Optimality in Policies

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

September 2, 2021

Overview

Review

Markov Decision Process

Markov decision process is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ where

- $\mathcal{S}$: (Finite) set of states
- $\mathcal{A}$: (Finite) set of actions
- $\mathcal{P}$: State transition probability

$$\mathcal{P}^a_{ss'} = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a), a_t \in \mathcal{A}$$

- $\mathcal{R}$: Reward for taking action a_t at state s_t and transitioning to state s_{t+1} is given by the deterministic function $\mathcal{R}$

$$r_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1})$$

- γ : Discount factor such that $\gamma \in [0, 1]$

Policy

Let π denote a policy that maps state space $\mathcal{S}$ to action space $\mathcal{A}$

Policy

- Deterministic policy: $a = \pi(s), s \in \mathcal{S}, a \in \mathcal{A}$
- Stochastic policy $\pi(a|s) = P[a_t = a|s_t = s]$

Value Functions with Policy

Given a MDP and a policy π, we define the value of a policy as follows :

State-value function

The value function $V^{\pi}(s)$ in state s is the expected (discounted) total return starting from state s and then following the policy π

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s\right)$$

The state-value function can be decomposed into immediate reward plus discounted value of successor state

$$V^{\pi}(s) = \mathbb{E}_{\pi}(r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s)$$

Action Value Function

Action Value Function

Action-value function

The action-value function $Q(s, a)$ under policy π is the expected return starting from state s and taking action a and then following the policy π

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a\right)$$

The action-value function can similarly be decomposed as

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}(r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) | s_t = s, a_t = a)$$

Expanding the expectation we have $Q^{\pi}(s, a)$ to be

$$Q^{\pi}(s, a) = \sum_{s'} \mathcal{P}_{ss'}^{a}\left[\mathcal{R}_{ss'}^{a} + \gamma \sum_{a'} \pi(a'|s') Q^{\pi}(s', a')\right]$$

Relationship between $V^\pi(\cdot)$ and $Q^\pi(\cdot)$

Using definitions of $V^\pi(s)$ and $Q^\pi(s,a)$, we can arrive at the following relationships

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^\pi(s,a)$$

$$Q^\pi(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]$$

Optimality in Policies

Optimal Policy

Define a partial ordering over policies

$$\pi \geq \pi', \quad \text{if} \quad V^{\pi}(s) \geq V^{\pi'}(s), \quad \forall s \in \mathcal{S}$$

Theorem

- There exists an optimal policy π_* that is better than or equal to all other policies.
- All optimal policies achieve the optimal value function, $V_*(s) = V^{\pi_*}(s)$
- All optimal policies achieve the optimal action-value function, $Q_*(s,a) = Q^{\pi_*}(s,a)$

Solution to an MDP

Solving an MDP <u>means</u> finding a policy π_* as follows

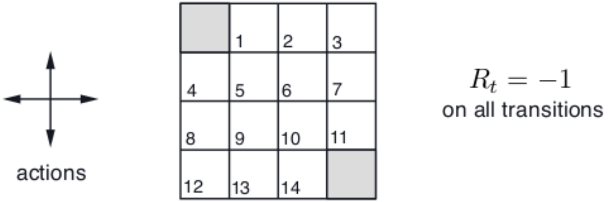
$$\pi_* = \arg\max_{\pi} \left[\mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \gamma^t r_{t+1} \right) \right]$$

is **maximum**

- The main goal in RL or solving an MDP means finding an **optimal value function** V_* or **optimal action value function** Q_* or **optimal policy** π_*

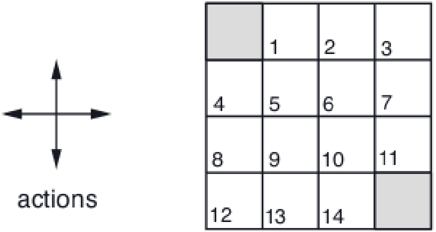
Grid World Problem

Consider a 4×4 grid world problem



- $\mathcal{S}$: $\{1, 2, \cdots, 14\}$ (non-terminal) + 2 terminal states (shaded)
- $\mathcal{A}$: {East, West, North, South}
- $\mathcal{P}$: Upon choosing an action from $\mathcal{A}$, state transitions are deterministic; except the actions that would take the agent off the grid in fact leave the state unchanged
- $\mathcal{R}$: Reward is -1 on all transitions until the terminal state is reached

Grid World Problem

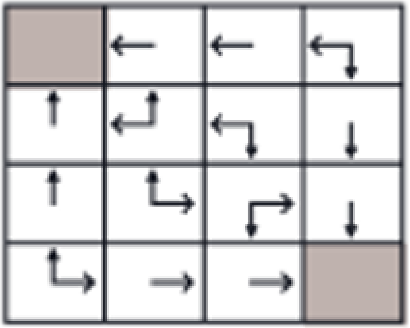


$R_t = -1$
on all transitions

Goal : Reach any of the goal state in as minimum plays as possible

Question : What could be an optimal policy to achieve the above objective ?

Grid World Problem : Optimal Policies



Question : How many optimal policies are there ?

Answer : There are infinite optimal policies (including some deterministic ones)

Towards Finding an Optimal Policy

Finding an Optimal Policy

<u>Question</u> : Suppose we are given $Q_*(s,a)$. Can we find an optimal policy ?

<u>Answer</u> : An optimal policy can be found by maximising over $Q_*(s,a)$

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

- If we know $Q_*(s,a)$, we immediately have an optimal policy
- There is always a <u>deterministic optimal policy</u> for any MDP

Relationship between $V_*(\cdot)$ and $Q_*(\cdot, \cdot)$

<u>**Question**</u> : Suppose we are given $Q_*(s, a), \forall s \in \mathcal{S}$. Can we find $V_*(s)$?

$$V_*(s) = \max_a Q_*(s, a)$$

<u>**Question**</u> : Suppose we are given $V_*(s), \forall s \in \mathcal{S}$. Can we find $Q_*(s, a)$?

$$Q_*(s, a) = \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

Towards Optimal Value Functions

Optimality Equation for State Value Function

Recall the Bellman Evaluation Equation for an MDP with policy π

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]$$

Question : Can we have a recursive formulation for $V_*(s)$?

$$V_*(s) = \max_a Q_*(s,a) = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s')\right)\right]$$

Optimality Equation for Action-Value Function

Similarly, there is a recursive formulation for $Q_*(\cdot, \cdot)$

$$Q_*(s, a) = \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_*(s', a') \right) \right]$$

Question : These are also a system of equations with $n = |\mathcal{S}|$ with n variables. Can we solve them ?

Answer : Optimality equations are non-linear system of equations with n unknowns and n non-linear constraints (i.e., the max operator).

Solving the Bellman Optimality Equation

- Bellman optimality equations are non-linear
- In general, there are no closed form solutions
- Iterative methods are typically used
- Exact and Approximate methods
  - ★ Exact methods (Model based) : Value iteration and Policy Iteration
  - ★ Approximate methods (Model free) : Q-learning and variants

Bellman Optimality Principle

Principle of Optimality

The tail of an optimal policy must be optimal

$$\textbf{OPT} \quad = \quad \textbf{HEAD} \quad + \gamma \ \textbf{TAIL (=OPT)}$$

- Any optimal policy can be subdivided into two components; an optimal first action, followed by an optimal policy from successor state s'.

Solution Methodology : Dynamic Programming

Bellman optimality equation :

$$V_*(s) = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

Optimal Substructure : Optimal solution can be decomposed into subproblems

Overlapping Subproblems : Value functions stores and reuses solutions

- Markov Decision Processes, generally, satisfy both these characterstics
- Dynamic Programming is a popular solution method for problems having such properties

Value Iteration Algorithm

Value Iteration : Idea

- Suppose we know the value $V_*(s')$
- Then the solution $V_*(s)$ can be found by one step look ahead

$$V_*(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

- Idea of value iteration is to perform the above updates iteratively

Value Iteration : Algorithm

Algorithm Value Iteration

1: Start with an initial value function $V_1(\cdot)$;
2: **for** $k = 1, 2, \cdots, K$ **do**
3: **for** $s \in \mathcal{S}$ **do**
4: Calculate
$$V_{k+1}(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s') \right) \right]$$
5: **end for**
6: **end for**

Value Iteration : Example

No noise and discount factor $\gamma = 1$

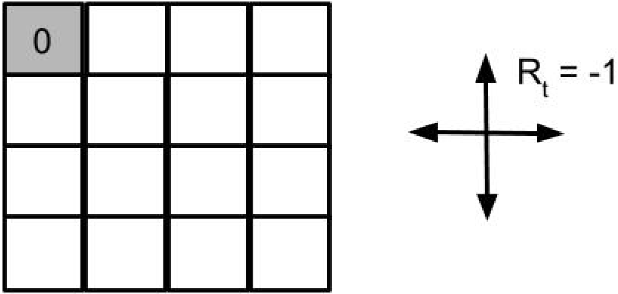


Figure Source: David Silver's UCL Course

Value Iteration : Example

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

Value Iteration : Remarks

- The sequence of value functions $\{V_1, V_2, \cdots,\}$ converge
- It converges to V_*
- Convergence is independent of the choice of V_1.
- Intermediate value functions need not correspond to a policy in the sense of satisfying the Bellman Evaluation Equation
- However, for any k, one can come up with a greedy policy as follows

$$\pi_{k+1}(s) \leftarrow \text{greedy} V_k(s)$$

Optimality Equation for Action-Value Function

There is a recursive formulation for $Q_*(\cdot,\cdot)$

$$Q_*(s,a) = \left[\sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\left(\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_*(s',a')\right)\right]$$

One could similarly conceive an iterative algorithm to compute optimal Q_* using the above recursive formulation !!