# Planning in Markov Decision Problems

Shivaram Kalyanakrishnan

February 10, 2018

**Abstract**

In this note, we introduce the Markov Decision Problem (MDP), which is a classical abstraction of sequential decision making. Given an MDP, the planning problem is to find a way to take actions so as to maximise expected long-term reward: in other words, to compute an optimal *policy*. We present the Value Iteration algorithm, which is a commonly-used method for MDP planning.

## 1 Introduction

The Markov Decision Problem (MDP) has been in use for several decades as a formal framework for sequential decision making under uncertainty. An MDP models an agent whose actions result in stochastic state transitions, while yielding associated rewards. The agent's natural aim is to consistently take actions that lead to high long-term reward. Thus, given an MDP, the central computational question is that of determining an optimal way for the agent to act. This is the problem of MDP *planning*.

Formally, an MDP $M = (S, A, T, R, \gamma)$ has a set of states $S$ in which an agent can be, and a set of actions $A$ that the agent can execute. Upon taking action $a$ from state $s$, the agent is transported to a state $s'$, selected from $S$ at random with probability $T(s, a, s')$. For this transition, the agent receives a reward $R(s, a, s')$.

To fully specify $M$, we need to define an objective for the agent. A *policy* (assumed stationary, deterministic, and Markovian) is a mapping from $S$ to $A$: when *following* a policy $\pi$, the agent takes action $\pi(s)$ when in state $s$. A natural objective is to find a policy that maximises the agent's expected long-term reward. Consider an agent that starts in some state $s^0$ at time 0 and continually follows $\pi$. The agent thereby encounters a trajectory over time:

$$s^0, \pi(s^0), r^0, s^1, \pi(s^1), r^1, s^2, \ldots.$$

The *value* of state $s$ under policy $\pi$ is given by

$$V^\pi(s) \stackrel{\text{def}}{=} \left[ \sum_{t=0}^{\infty} \gamma^t r^t | s^0 = s, \text{ for } t \geq 0 : a^t = \pi(s^t) \right], \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor. The discount factor specifies the relative importance of long-term rewards. We need to have $\gamma < 1$ in order that values remain well-defined. In special MDPs that have terminal states, which are guaranteed to be reached under every policy, it is okay to take $\gamma = 1$. $V^\pi : S \to \mathbb{R}$ is called the value function of $\pi$.

MDPs can be specified using state transition diagrams. Figure 1 provides an example of a 2-state, 2-action MDP.

| Transition probabilities | Rewards |
|---|---|
| $T(s_1, a_1, s_1) = 1$ | $R(s_1, a_1, s_1) = 0$ |
| $T(s_1, a_1, s_2) = 0$ | $R(s_1, a_1, s_2) = 0$ |
| $T(s_1, a_2, s_1) = 1/2$ | $R(s_1, a_2, s_1) = 0$ |
| $T(s_1, a_2, s_2) = 1/2$ | $R(s_1, a_2, s_2) = 2$ |
| $T(s_2, a_1, s_1) = 1$ | $R(s_2, a_1, s_1) = 1$ |
| $T(s_2, a_1, s_2) = 0$ | $R(s_2, a_1, s_2) = 0$ |
| $T(s_2, a_2, s_1) = 1/4$ | $R(s_2, a_2, s_1) = -1$ |
| $T(s_2, a_2, s_2) = 3/4$ | $R(s_2, a_2, s_2) = 1$ |

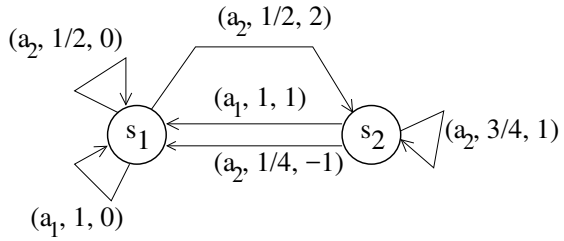


Figure 1: State transition diagram corresponding to an MDP with $S = \{s_1, s_2\}$ and $A = \{a_1, a_2\}$. Each transition is annotated with (action, transition probability, reward). Transitions with zero probabilities are not shown. The discount factor is $\gamma = \frac{2}{3}$.

Let $\Pi$ be the set of distinct policies corresponding to the MDP $(S, A, T, R, \gamma)$ (thus $|\Pi| = |A|^{|S|}$). It is a key property of MDPs that this set contains a policy $\pi^*$ such that $\forall s \in S, \forall \pi \in \Pi$,

$$V^{\pi^\star}(s) \geq V^{\pi}(s). \tag{2}$$

Such a policy $\pi^\star$ is called an *optimal* policy. The MDP planning problem is precisely that of finding an optimal policy for a given MDP $M = (S, A, T, R, \gamma)$.

## 2 Policy Evaluation

Before we attempt a solution to the MDP planning problem, we take up the simpler problem of computing the value function $V^\pi$ of a *given* policy $\pi$. From the definition of values in (1), it can be shown that $\forall s \in S$,

$$V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s') \left( R(s, \pi(s), s') + \gamma V^{\pi}(s') \right).$$

This definition implies that the value function of a policy can be computed efficiently by solving a system of $n$ linear equations, which are called Bellman's Equations.

The four policies that apply to the MDP from Figure 1 are $\pi^{11}$, $\pi^{12}$, $\pi^{21}$, and $\pi^{22}$, where for $i, j \in \{1, 2\}$, we denote by $\pi^{ij}$ that policy that takes action $a_i$ from state $s_1$ and action $a_j$ from state $s_2$. Write down the two Bellman's Equations that arise for each policy, and solve them. You should obtain the following results.

$$V^{\pi^{11}}(s_1) = 0, V^{\pi^{11}}(s_2) = 1.$$

$$V^{\pi^{12}}(s_1) = 0, V^{\pi^{12}}(s_2) = \frac{3}{2}.$$

$$V^{\pi^{21}}(s_1) = \frac{15}{8}, V^{\pi^{21}}(s_2) = \frac{9}{4}.$$

$$V^{\pi^{22}}(s_1) = \frac{9}{5}, V^{\pi^{22}}(s_2) = \frac{21}{10}.$$

For this MDP, $\pi^{21}$ is the sole optimal policy. In general an MDP can have multiple optimal policies. However, they will all have the same value function, which is denoted $V^\star$.

## 3 Finding an Optimal Policy

Policy evaluation provides us with an obvious method to find an optimal policy: evaluate every policy, and compare value functions to identify an optimal policy. While simple and correct, this approach can be inefficient in practice, since it requires the enumeration of an exponential number of policies.

A more practical approach is to directly solve for the optimal value function, which is the solution to another set of equations called Bellman's Optimality Equations: for $s \in S$,

$$V^\star(s) = \max_{a \in A} \left( \sum_{s' \in S} T(s,a,s') \left( R(s,a,s') + \gamma V^\star(s') \right) \right).$$

Unlike Bellman's Equations, notice that Bellman's Optimality Equations are not linear. A common way to solve them (approximately) is through iteration: we begin with an arbitrary guess $V_0$, which is updated by applying the non-linear Bellman Optimality Operator. In the limit, the iteration converges to $V^\star$. In practice, the iteration is stopped when successive iterates get within some small threshold of each other, which ensures that $V^\star$ is also within a small threshold of them. The procedure below is called Value Iteration.

**Value Iteration**

$V_0 \leftarrow$ Arbitrary initial guess of $V^\star$.
$t \leftarrow 0$.
Repeat
    For $s \in S$
        $V_{t+1}(s) \leftarrow \max_{a \in A} \left( \sum_{s' \in S} T(s,a,s') \left( R(s,a,s') + \gamma V_t(s') \right) \right)$.
    $t \leftarrow t + 1$.
Until $V_t \approx V_{t-1}$.
Return $V_t$.

If $V^\star$ is known, an optimal policy $\pi^\star$ can be obtained by taking, for $s \in S$:

$$\pi^\star(s) \leftarrow \operatorname*{argmax}_{a \in A} \left( \sum_{s' \in S} T(s,a,s') \left( R(s,a,s') + \gamma V^\star(s') \right) \right).$$