ECE 1508: Reinforcement Learning

Chapter 2: Model-based RL

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Fall 2025

Classical RL Methods: *Recall*

Ultimate goal in an RL problem is to find the optimal policy

As mentioned, we have *two major challenges* in this way

1. *We need to compute values explicitly*
2. *We often deal with settings with huge state spaces?*

In this part of the course, we are going to handle the first challenge

- *This chapter $\rightsquigarrow$ Model-based methods*
- *Next chapter $\rightsquigarrow$ Model-free methods*

A Good Start Point: *Model-based RL*

In a nutshell, in model-based methods

> *we are able to describe mathematically the behavior of environment*

This might come from the *nature of problem* or *simply postulated by us*

Model-based RL
- Bellman Equation
  - value iteration
  - policy iteration

Model-free RL
- on-policy methods
  - temporal difference
  - Monte Carlo
  - SARSA
- off-policy methods
  - Q-learning

© A. Bereyhi 2024 - 2025

Complete State is *Markov Process*

When we formulated the RL framework, we stated that

a complete state must describe a Markov process

Markov Process

Sequence $S_1 \to S_2 \to \dots$ *describe a Markov process if*

$$\Pr\{S_{t+1} = s_{t+1} | S_t = s_t, \dots, S_1 = s_1\} = \Pr\{S_{t+1} = s_{t+1} | S_t = s_t\}$$

Following this fact, we introduced the concepts of

rewarding and transition functions

Recall: *Transition and Rewarding*

Both these mappings only depend on current state and action

Transition function maps state S_t and action A_t to the next state S_{t+1}

$$\mathcal{P}(\cdot) : \mathbb{S} \times \mathbb{A} \mapsto \mathbb{S}$$

Rewarding function maps state S_t and action A_t to reward R_{t+1}

$$\mathcal{R}(\cdot) : \mathbb{S} \times \mathbb{A} \mapsto \left\{ r^1, \dots, r^L \right\}$$

We said that *these mappings are in general random*

Describing Markov Trajectory

Markovity of the state indicates that we observe the following trajectory

$$S_0, A_0 \to (R_1, S_1), A_1 \to \dots \to (R_t, S_t), A_t \to (R_{t+1}, S_{t+1})$$

This trajectory describes a Markov process with conditional distribution

$$\begin{aligned} p(r, \bar{s}|s, a) &= \Pr\{R_{t+1} = r, S_{t+1} = \bar{s}|S_t = s, A_t = a\} \\ &= \Pr\{R_t = r, S_t = \bar{s}|S_{t-1} = s, A_{t-1} = a\} \\ &\vdots \\ &= \Pr\{R_1 = r, S_1 = \bar{s}|S_0 = s, A_0 = a\} \end{aligned}$$

The above trajectory describes a Markov Decision Process (MDP)

Finite MDPs

In this course, we focus on *finite* MDPs

Finite MDP

The Markov process

$$S_0, A_0 \rightarrow (R_1, S_1), A_1 \rightarrow \dots \rightarrow (R_t, S_t), A_t$$

is a finite MDP if rewards, actions and states belong to a finite set, i.e.,

$$r \in \{r^1, \dots, r^L\} \qquad a \in \{a^1, \dots, a^M\} \qquad s \in \{s^1, \dots, s^N\}$$

MDPs are completely described by conditional distribution $p(r, \bar{s}|s, a)$

We call $p(r, \bar{s}|s, a)$ *hereafter rewarding-transition model*

Model-based RL via *MDP*

+ *What makes it now model-based RL?*
– We assume that *rewarding-transition model* $p(r, \bar{s}|s, a)$ is given to us
+ *But you said for model-based RL, we should know the transition and rewarding functions!*
– Well, we can describe them using $p(r, \bar{s}|s, a)$!

Rewarding Model

Assume we are in state $S_t = s$ and act $A_t = a$; then, R_{t+1} is a random variable whose distribution is given by

$$p(r|s, a) = \sum_{n=1}^{N} p(r, s^n|s, a)$$

We call this distribution hereafter rewarding model

Model-based RL via *MDP*

Similarly, we can describe the *transition function*

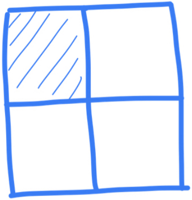
Transition Model

Assume we are in state $S_t = s$ and act $A_t = a$; then, next state S_{t+1} is a random variable whose distribution is given by

$$p\left(\bar{s}|s, a\right) = \sum_{\ell=1}^{L} p\left(r^{\ell}, \bar{s}|s, a\right)$$

We call this distribution hereafter transition model

Example: *Dummy Grid World*



We have a grid board where at each cell we can move

$$\mathbb{A} = \{0 \equiv \texttt{left}, 1 \equiv \texttt{down}, 2 \equiv \texttt{right}, 3 \equiv \texttt{up}\}$$

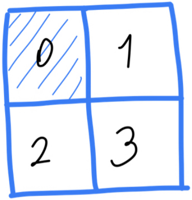
Our ultimate goal is to arrive at top-left corner

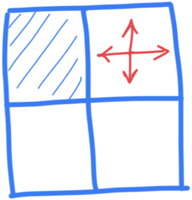
through shortest path

This problem describes an MDP with deterministic rewarding-transition model

- *State is the cell index*
- *Action is the direction we move*
- *Reward is -1 each time we move until we get to destination*
  - ↳ *Reward is -0.5 when we hit the corners*

Example: *Dummy Grid World*





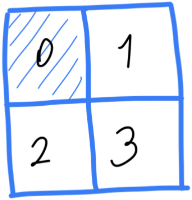
$\mathbb{S} = \{0, 1, 2, 3\}$

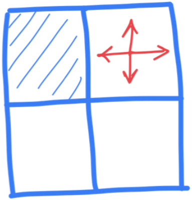
$\mathbb{A} = \{0 \equiv \texttt{left}, 1 \equiv \texttt{down}, 2 \equiv \texttt{right}, 3 \equiv \texttt{up}\}$

Let's write the rewarding-transition model down

$$p\left(r, \bar{s} | 3, 3\right) = \begin{cases} 1 & (r, \bar{s}) = (-1, 1) \\ 0 & (r, \bar{s}) \neq (-1, 1) \end{cases}$$

Example: *Dummy Grid World*





$\mathbb{S} = \{0, 1, 2, 3\}$

$\mathbb{A} = \{0 \equiv \texttt{left}, 1 \equiv \texttt{down}, 2 \equiv \texttt{right}, 3 \equiv \texttt{up}\}$

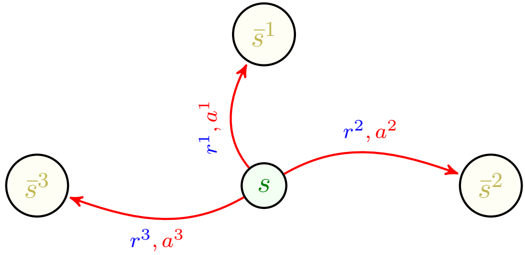
Let's write the rewarding-transition model down

$$p\left(r, \bar{s} | 0, a\right) = \begin{cases} 1 & (r, \bar{s}) = (0, 0) \\ 0 & (r, \bar{s}) \neq (0, 0) \end{cases}$$

$\rightsquigarrow$ $s = 0$ is terminal state

Transition Diagram

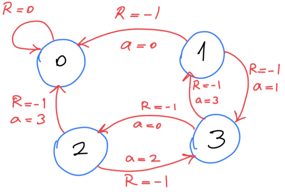
It is sometimes helpful to show transition model via a *transition diagram*



This diagram describes a graph

- *Each node is a possible state: we have in total N nodes*
- *Node s is connected to $\bar{s}$ if the probability of transition is non-zero*
  - ↳ *We could specify the action that can lead us to the new state*
  - ↳ *The graph could have loops including self-loop*

Transition Diagram: *Dummy Grid World*



In our dummy grid world, we have four states

- *If we are in terminal state we always remain there with no rewards*
- *From state $s = 1$ we can go to states $\bar{s} = 0, 3$ depending on action*
  - ↳ *We can also remain in state $s = 1$ and reward with -0.5 if we hit corners*
- *From state $s = 2$ we can go to states $\bar{s} = 0, 3$ depending on action*
  - ↳ *We can also remain in state $s = 1$ and reward with -0.5 if we hit corners*
- *From state $s = 3$ we can go to states $\bar{s} = 1, 2$ depending on action*
  - ↳ *We can also remain in state $s = 1$ and reward with -0.5 if we hit corners*

Expected Action Reward

As we said, *using rewarding-transition model we can describe the environment completely: for instance, let's see what would be the expected immediate reward that we get if in state s we act a*

$$\bar{\mathcal{R}}\left(s,a\right) = \mathbb{E}\left\{R_{t+1}\middle|s,a\right\} \rightsquigarrow \text{we simplify notation } S_t = s \text{ to } s$$

$$= \sum_{\ell=1}^{L} r^\ell p\left(r^\ell|s,a\right)$$

$$= \sum_{\ell=1}^{L} r^\ell \sum_{n=1}^{N} p\left(r^\ell, s^n|s,a\right)$$

$$= \sum_{\ell=1}^{L}\sum_{n=1}^{N} r^\ell p\left(r^\ell, s^n|s,a\right) \rightsquigarrow \text{rewarding-transition model}$$

Expected Action Reward

$\bar{\mathcal{R}}(s, a)$ describes

the reward we expect to see immediately after acting a in state s

We are going to see this expectation a lot, so maybe we could give it a name

Expected Action Reward

The expected reward for a state-action pair (s, a) is defined as

$$\bar{\mathcal{R}}(s, a) = \mathbb{E}\{R_{t+1} | s, a\} = \sum_{\ell=1}^{L} \sum_{n=1}^{N} r^{\ell} p\left(r^{\ell}, s^{n} | s, a\right)$$

Obviously, $\bar{\mathcal{R}}(s, a)$ does not depend on policy

Expected Policy Reward

+ *Can we relate it also to our policy?*
– Sure! We could average over our policy

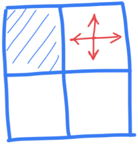
Expected Policy Reward

The expected immediate reward of policy π at state s is defined as

$$\bar{\mathcal{R}}_\pi(s) = \mathbb{E}_\pi\{R_{t+1}|s\} = \sum_{m=1}^{M} \mathbb{E}\{R_{t+1}|s, a^m\}\,\pi(a^m|s)$$
$$= \sum_{m=1}^{M}\sum_{\ell=1}^{L}\sum_{n=1}^{N} r^\ell p\left(r^\ell, s^n|s, a\right)\pi(a^m|s)$$

It describes reward we expect to see immediately after state s while playing π

Example: *Dummy Grid World*



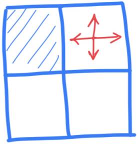
In our dummy grid world, we can easily compute the expected immediate reward

$$\bar{\mathcal{R}}(1, a) = \begin{cases} -1 & a \in \{0, 1\} \\ -0.5 & a \in \{2, 3\} \end{cases}$$

Obviously in terminal state we always get zero expected reward, e.g., for all a

$$\bar{\mathcal{R}}(0, a) = 0$$

Example: *Dummy Grid World*



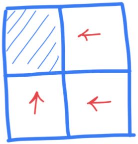
Now assume that we play uniformly at random, i.e., for all a and s

$$\pi(a|s) = \frac{1}{4}$$

In this case the expected policy reward is

$$\bar{\mathcal{R}}_\pi(1) = \sum_{a=0}^{3} \bar{\mathcal{R}}(1, a)\, \pi(a|1) = -0.75$$

Example: *Dummy Grid World*



But if we change to above deterministic policy: the expected reward changes to

$$\bar{\mathcal{R}}_\pi(1) = \sum_{a=0}^{3} \bar{\mathcal{R}}(1, a)\, \pi(a|1) = \bar{\mathcal{R}}(1, 0) = -1$$

and we can easily show that

$$\bar{\mathcal{R}}_\pi(0) = 0 \qquad \bar{\mathcal{R}}_\pi(2) = -1 \qquad \bar{\mathcal{R}}_\pi(3) = -1$$

Computing Value Functions: *Naive Approach*

Now that we have a concrete model for our environment: *we should go ahead and compute the value function, as we want to optimize it*

Let's start with direct computation

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\{G_t | s\} \\ &= \mathbb{E}_\pi\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s\} \\ &= \mathbb{E}_\pi\{R_{t+1} | s\} + \gamma \mathbb{E}_\pi\{R_{t+2} | s\} + \gamma^2 \mathbb{E}_\pi\{R_{t+3} | s\} + \dots \\ &= \bar{\mathcal{R}}_\pi(s) + \gamma \mathbb{E}_\pi\{R_{t+2} | s\} + \gamma^2 \mathbb{E}_\pi\{R_{t+3} | s\} + \dots \end{aligned}$$

+ *How can we compute next terms?*
– *We could use the rewarding-transition model of MDP*

Computing Value Functions: *Naive Approach*

Let's try the second term for example: we first define the notation

$$\mathbb{E}_\pi \left\{ R_{t+2} | s, s^n, a^m, a^j \right\} = \mathbb{E}_\pi \left\{ R_{t+2} | S_t = s, S_{t+1} = s^n, A_t = a^m, A_{t+1} = a^j \right\}$$

We can easily compute $\mathbb{E}_\pi \left\{ R_{t+2} | s, s^n, a^m, a^j \right\}$ *as*

$$\begin{aligned} \mathbb{E}_\pi \left\{ R_{t+2} | s, s^n, a^m, a^j \right\} &= \sum_{\ell=1}^{L} r^\ell p \left(r^\ell | s, s^n, a^m, a^j \right) \\ &= \sum_{\ell=1}^{L} r^\ell p \left(r^\ell | s^n, a^j \right) \end{aligned}$$

Computing Value Functions: *Naive Approach*

We can then say that

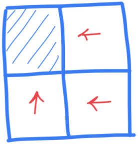
$$\mathbb{E}_\pi \left\{ R_{t+2} | s \right\} = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{M} \mathbb{E}_\pi \left\{ R_{t+2} | s, s^n, a^m, a^j \right\} p \left(a^m, s^n, a^j | s \right)$$

and write down $p \left(a^m, s^n, a^j | s \right)$ *using chain rule*

$$\begin{aligned} p \left(a^m, s^n, a^j | s \right) &= p \left(a^m | s \right) p \left(s^n | s, a^m \right) p \left(a^j | s, a^m, s^n \right) \\ &= \pi \left(a^m | s \right) \underbrace{p \left(s^n | s, a^m \right)}_{\text{transition model}} \pi \left(a^j | s^n \right) \end{aligned}$$

+ *How can we compute the next term?*
– We should repeat the same approach: *there will be more nested sums*

Example: *Dummy Grid World*



Let's start with the above policy: π^1

$$v_{\pi^1}(1) = \mathbb{E}_{\pi^1}\left\{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | 1\right\}$$

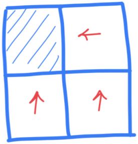
Following policy at $s = 1$ *we end up at terminal state at next time*

$$v_{\pi^1}(1) = \mathbb{E}_{\pi^1}\left\{R_{t+1} + \gamma 0 + \gamma^2 0 + \ldots | 1\right\} = \bar{\mathcal{R}}_{\pi^1}(1) = -1$$

Same way, we can conclude that

$$v_{\pi^1}(0) = 0 \qquad v_{\pi^1}(2) = -1 \qquad v_{\pi^1}(3) = -2$$

Example: *Dummy Grid World*

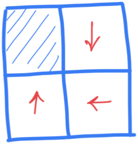


Let's change to policy π^2: we could follow same steps to show that

$$v_{\pi^2}(0) = 0 \qquad v_{\pi^2}(1) = -1 \qquad v_{\pi^2}(2) = -1 \qquad v_{\pi^2}(3) = -2$$

We note that it returns the same values as policy π^1

Example: *Dummy Grid World*



Let's now look at policy π^3: we could follow same steps to show that

$$v_{\pi^3}(0) = 0 \qquad v_{\pi^3}(1) = -3 \qquad v_{\pi^3}(2) = -1 \qquad v_{\pi^3}(3) = -2$$

We can see that

$$\pi^1 = \pi^2 \geqslant \pi^3$$

Computing Value Functions: *Practical Approach*

+ *But, we should compute infinite terms in general!*
- Well, if we are lucky: *the sequence either terminates or shows a pattern*
+ *What if that doesn't happen?*
- Then, this approach really does not work!

This is why we called it the naive approach, since we never use this approach: in practice, we always invoke

Bellman equation

and find the value via dynamic programming

Future Return: *Recursive Property*

Even though future return looks infinte, it has a simple recursive property

$$
\begin{aligned}
G_t &= \sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \\
&= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
&= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \dots \right) \\
&= R_{t+1} + \gamma G_{t+1}
\end{aligned}
$$

We can use this property to find a fixed-point equation for the value function!

Value Function: *Recursive Property*

Say we are playing with policy π: *we can write the value function as*

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\{G_t | s\} \\ &= \mathbb{E}_\pi\{R_{t+1} + \gamma G_{t+1} | s\} \\ &= \mathbb{E}_\pi\{R_{t+1} | s\} + \gamma \mathbb{E}_\pi\{G_{t+1} | s\} \\ &= \bar{\mathcal{R}}_\pi(s) + \gamma \underbrace{\mathbb{E}_\pi\{G_{t+1} | s\}}_{?} \end{aligned}$$

+ *Isn't that term again the value function at s?*
– Be careful! It's not

Attention

The second term is not the value of state s

$$\mathbb{E}_\pi\{G_{t+1} | s\} = \mathbb{E}_\pi\{G_{t+1} | S_t = s\} \neq \mathbb{E}_\pi\{G_t | S_t = s\} = v_\pi(s)$$

Value Function: *Recursive Property*

Let's do some marginalization

$$\mathbb{E}_\pi \left\{ G_{t+1} | s \right\} = \sum_{n=1}^{N} \mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n \right\} \Pr \left\{ S_{t+1} = s^n | S_t = s \right\}$$

$$= \sum_{n=1}^{N} \mathbb{E}_\pi \left\{ G_{t+1} | s, s^n \right\} p \left(s^n | s \right)$$

Well, we need to specify the two terms in under summation, i.e.,

- $\mathbb{E}_\pi \left\{ G_{t+1} | s, s^n \right\}$
- $p \left(s^n | s \right) = \Pr \left\{ S_{t+1} = s^n | S_t = s \right\}$

Value Function: *Recursive Property*

Recall the trajectory

$$S_0, A_0 \to (R_1, S_1), A_1 \to \dots \to (R_{t+1}, S_{t+1}), A_{t+1} \to (R_{t+2}, S_{t+2})$$

If we know state S_{t+1} any reward after $t+1$ only depends on S_{t+1}, i.e.,

$$\mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n \right\} = \mathbb{E}_\pi \left\{ G_{t+1} | S_{t+1} = s^n \right\}$$

This indicates that

$$\mathbb{E}_\pi \left\{ G_{t+1} | s, s^n \right\} = v_\pi \left(s^n \right)$$

i.e., the value function at state s^n

Value Function: *Recursive Property*

We can further find $p(s^n|s)$ from transition model and policy

$$
\begin{aligned}
p_\pi(s^n|s) &= \sum_{m=1}^{M} p(s^n, a^m|s) \\
&= \sum_{m=1}^{M} p(a^m|s)\, p(s^n|a^m, s) \\
&= \sum_{m=1}^{M} \pi(a^m|s) p(s^n|s, a^m)
\end{aligned}
$$

$\rightsquigarrow$ depends on policy

We know have both terms in terms of transition model and policy

Value Function: *Recursive Property*

Replacing into the equation, where we left we have

$$\begin{aligned}\mathbb{E}_\pi\left\{G_{t+1}|s\right\} &= \sum_{n=1}^{N} \mathbb{E}_\pi\left\{G_{t+1}|s, s^n\right\} p\left(s^n|s\right)\\ &= \sum_{n=1}^{N} v_\pi\left(s^n\right) p_\pi\left(s^n|s\right)\\ &= \sum_{n=1}^{N}\sum_{m=1}^{M} v_\pi\left(s^n\right) p\left(s^n|s, a^m\right) \pi\left(a^m|s\right)\end{aligned}$$

We can also present it by shorter notation as

$$\mathbb{E}_\pi\left\{G_{t+1}|s\right\} = \mathbb{E}_\pi\left\{v_\pi\left(S_{t+1}\right)|s\right\}$$

Value Function: *Recursive Property*

Back to computation of value function, we have

$$\begin{aligned} v_\pi(s) &= \bar{\mathcal{R}}_\pi(s) + \gamma \mathbb{E}_\pi\left\{G_{t+1} | s\right\} \\ &= \bar{\mathcal{R}}_\pi(s) + \gamma \mathbb{E}_\pi\left\{v_\pi(S_{t+1}) | s\right\} \\ &= \bar{\mathcal{R}}_\pi(s) + \gamma \sum_{n=1}^{N} v_\pi(s^n) p_\pi(s^n | s) \end{aligned}$$

This is a recursive equation that relates value of one state to other values

which is a Bellman equation

Bellman Equation: *Value*

Bellman Equation for Value Function

For any policy π the value function at each state s satisfies

$$v_\pi(s) = \bar{\mathcal{R}}_\pi(s) + \gamma \sum_{n=1}^{N} v_\pi(s^n)\, p_\pi(s^n | s)$$

+ *Well! What is the use of Bellman equation?*
– It describes a *fixed-point equation that can be solved for $v_\pi(s)$!*

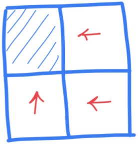
Bellman Equation: *Breaking Down*

$$v_\pi(s) = \bar{\mathcal{R}}_\pi(s) + \gamma \sum_{n=1}^{N} v_\pi(s^n) p_\pi(s^n|s)$$

In general, we have N possible state $\rightsquigarrow$ *we have N possible values*

- *Bellman equation relates each value to other $N-1$ values*
  - ↳ *For each s, Bellman equation has N unknowns $v_\pi(s^1), \ldots, v_\pi(s^N)$*
- *We can write the Bellman equation for all N states*
  - ↳ *We have N equations each with N unknowns*
- *We solve this system of equations for unknowns $v_\pi(s^1), \ldots, v_\pi(s^N)$*

Example: *Dummy Grid World*



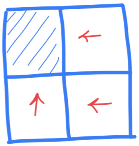
Let's try with our dummy grid world: we saw that

$$\bar{\mathcal{R}}_\pi(0) = 0 \qquad \bar{\mathcal{R}}_\pi(1) = -1 \qquad \bar{\mathcal{R}}_\pi(2) = -1 \qquad \bar{\mathcal{R}}_\pi(3) = -1$$

Now let's consider the values unknown

$$v_\pi(0), v_\pi(1), v_\pi(2), v_\pi(3)$$

Example: *Dummy Grid World*



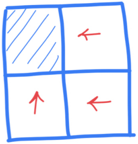
We set $\gamma = 1$ *and start with state* $s = 0$

$$v_\pi(0) = \bar{\mathcal{R}}_\pi(0) + \sum_{\bar{s}=0}^{3} v_\pi(\bar{s}) \, p_\pi(\bar{s}|0)$$

We know that

$$p_\pi(\bar{s}|0) = \begin{cases} 1 & \bar{s} = 0 \\ 0 & \bar{s} \neq 0 \end{cases}$$

Example: *Dummy Grid World*

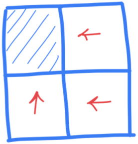


This concludes that at state $s = 0$, *Bellman equation reads*

$$v_\pi(0) = 0 + v_\pi(0)$$

which is an obvious equation; let's try $s = 1$

Example: *Dummy Grid World*



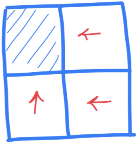
At state $s = 0$, we have

$$v_\pi(1) = \bar{\mathcal{R}}_\pi(1) + \sum_{\bar{s}=0}^{3} v_\pi(\bar{s})\, p_\pi(\bar{s}|1)$$

Again we can easily say based on the policy that

$$p_\pi(\bar{s}|1) = \begin{cases} 1 & \bar{s} = 0 \\ 0 & \bar{s} \neq 0 \end{cases}$$

Example: *Dummy Grid World*



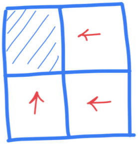
This conclues that at state $s = 1$, Bellman equation reads

$$v_\pi(1) = -1 + v_\pi(0)$$

which relates $v_\pi(1)$ to $v_\pi(0)$. If we keep repeating we get further

$$v_\pi(2) = -1 + v_\pi(0)$$
$$v_\pi(3) = -1 + v_\pi(2)$$

Example: *Dummy Grid World*



We now have the system of equations

$$v_\pi(1) = -1 + v_\pi(0)$$
$$v_\pi(2) = -1 + v_\pi(0)$$
$$v_\pi(3) = -1 + v_\pi(2)$$

We also know that $s = 0$ is a terminal state, and thus $v_\pi(0) = 0$: so, we get

$$v_\pi(1) = -1 \qquad v_\pi(2) = -1 \qquad v_\pi(3) = -2$$

Bellman Equation: *Action-Value*

We can find a Bellman equation for action-value function as well: *say we play with policy* π

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi\left\{G_t | s, a\right\} \\ &= \mathbb{E}_\pi\left\{R_{t+1} + \gamma G_{t+1} | s, a\right\} \\ &= \mathbb{E}\left\{R_{t+1} | s, a\right\} + \gamma \mathbb{E}_\pi\left\{G_{t+1} | s, a\right\} \\ &= \bar{\mathcal{R}}(s,a) + \gamma \underbrace{\mathbb{E}_\pi\left\{G_{t+1} | s, a\right\}}_{?} \end{aligned}$$

We need to compute

$$\mathbb{E}_\pi\left\{G_{t+1} | s, a\right\}$$

in terms of the rewarding-transition model and policy

Action-Value: *Recursive Property*

We apply the marginalization trick

$$\mathbb{E}_\pi \left\{ G_{t+1} | s, a \right\} = \sum_{n=1}^{N} \mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n, A_t = a \right\} p \left(s^n | s, a \right)$$

Attention

Recalling the trajectory of the MDP, we should note that

$$q_\pi \left(s^n, a \right) \neq \mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n, A_t = a \right\} = v_\pi \left(s^n \right)$$

In fact, once we know S_{t+1}, the previous action does not contain any extra information! We only gain information, if we observe A_{t+1}, i.e.,

$$\mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n, A_{t+1} = a \right\} = q_\pi \left(s^n, a \right)$$

Action-Value: *Recursive Property*

So, we can replace it into original equation to get

$$\mathbb{E}_\pi \left\{ G_{t+1} | s, a \right\} = \sum_{n=1}^{N} \mathbb{E}_\pi \left\{ G_{t+1} | S_t = s, S_{t+1} = s^n, A_t = a \right\} p\left(s^n | s, a\right)$$
$$= \sum_{n=1}^{N} v_\pi\left(s^n\right) p\left(s^n | s, a\right)$$

This implies that

$$q_\pi\left(s, a\right) = \bar{\mathcal{R}}\left(s, a\right) + \gamma \sum_{n=1}^{N} v_\pi\left(s^n\right) p\left(s^n | s, a\right)$$
$$= \bar{\mathcal{R}}\left(s, a\right) + \gamma \mathbb{E}\left\{ v_\pi\left(S_{t+1}\right) | s, a \right\}$$

Bellman Equation: *Action-Value*

Bellman Equation I for Action-Value Function

For any policy π the action-value function at each pair (s, a) satisfies

$$q_\pi(s, a) = \bar{\mathcal{R}}(s, a) + \gamma \sum_{n=1}^{N} v_\pi(s^n)\, p(s^n | s, a)$$

After doing Assignment 1, you will immediately conclude the following extension

Bellman Equation II for Action-Value Function

For any policy π the action-value function at each pair (s, a) satisfies

$$q_\pi(s, a) = \bar{\mathcal{R}}(s, a) + \gamma \sum_{n=1}^{N} \sum_{m=1}^{M} q_\pi(s^n, a^m)\, \pi(a^m | s^n)\, p(s^n | s, a)$$

Computing Action-Value via Bellman Equation

We can again use the recursive equation

$$q_\pi(s,a) = \bar{\mathcal{R}}(s,a) + \gamma \sum_{n=1}^{N} \sum_{m=1}^{M} q_\pi(s^n, a^m)\, \pi(a^m | s^n)\, p(s^n | s, a)$$

to find the action-value function: *we have in this case NM possible values*

- *Bellman equation relates each action-value to other action-values*
  - ↳ *For each s and a, Bellman equation has NM unknowns $q_\pi(s^n, a^m)$*
- *We can write the Bellman equation for all NM cases*
- *We solve this system of equations for unknowns $q_\pi(s^n, a^m)$*