\pi[R_t | S_t = s] \\
= \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \ldots | S_t = s]
\]
Bellman equation for the state value function

\[ V^\pi(s) = \mathbb{E}_\pi[R_t | S_t = s] \]
\[ = \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \ldots | S_t = s] \]
\[ = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s] \]
Bellman equation for the state value function

\[ V^\pi(s) = \mathbb{E}_\pi[R_t | S_t = s] \]

\[ = \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \ldots | S_t = s] \]

\[ = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s] \]

\[ = \mathbb{E}_\pi[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | S_t = s] \]
The expectation here describes what we expect the return to be if we continue from state $s$ following policy $\pi$. The expectation can be written explicitly by summing over all possible actions and all possible returned states.

$$\mathbb{E}_\pi[r_{t+1}|s_t = s] = \sum_a \pi(s, a) \sum_{s'} P_{ss'} R_{ss'}^a$$

$$\mathbb{E}_\pi[\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_t = s] = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \gamma \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_{t+1} = s']$$
By distributing the expectation between these two parts, we can then manipulate our equation into the form:

\[
V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P^a_{ss'} \left[ R^a_{ss'} + \gamma \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s' \right] \right]
\]
Bellman equation for the state value function

By distributing the expectation between these two parts, we can then manipulate our equation into the form:

\[
V_\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \left[ R_{ss'}^a + \gamma \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s' \right] \right]
\]

\[
= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \left[ R_{ss'}^a + \gamma V_\pi(s') \right]
\]
Bellman equations is that they let us express values of states as values of other states. This means that if we know the value of $s_{t+1}$, we can very easily calculate the value of $s_t$.

Bellman equations is a foundation for iterative approaches to solve reinforcement learning task.