Notes for chapter 4: Markov Decision Processes

The Difficulty of Sequential Decisioning Under Uncertainty

Markov Decision Processes have 2 distinct high-level features:

- At each timestep t an action A_t is picked after observing the state S_t
- Given an observed state S_t the probabilities of the next reward R_{t+1} and the next state S_{t+1} are a function of S_t as well as A_t

We are tasked with maximizing the expected future return from each state. This is a hard problem in general because there is cyclic interplay between:

- actions depending on state, on one hand, and
- next state/reward depending on current state and action, on the other hand.

Furthermore, there is also the challenge that actions might have consequences for future rewards.

Formal Definitions of a Markov Decision Process

Definition (Markov Decision Process)**.** A Markov Decision Process consists of:

- A countables set of states $\mathcal{S}$, a set of terminal states $\mathcal{T} \subset \mathcal{S}$ and a countable set of actions $\mathcal{A}$
- A time-indexed sequence of states $S_t \in \mathcal{S}, t = 0, 1, 2, ...$, a time-indexed sequence of environment generated rewards $R_t \in \mathcal{D}, t = 0, 1, 2, ...$ and a time-indexed sequence of agent-controllable actions $A_t \in \mathcal{A}, t = 0, 1, 2, ...$
- Transistion probabilities that statisfy the Markov Property, i.e. for all $t = 0, 1, 2, ...$ it must hold that:
$$\mathbb{P}[(S_{t+1}, R_{t+1})|(S_t, A_t, S_{t-1}, A_{t-1}, ..., S_0, A_0)] = \mathbb{P}[(S_{t+1}, R_{t+1})|(S_t, A_t))]$$

As with the Markov Reward Process we define the set of nonterminal states as $\mathcal{N} = \mathcal{S} \backslash \mathcal{T}$.
We can express the transistion probabilities as a state-reward transistion probability function:

$$\mathcal{P}_\mathcal{R} : \mathcal{N} \times \mathcal{A} \times \mathcal{D} \times \mathcal{S} \to [0, 1],\, \mathcal{P}_\mathcal{R}(s, a, r, s') = \mathbb{P}[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a]$$
$$\text{for } t = 0, 1, 2, ..., s \in \mathcal{N}, a \in \mathcal{A}, r \in \mathcal{D} \text{ and } s' \in \mathcal{S}.$$
$$\text{Such that } \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{D}} \mathcal{P}_\mathcal{R}(s, a, r, s') = 1,\ \forall s \in \mathcal{N} \ \&\ \forall a \in \mathcal{A}.$$

Any Markov Decision Process can be characterized by it's state-reward transistion probability function $\mathcal{P}_\mathcal{R}$ Given $\mathcal{P}_\mathcal{R}$ we can construct:

- The state transistion probability function
$$\mathcal{P} : \mathcal{N} \times \mathcal{A} \times \mathcal{S} \to [0, 1],\ \mathcal{P}(s, a, s') = \sum_{r \in \mathcal{D}} \mathcal{P}_\mathcal{R}(s, a, r, s')$$
- And the reward transistion function:
$$\mathcal{R}_T : \mathcal{N} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R},\ \mathcal{R}_T(s, a, s') = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'] = \sum_{r \in \mathcal{D}} \frac{\mathcal{P}_\mathcal{R}(s, a, r, s')}{\mathcal{P}(s, a, s')} \cdot r$$

The reward specifications of must MDP's can be expressed as $\mathcal{R}_T$, also note that we can express $\mathcal{R}_T$ into a more compact form sufficient for performing key calculations involving MDP's.

$$\mathcal{R} : \mathcal{N} \to \mathbb{R},\, \mathcal{R}(s) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s')\mathcal{R}_T(s, a, s') = \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{D}} \mathcal{P}_\mathcal{R}(s, a, r, s') \cdot r$$

The Policy Function

The Agent will perform an action according to a probability distribution that is a function of our current state S_t. This function is our Policy,

Definition (Policy Function)**.** A Policy Function is a function $\pi : \mathcal{N} \times \mathcal{A} \to [0, 1]$,

$$\pi(s, a) = \mathbb{P}[A_t = a | S_t = s],\ t = 0, 1, 2, ...,\ \forall s \in \mathcal{N}.$$

$$\text{Such that } \sum_{a \in \mathcal{A}} \pi(s, a) = 1,\ \forall s \in \mathcal{N}.$$

Note that the above definition assumes that π statisfies the Markov Property, and that π is invariant in time t. When we have a policy such that the action probability distribution is concentrated on a single action, we refer to it as a deterministic policy.

Definition (Deterministic Policy Property)**.** A deterministic policy $\pi_D : \mathcal{N} \to \mathcal{A}$ has the property that $\forall s \in \mathcal{N}$:

$$\pi(s, \pi_D) = 1 \text{ and } \pi(s, a) = 0,\ \forall a \in \mathcal{A} \text{ with } a \neq \pi_D(s).$$

A policy that is not deterministic is called a stochastic policy.

[MDP, Policy] := MRP

If we evaluate a MDP with a fixed Policy we get a MRP. That is implied by the combination of the MRP and the fixed Policy. Let's say we have a fixed Policy and a MDP specified by it's state-reward transistion function $\mathcal{P}_\mathcal{R}$. Then the state-reward transistion function of the MRP implied by the evaluation of the MDP with the fixed policy is defined as:

$$\mathcal{P}^\pi_\mathcal{R}(s, r, s') = \sum_{a \in \mathcal{A}} \pi(s, a)\mathcal{P}_\mathcal{R}(s, a, r, s')$$

Likewise:

- $\mathcal{P}^\pi(s, s') = \sum_{a \in \mathcal{A}} \sum_{r \in \mathcal{D}} \pi(s, a)\mathcal{P}_\mathcal{R}(s, a, r, s') = \sum_{a \in \mathcal{A}} \pi(s, a)\mathcal{P}(s, a, s')$.
- $\mathcal{R}^\pi_T(s, s') = \sum_{a \in \mathcal{A}} \pi(s, a)\mathcal{R}_T(s, a, s')$.
- $\mathcal{R}^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s, a)\mathcal{R}(s, a)$

So anytime we talk about a MDP with a fixed policy we're effectively talking about the implied MRP.

MDP Value Function for a fixed policy

Definition. The Value Function for a MDP with fixed policy π is:

$$V^\pi : \mathcal{N} \to \mathbb{R},\ V^\pi(s) = \mathop{\mathbb{E}}_{\pi, \mathcal{P}_\mathcal{R}}[G_t | S_t = s],\ \forall s \in \mathcal{N},\ t = 0, 1, 2, ... \,.$$

$$\text{Where } G_t \text{ is our future return function: } G_t = \sum_{i=t+1}^{\infty} \gamma^{i-t+1} R_i$$

.

We can also apply the MRP Bellman equation on V^π:

$$\begin{aligned} V^\pi(s) &= \mathop{\mathbb{E}}_{\pi,\mathcal{P}_\mathcal{R}}[G_t|S_t = s] \\ &= \mathcal{R}^\pi(s) + \gamma \cdot \sum_{s' \in \mathcal{N}} \mathcal{P}^\pi(s,s')V^\pi(s') \\ &= \sum_{a \in \mathcal{A}} \pi(s,a)\mathcal{R}(s,a) \ + \ \gamma \sum_{s' \in \mathcal{N}} \sum_{a \in \mathcal{A}} \pi(s,a)\mathcal{P}(s,a,s')V^\pi(s') \\ &= \sum_{a \in \mathcal{A}} \pi(s,a) \cdot (\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s')V^\pi(s')), \ \forall s \in \mathcal{N}. \end{aligned} \tag{1}$$

Another crucial function for MDP's, Dynamic Programming and RL algorithms is the Action-Value Function for a MDP with fixed policy π:

Definition (Action Value Function for a MDP with fixed policy)**.**

$$Q^\pi : \mathcal{N} \times \mathcal{A} \to \mathbb{R}, \ Q^\pi(s,a) = \mathop{\mathbb{E}}_{\pi,\mathcal{P}_\mathcal{R}}[G_t|S_t = s, A_t = a], \ \forall s \in \mathcal{N}, \ \forall t = 0,1,2,....$$

The interpertation of this function is that it's the expected return of future rewards by taking an action a and subsequantially following policy π. In contrast to the Value Function which is the expected return of future rewards by following policy π. Hence we can percieve V^π as the weighted average of Q^π over all possible actions. Precisely:

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(s,a)Q^\pi(s,a), \ \forall s \in \mathcal{N}. \tag{2}$$

Using the final result of equation (1) we can derive that:

$$Q^\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s')V^\pi(s'), \ \forall a \in \mathcal{A}, \ \forall s \in \mathcal{N}. \tag{3}$$

Combining this with (2) gives;

$$Q^\pi(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s') \sum_{a \in \mathcal{A}} \pi(s',a)Q^\pi(s',a), \ \forall a \in \mathcal{A}, \ \forall s \in \mathcal{N}. \tag{4}$$

Optimal Value Function & Optimal Policies

Definition (Optimal Value Function)**.**

$$V^* : \mathcal{N} \to \mathbb{R}, \ V^*(s) = \max_{\pi \in \Pi} V^\pi(s), \ \forall s \in \mathcal{N}$$

. For all non-terminal states we consider all possible stationairy policies $\pi \in \Pi$ and maximize $V^\pi(s)$ across these choices of π. Note that maximization of $V^\pi(s)$ is done for all $s \in \mathcal{N}$, hence there might exist different optimal policies for different s.

Definition (Optimal Action-Value Function)**.**

$$Q^* : \mathcal{N} \times \mathcal{A} \to \mathbb{R}, \ Q^*(s,a) = \max_{\pi \in \Pi} Q^\pi(s,a), \ \forall s \in \mathcal{N}, \ \forall a \in \mathcal{A}$$

.

Let's look at the optimal value function $V^*(s)$ for a given $s \in \mathcal{N}$ we consider all possible actions $a \in \mathcal{A}$ we can choose at this state, and pick the one that gives the highest action value. This results in the following equation:

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a), \ \forall s \in \mathcal{N}. \tag{5}$$

Now once again consider the Optimal Action-Value Function $Q^*(s)$. First we get the immediate expected reward $\mathcal{R}(s,a)$. Then we consider all possible nect states $s' \in \mathcal{N}$ and their state-transistion probabilities and we recursively act optimally. This gives us the following equation:

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s')V^*(s'),\ \forall a \in \mathcal{A},\ \forall s \in \mathcal{N}. \tag{6}$$

Substituting in equation (5) gives us the MDP State-Value Bellman Optimality Equation:

$$V^*(s) = \max_{a \in \mathcal{A}} [\mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s')V^*(s')],\ \forall s \in \mathcal{N}. \tag{7}$$

Substituting (5) in (6) gives us the MDP Action-Value Bellman Optimality Equation:

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s') \max_{a \in \mathcal{A}} Q^*(s',a),\ \forall s \in \mathcal{N}. \tag{8}$$

Definition (Optimal Policy)**.** $\pi^* \in \Pi$ is an optimal policy if

$$V^{\pi^*}(s) \geq V^{\pi}(s),\ \forall s \in \mathcal{N},\ \forall \pi \in \Pi.$$

Theorem. For any Discrete-Time Time-Homogeneous Countable-Spaces MDP

- There exists an optimal policy $\pi^* \in \Pi$ i.e. $\exists \pi^* \in \Pi : V^{\pi^*}(s) \geq V^{\pi}(s),\ \forall s \in \mathcal{N},\ \forall \pi \in \Pi$.
- All optimal policies achieve the optimal value function. That is, $V^{\pi^*}(s) = V^*(s),\ \forall s \in \mathcal{N}$.
- All optimal policies achieve the optimal Action-Value function. That is, $Q^{\pi^*}(s,a) = Q^*(s,a),\ \forall s \in \mathcal{N},\ \forall a \in \mathcal{A}$.

Lemma. For any two optimal policies $\pi_1^*,\ \pi_2^* \in \Pi$ it holds that $V^{\pi_1^*}(s) = V^{\pi_2^*}(s),\ \forall s \in \mathcal{N}$.

Proof (L). Since π_1^* is an optimal policy $V^{\pi_1^*}(s) \geq V^{\pi_2^*}(s),\ \forall s \in \mathcal{N}$, likewise since π_2^* is an optimal policy $V^{\pi_1^*}(s) \leq V^{\pi_2^*}(s),\ \forall s \in \mathcal{N}$. Hence, $V^{\pi_1^*}(s) = V^{\pi_2^*}(s),\ \forall s \in \mathcal{N}$. □

Proof (T). Now all we have to do is proof that there exists an optimal policy that achieves the optimal value function and the optimal action value function. Define the following Deterministic Policy as a candidate Optimal Policy:

$$\pi_D^* : \mathcal{N} \to \mathcal{A},\ \pi_D^* = \arg\max_{a \in \mathcal{A}} Q^*(s,a),\ \forall s \in \mathcal{N} \tag{9}$$

First we show that π_D^* achieves the Optimal Value Functions V^* and Q^*. Since $\pi_D^* = \arg\max_{a \in \mathcal{A}} Q^*(s,a)$ and $V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a)$ for all $s \in \mathcal{N}$, we can infer that:

$$V^*(s) = Q * (s, \pi_D^*(s))$$

This tells us that the optimal value funtion $V^*(s)$ is achieved if if from each non-terminal state, we take the action prescribed by π_D^*. Likewise $Q^*(s,a)$ is achieved if from each non-terminal state, we take the action a followed by future actions prescribed by π_D^*. Formally this is says:

$$V^{\pi_D^*}(s) = V^*(s),\ \forall s \in \mathcal{N}$$

$$Q^{\pi_D^*}(s,a) = Q^*(s,a),\ \forall s \in \mathcal{N},\ \forall a\ in \mathcal{A}$$

Finally we must show that π_D^* is indeed an optimal policy. Assume for a contradiction that this is not the case, then by the definition of optimal policies $\exists \pi \in \Pi$ and a state $s \in \mathcal{N}$ such that $V^{\pi}(s) > V^{\pi_D^*}(s)$. Since $V^{\pi_D^*}(s) = V^*(s)$ we have that $V^{\pi}(s) > V * (s)$ which contradicts Optimal Value Function definition: $V^*(s) = \max_{\pi \in \Pi} V^{\pi}(s),\ \forall s \in \mathcal{N}$. Therefore, π_D^* must be an optimal policy. □

Equation (9) is a key construction that goes hand-in-hand with the Bellman Optimality Equations in designing the various Dynamic Programming and RL algorithms to solve the MDP control problem.