Reinforcement Learning Notes

Varad Vaidya

January 18, 2024

Lecture notes from the various YouTube playlist related to Reinforcement Learning, combined with the course E1277 Reinforcement Learning at IISc Bangalore by Prof. Gugan Thoppe. The plan is to merge the notes from david silver's course and the course at IISc Bangalore, at appropriate places, to make a single set of notes.

Since the notes are being merged from two different sources, proper crediting of the two (and many more) is hard. In general, the notes will follow, from the standard books of Reinforcement Learning, and are my interpretation of the same.

Disclaimer: This document will inevitably contain some mistakes— both simple typos and legitimate errors. Keep in mind that these are the notes of a graduate student in the process of learning the material, so take what you read with a grain of salt. If you find mistakes and feel like telling me, I will be grateful and happy to hear from you, even for the most trivial of errors. You can reach me by email at vaidyavarad2001@gmail.com.

For more notes like this, visit varadVaidya.

Varad Vaidya,
Fall Term: 2023,
Last Update: January 18, 2024,

Contents

1 Lecture 2 — Markov Decision Processes

Markov decision process formally describe an enviroment for reinforcement learning. The nice case for this setting is that the environemtn is fully observable. Thus, the current state completely characterizes the process. This is called the Markov property.

Thus, all RL problems can bve formalised in terms of MDPs. Optimal Control problem can be formalised as continuous MDPs. Partially observable problems can be always converted to MDPs.

1.1 Markov Processes or Markov Chains

A Markov chain is simply a Markov Decision Process without decision. It is one the most simplest stocastic process, and has no "memory" of the past. So, just the present stat determines its future dynamics. In this context, we will be considering Discrete Time Markov Chains (DTMCs).

DTMC involves two concepts:

- Discrete Time Stochastic Process (DTSP)
- Row Stochastic Matrix.

The two are defined as follows:

Definition 1.1 (Discrete Time Stochastic Process (DTSP)). A DTSP is a sequence $(X_n)_{n\geq 0}$ of random variables defined on the same proboability space $(\Omega, \mathcal{F}, \mathbb{P})$, taking values in the same set $\mathcal{S}$, i.e.

$$X_n : \Omega \to \mathcal{S} \quad \forall n \geq 0$$

where $\mathcal{S}$ is called a state space, and in finite or countably infinite. We call the a variable as a "state" when the state is an element of the state space $\mathcal{S}$. The cardinality of the state space is denoted by $|\mathcal{S}|$.

Definition 1.2 (Row Stochastic Matrix). A matrix $\mathcal{P} \in \mathbb{R}^{|S|\times|S|}$ is called a row stochastic matrix if it satisfies the following conditions:

$$\mathcal{P}_{ij} \in [0,1] \quad \forall i,j \in \mathcal{S}$$

$$\sum_{j=1}^{|\mathcal{S}|} \mathcal{P}_{ij} = 1 \quad \forall i \in \mathcal{S}$$

Thus, with these two definitions we can define a Markov Chain (or DTMC) as follows:

Definition 1.3 (Markov Chain). Let $\mathcal{S}$ be a finite state space, and ν be a distribution over $\mathcal{S}$.

$$\nu = (\nu_1, \nu_2, \ldots, \nu_{|\mathcal{S}|}) \quad \text{s.t} \quad \nu_i \in [0,1]\ \forall i \in \mathcal{S}, \quad \sum_{i\in\mathcal{S}} \nu_i = 1$$

Further, let $\mathcal{P}$ be a row stocastic matrix over $\mathcal{S}$.

Then a DTSP $(X_n)_{n\geq 0}$ is called a Markov Chain with initial distribution ν and transition matrix $\mathcal{P}$ if it satisfies the following conditions:

- Initial state is distributed according to ν, i.e.

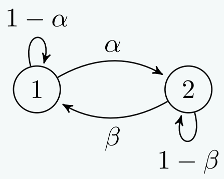
$$\mathbb{P}(X_0 = i) = \nu_i \quad \forall i \in \mathcal{S}$$

- Present is independent of the past given the present.

$$\begin{aligned}\mathbb{P}\left(X_{n+1}=i_{n+1}|X_n=i_n,X_{n-1}=i_{n-1},\ldots,X_0=i_0\right)&=\mathbb{P}\left(X_{n+1}=i_{n+1}|X_n=i_n\right)\\&=\mathcal{P}_{i_ni_{n+1}}\quad\forall n\geq 0\end{aligned}$$

Example (Markov Chains). Let a Markov chain be defined as follows:

$$\mathcal{S}=\{1,2\}\quad \nu=(p,1-p)\quad \mathcal{P}=\begin{bmatrix}1-\alpha & \alpha\\ \beta & 1-\beta\end{bmatrix}$$

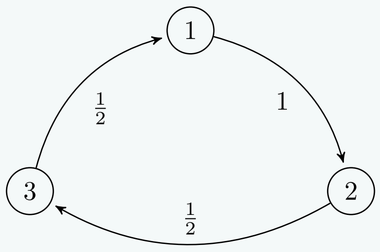


Thus, we can have say:

$$\mathbb{P}(X_3=1|X_0=1,X_1=1,X_2-2)=\beta$$

Another example that we can have is:

$$\mathcal{S}=\{1,2,3\}\quad \nu=(\nu_1,\nu_2,\nu_3)\quad \mathcal{P}=\begin{bmatrix}0 & 1 & 0\\ 0 & \frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & 0 & \frac{1}{2}\end{bmatrix}$$



given, $\nu=(1,0,0)$, the outcomes of the Markov chain if we sample it are:

$$\begin{aligned}X_n&=1,2,3,3,1,2,\ldots\\&=1,2,3,1,2,\ldots\end{aligned}$$

Note that, to define the DTMC, we need the initial distribution, but the results wont change on the choice of the distribution ν, assuming that the Markov chain is ergodic. Hence, while defining the markov chain, the initial distribution is not mentioned, and can be assumed arbitrarily.

Theorem 1.1 (Necessary and Sufficient Conditions for DTSP to be Markov Chains). A DTSP $(X_n)_{n\geq 0}$ on $\mathcal{S}$ is a Markov chain $\langle\mathcal{S},\mathcal{P},\nu\rangle$ if and only if:

$$\mathbb{P}\left(X_n=i_n,X_n=i_n,\ldots,X_0=i_0\right)=\nu_{i_0}\mathcal{P}_{i_0i_1}\mathcal{P}_{i_1i_2}\ldots\mathcal{P}_{i_{n-1}i_n}\quad\forall n\geq 0$$

Proof. To simplify the proof, assume that $\mathcal{P}_{ij} > 0 \quad \forall i, j \in \mathcal{S}$. The theorem remains true in the general case, but requires more book-keeping.

Suppose, $(X_n)_{n\geq 0}$ is a Markov chain $\langle \mathcal{S}, \mathcal{P}, \nu \rangle$.

Using the fact,

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B|A)$$
$$\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A)\mathbb{P}(B|A)\mathbb{P}(C|A \cap B)$$

We get,

$$\begin{aligned}\mathbb{P}(X_0 = i_0, \dots, X_n = i_n) &= \mathbb{P}\left(\{X_0 = i_0\} \cap \{X_1 = i_1\} \cap \dots \cap \{X_n = i_n\}\right)\\ &= \mathbb{P}(X_0 = i_0)\mathbb{P}(X_1 = i_1|X_0 = i_0)\dots\mathbb{P}(X_n = i_n|X_{n-1} = i_{n-1})\\ &= \mathbb{P}(X_0 = i_0)\mathbb{P}(X_1 = i_1|X_0 = i_0)\dots\mathbb{P}(X_n = i_n|X_{n-1} = i_{n-1})\\ &\dots \text{Using the memoeryless property of Markov chains}\\ &= \nu_{i_0}\mathcal{P}_{i_0 i_1}\dots\mathcal{P}_{i_{n-1} i_n}\end{aligned}$$

This proves the forward claim. To show the reverse claim:

Put $n = 0$, in the claim, to trivially get the initial distribution back, showing one part of the definition. For the other part:

$$\begin{aligned}\mathbb{P}(X_n = i_n|X_{n-1} = i_{n-1}, \dots, X_0 = i_0) &= \frac{\mathbb{P}(X_n = i_n, \dots, X_0 = i_0)}{\mathbb{P}(X_{n-1} = i_{n-1}, \dots, X_0 = i_0)}\\ &= \frac{\nu_{i_0}\mathcal{P}_{i_0 i_1}\dots\mathcal{P}_{i_{n-1} i_n}}{\nu_{i_0}\mathcal{P}_{i_0 i_1}\dots\mathcal{P}_{i_{n-1} i_n}}\\ &= \mathcal{P}_{i_{n-1} i_n}\end{aligned}$$

Now we need to show:

$$\begin{aligned}&\mathbb{P}(X_n = i_n|X_{n-1} = i_{n-1})\\ &= \frac{\mathbb{P}(X_{n-1} = i_{n-1}, X_n = i_n)}{\mathbb{P}(X_{n-1} = i_{n-1})}\\ &= \frac{\sum_{i_0 \in \mathcal{S}} \mathbb{P}(X_0 = i_0 \dots X_{n-1} = i_{n-1}, X_n = i_n)}{\sum_{i_0 \in \mathcal{S}} \mathbb{P}(X_{n-1} = i_{n-1}, X_0 = i_0)}\\ &= \mathcal{P}_{i_{n-1} i_n}\end{aligned}$$

☺

1.2 Markov Reward Processes

A Markov Reward Process is a Markov chain with values.

Definition 1.4 (Markov Reward Process). A Markov Reward Process is a tuple $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$ consisting of:

- a finite set of states $\mathcal{S}$
- a state transition probability matrix $\mathcal{P}$

$$\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s'|S_t = s]$$

- a reward function $\mathcal{R}$

$$\mathcal{R}_s = \mathbb{E}[R_{t+1}|S_t = s]$$

- a discount factor $\gamma \in [0, 1]$

Definition 1.5 (Return). The return G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

NBOTE: the goial of RL is to maximise the expected return from the start state.

The discount factor γ determines the present value of future rewards. THe discount factor of 0 makes the agent myopic, it only cares about immediate rewards. The discount factor of 1 makes the agent strive for a long-term reward.

Why do we use discounting?

Most Markov reward processes and decision process are discounted. This is done so account for the uncertainity in the dynamics of the environement. It also allows us to converge to a solution in the infinite/cyclic Markov processes. Somtimes it is possible to use undiscounted Markov processes, if all sequences terminate in a finite number of steps.

1.2.1 Value Function

The value function $v(s)$ gives the long-term value of state s. It is the expected return starting from state s.

Definition 1.6 (Value Function). The value function $v(s)$ of an MRP is the expected return starting from state s.

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

1.2.2 Bellman Equation for MRPs

The value function can be decomposed into two parts:

- immediate reward R_{t+1}
- discounted value of successor state $\gamma v(S_{t+1})$

Thus we have:

$$\begin{aligned} v(s) &= \mathbb{E}[G_t \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots) \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s] && \text{\dots using the law of iterated expectations} \end{aligned}$$

This can be explained with what is called the tree backup diagram.

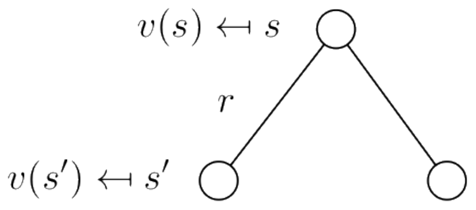


Figure 1: Recursive relationship between $v(s)$ and $v(s')$

From the Figure 1, we can see that the value of the state s is the expected reward plus average of all the values of all the possible successor states.

$$\Rightarrow v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

Thus, the Bellman equation can be expreseed as a linear system of equations.

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

$$\begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(3) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \mathcal{R}_2 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} & \dots & \mathcal{P}_{1n} \\ \mathcal{P}_{21} & \mathcal{P}_{22} & \dots & \mathcal{P}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \mathcal{P}_{n2} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(3) \end{bmatrix}$$

SInce this is a linear system of equations, we can solve it using linear algebra.

$$\begin{aligned} v &= \mathcal{R} + \gamma \mathcal{P} v \\ (I - \gamma \mathcal{P}) v &= \mathcal{R} \\ v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R} \end{aligned}$$

Too large to compute in practice. Thus, we use iterative methods to solve this equation.

1.3 Markov Decision Processes

A Markov Decision Process is a Markov Reward Process with decisions. It is an environment in which all states are Markov. Thus the next state that the MDp transitions to depends on the current state and the action that the agent takes in the current state.

Definition 1.7 (Markov Decision Process)**.** A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ consisting of:

- a finite set of states $\mathcal{S}$
- a finite set of actions $\mathcal{A}$
- a state transition probability matrix $\mathcal{P}$

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

- a reward function $\mathcal{R}$

$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

- a discount factor $\gamma \in [0, 1]$

1.3.1 Policy

Formalises what it means to take decisions.

Definition 1.8 (Policy)**.** A policy π is a distribution over actions given states.

$$\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$$

The policies fully define the behaviour of the agent. Usually, the policies are stationary, i.e. they do not change over time. The policies are dependent only on the current state and not on the history of the agent.

NOTE:

- We can always recover the MRP or a Markov Process from an MDP, given and MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- If we have an policy and we sample the states using the policy, the state sequence is a Markov Process $\langle \mathcal{S}, \mathcal{P}^{\pi} \rangle$
- Similarly, the state and reward sequence is a Markov Reward Process $\langle \mathcal{S}, \mathcal{P}^{\pi}, \mathcal{R}^{\pi}, \gamma \rangle$

where the transition dynamics and reward function are the average over the policy.

$$\mathcal{P}^{\pi}_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^{a}_{ss'}$$

$$\mathcal{R}^{\pi}_{s} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^{a}_{s}$$

1.3.2 Value Function and Action Value Function

Definition 1.9 (Value Function)**.** The value function $v_{\pi}(s)$ of an MDP is the expected return starting from state s, and then following policy π.

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

Definition 1.10 (Action Value Function)**.** The action-value function $q_{\pi}(s, a)$ of an MDP is the expected return starting from state s, taking action a, and then following policy π.

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

1.3.3 Bellman Expectation Equation for MDPs

The action value function can be decomposed into two parts similar to the value function.

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

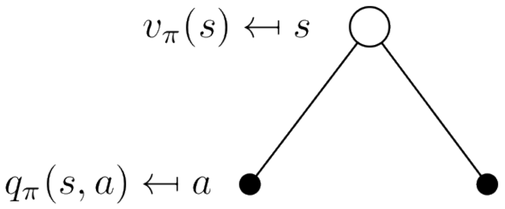


Figure 2: The relationship between q_π and v_π

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

The Figure 2 shows how q_π and v_π are related. The black dots represent the possibel actions, while the circles represent the states. The probabilities of chossing the action depends on the policy π. For each of the action we might take from state s, we have a q-value $q_\pi(s,a)$, that describes the expected return from taking action a in state s. Thus, the value of the state s, is calculated by taking the average of all the q values after doing a one step look-ahead. Simlarly the q value of the state action pair is calculated by averaging the value of the state that we might transition into according to the MDP dynamics, after taking action a from state s. The tree backup diagram for the q value is shown in Figure 3.

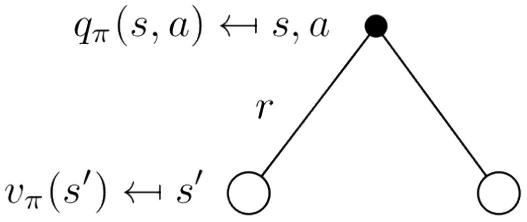

$q_\pi(s,a) \hookleftarrow s,a$
r
$v_\pi(s') \hookleftarrow s'$

Figure 3: The relationship between q_π and v_π

$$\Rightarrow q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

Combining Figure 2 and Figure 3, we get Figure 4 a recursive relationship between $v(s)$ and $v(s')$. Thus, we are averaging over the policy, and the transition dynamics of the MDP.

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

The Bellman Expectation Equation can be written concisely in matrix form.

$$v_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_\pi$$
$$v_\pi = (I - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}^\pi$$

1.3.4 Optimal Value Function

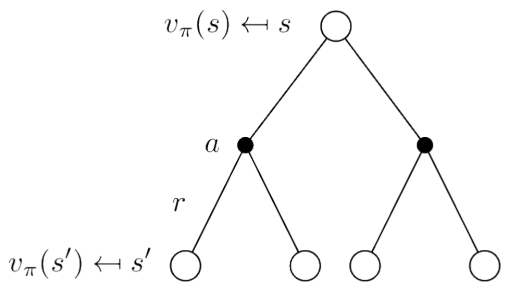


Figure 4: Recursive relationship between $v(s)$ and $v(s')$

Definition 1.11 (Optimal Value Function)**.** The optimal value function $v_*(s)$ is the maximum value function over all policies.

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

The optimal action value function $q_*(s,a)$ is the maximum action value function over all policies.

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

If we know the optimal action-value function, we can easily construct the optimal policy. So, we can say that the RL problem is solved if we know $q_*(s,a)$. The way to compare policies is to compare the value functions of the policies.

$$\pi \geq \pi' \Leftrightarrow v_\pi(s) \geq v_{\pi'}(s) \quad \forall s \in \mathcal{S}$$

Theorem 1.2 (Optimal Policy)**.** For any MDP:

- There exists an optimal policy π_* that is better than or equal to all other policies,

$$\pi_* \geq \pi \quad \forall \pi$$

- All optimal policies achieve the optimal value function,

$$v_{\pi_*}(s) = v_*(s) \quad \forall s \in \mathcal{S}$$

- All optimal policies achieve the optimal action-value function,

$$q_{\pi_*}(s,a) = q_*(s,a) \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$

The optimal policy can be found by maximising over $q_*(s,a)$.

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in \mathcal{A}} q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

1.3.5 Bellman Optimality Equation

The optimal vale functions are recursively related by the Bellman optimality equation. Before we looked at the Expectation equation, looking at the average value of the state. Now, we look at the maximum value of the state. Thus, taking an action a from the state s, we choose the action that has the maximum state-action value. The tree backup diagram for the optimal value function is shown in Figure 5

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

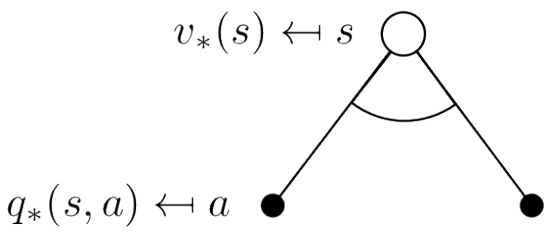

$v_*(s) \leftarrow\!\!\dashv s$
$q_*(s,a) \leftarrow\!\!\dashv a$

Figure 5: Bellman Optimality Equation for $v_*(s)$

Similarly, the same argument can be made for the optimal action value function, with its tree backup diagram shown in Figure 6.

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

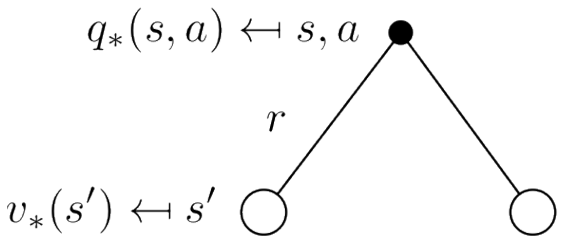

$q_*(s,a) \leftarrow\!\!\dashv s,a$
r
$v_*(s') \leftarrow\!\!\dashv s'$

Figure 6: Bellman Optimality Equation for $q_*(s,a)$

Combining Figure 5 and Figure 6 , we get Figure 7, with the Bellman Optimality Equation for $v_*(s)$, being written in a recursive form as

$$v_*(s) = \max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s') \right)$$

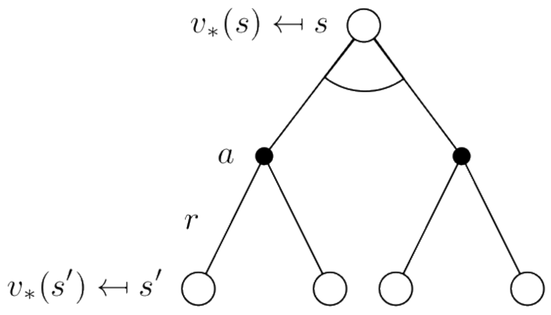

$v_*(s) \leftarrow\!\!\dashv s$
a
r
$v_*(s') \leftarrow\!\!\dashv s'$

Figure 7: Recursive relationship between $v_*(s)$ and $v_*(s')$

Similarly, the Bellman Optimality Equation for $q_*(s,a)$ can be written in a recursive form as

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a' \in \mathcal{A}} q_*(s',a')$$

Solving the Bellman Optimality Equation

- Bellman optimality equation is a non-linear equation
- It can be solved using iterative methods
  - Value Iteration
  - Policy Iteration
  - Q-learning
  - Sarsa