Fundamental Results in MDP Theory and RL

Dipendra Misra (dkm@cs.cornell.edu), Cornell University, New York.

1. Markov Decision Process (MDP) is a tuple $\langle S, A, P, R, \gamma \rangle$, where S is a set of states, A is a set of actions, $P : S \times A \times S \rightarrow [0, 1]$ is the transition function where $P^a_{s,s'}$ for a given $s, s' \in S, a \in A$ denotes the probability of transitioning to state s' from state s on taking action a. $R : S \times A \times S \rightarrow \mathbb{R}$ is the reward function where $R^a_{s,s'}$ for a given $s, s' \in S, a \in A$ denotes the expected reward achieved on transitioning to state s' from state s on taking action a. $\gamma \in [0, 1)$ denotes the discounting factor (see below).

2. MDP is finite if S, A are finite else it is infinite. MDP is called discrete if S, A are discrete space else it is continuous.

3. Given a starting state s_0, an agent can take an action a according to some function, to modify the state to s_1 and receive a reward of r_1 in the process. It can then take another action and so on, thereby generating a sequence of state, action, reward $\tau = \langle s_0, a_1, r_1, s_1, a_2, r_2, s_2 \cdots a_k, r_k, s_k \rangle$ called a rollout τ. Here $r_i = R^{a_i}_{s_{i-1},s_i}$ and s_i is sampled with probability $P^{a_i}_{s_{i-1},s_i}$ given a_i, s_{i-1}. Length of this rollout is k (equal to number of actions).

4. MDP encodes Markov assumption in the form that reward and transition probability are independent of history of actions and states, given the last state. If this assumption is removed, the generalization is called Contextual Decision Process (CDP).

5. MDP defines a task where an agent has to maximize expected discounted reward. Formally, given a distribution over initial state $s_0 \sim \rho(s)$, the objective of the agent is to maximize J where:

$$J = E_{s_0 \sim \rho(s)}[\sum_{t \geq 0} \gamma^t r_{t+1} \mid s_0] \tag{1}$$

   r_t is the reward at time step t and is discounted by a factor of γ^t.

6. A task is episodic if every rollout terminates after a finite number of steps otherwise the task is called non-ending task. For an episodic task, the supremum of the number of steps is called the time horizon. For a non-ending task, $\frac{1}{1-\gamma}$ is called the effective time horizon.

7. Agent can only maximize the objective based on its choice of action. One way to encode this is in the form a policy $\pi : S \times A \rightarrow [0, 1]$ which denotes a probability distribution over action given a state s. For a given policy π, the above objective can be computed and is denoted as J^π. When policy is deterministic then we use the notation $\pi(s)$ to denote the action in state s.

8. Solving an MDP or reinforcement learning problem means finding a policy that optimizes J. Formally, we want to find the policy (aka control) $\arg\max_\pi J^\pi$.

9. An episodic MDP task with time horizon 1 is called a Contextual Bandit setting.

10. State value function of a policy (denoted V^π) is the function $S \rightarrow \mathbb{R}$ that gives the expected total discounted reward received on following policy π given a starting state $s \in S$.

$$V^\pi(s) = E[\sum_{t \geq 0} \gamma^t r_{t+1} \mid s_0, \pi] \tag{2}$$

11. Similarly state-action value function of a policy is the function $S \times A \rightarrow \mathbb{R}$ that gives the expected total discounted reward received on following policy π given a starting state s and an action a that is performed in the state.

$$Q^\pi(s,a) = E[\sum_{t \geq 0} \gamma^t r_{t+1} \mid s_0, a, \pi] \tag{3}$$

12. $J^\pi = \sum_s \rho(s) V^\pi(s)$

13. $V^\pi(s) = \sum_a \pi(s,a) Q^\pi(s,a)$ implying together with 12 that $J^\pi = \sum_s \rho(s) \sum_a \pi(s,a) Q^\pi(s,a)$

14. $V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^\pi(s')\}$

    $Q^\pi(s,a) = \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^\pi(s')\}$

    $Q^\pi(s,a) = \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma \sum_{a'} \pi(s,a') Q^\pi(s',a')\}$ (from previous equation and 12)

    [**Bellman self consistency equations**]

15. A policy π_1 is as good as policy π_2 (denoted $\pi_1 \geq \pi_2$) iff $V^{\pi_1}(s) \geq V^{\pi_2}(s)$ $\forall s \in S$. If the inequality is strict even for a single state then π_1 is strictly better than π_2.

16. For every MDP, there exists at least one policy π^* such that $\pi^* \geq \pi$ $\forall \pi$. It can be further shown that atleast one deterministic optimal policy also exists. Denote $V^* = V^{\pi^*}$ and $Q^* = Q^{\pi^*}$. Note that all optimal policies have the same state and action-value value functions.

17. $\arg\max Q^*(s,a)$ is an optimal deterministic policy.

18. If $\pi_2(s) = \arg\max_a Q^{\pi_1}(s,a)$ then it can be shown that $\pi_2 \geq \pi_1$. [**Policy Improvement**]

19. $V^*(s) = \max_a Q^*(s,a)$

    $V^*(s) = \max_a \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma V^*(s')\}$

    $Q^*(s,a) = \sum_{s'} P^a_{s,s'} \{R^a_{s,s'} + \gamma \max_{a'} Q^*(s',a')\}$

    [**Bellman optimality equations**]

    One way to get intuition of the first optimality equation is that if we keep on doing policy improvement, then at some point we can no longer improve the policy that is $\pi_2(s) = \arg\max_a Q^{\pi_1}(s,a)$ is same as π_1 and we get $\pi_1(s) = \arg\max_a Q^{\pi_1}(s,a)$. This denotes the optimality condition.

20. Given a policy π and starting state s_0, the policy gradient objective is:

    $$J = V^\pi(s_0) \tag{4}$$

    then

    $$\nabla J = \sum_s d^\pi(s; s_0) \sum_a \nabla \pi(s,a) Q^\pi(s,a) \tag{5}$$

    where $d^\pi(s) = \sum_{t \geq 0} \gamma^t P(s_t = s; \pi, s_0)$ is the discounted state visitation distribution with $P(s_t = s; \pi, s_0)$ denoting the probability of reaching state s after t steps. [**Policy Gradient Theorem**] Further it can be shown that

    $$J = V^\pi(s_0) = \sum_s d^\pi(s; s_0) \sum_a \pi(s,a) R^a_{s,s'} \tag{6}$$