Exact Methods : Policy and Value Iteration

Easwar Subramanian

TCS Innovation Labs, Hyderabad

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

August 23, 2022

Overview

1. Review
2. Policy Iteration
3. Value Iteration

Review

Solution to an MDP

Solving an MDP means 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**

- Denote optimal value function $V_*(s) = V^{\pi_*}(s)$
- Denote optimal action value function $Q_*(s,a) = Q^{\pi_*}(s,a)$
- 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_*

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, $Q_*(s,a) = Q^{\pi_*}(s,a)$

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

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

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

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

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

Greedy Policy

For a given $Q^\pi(\cdot,\cdot)$, define $\pi'(s)$ as follows

$$\pi'(s) = \text{greedy}(Q) = \begin{cases} 1 & \text{if } a = \arg\max_{a\in\mathcal{A}} Q^\pi(s,a) \\ 0 & \text{Otherwise} \end{cases}$$

For a given $V^\pi(\cdot)$, define $\pi'(s)$ as follows

$$\pi'(s) = \text{greedy}(V) = \begin{cases} 1 & \text{if } a = \arg\max_{a\in\mathcal{A}} \left[\sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right)\right] \\ 0 & \text{Otherwise} \end{cases}$$

Policy Iteration

Policy Iteration

Question : Is there a way to arrive at π_* starting from an arbitrary policy π ?

Answer : Policy Iteration

- Evaluate the policy π
  - ★ Compute $V^{\pi}(s) = \mathbb{E}_{\pi}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s)$
- Improve the policy π

$$\pi'(s) = \text{greedy}(V^{\pi}(s))$$

$$\pi_0 \xrightarrow{\text{E}} V^{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} V^{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi^* \xrightarrow{\text{E}} V^*,$$

Policy Evaluation

- **Problem** : Evaluate a given policy π
- Compute $V^{\pi}(s) = \mathbb{E}_{\pi}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s)$
- **Solution 1** : Solve a system of linear equations using any solver
- **Solution 2** : Iterative application of Bellman Evaluation Equation
- Iterative update rule :

$$V^{\pi}_{k+1}(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V^{\pi}_{k}(s')\right]$$

- The sequence of value functions $\{V^{\pi}_1, V^{\pi}_2, \cdots, \}$ converge to V^{π}

Policy Improvement

Suppose we know V^π. How to improve policy π ?

The answer lies in the definition of action value function $Q^\pi(s,a)$. Recall that,

$$
\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\right) \\
&= \mathbb{E}(r_{t+1} + \gamma V^\pi(s_{t+1})|s_t = s, a_t = a) \\
&= \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]
\end{aligned}
$$

- If $Q^\pi(s,a) > V^\pi(s) \implies$ Better to select action a in state s and thereafter follow the policy π
- This is a special case of the policy improvement theorem

Policy Improvement Theorem

Theorem

Let π and π' be any pair of deterministic policies such that, for all $s \in \mathcal{S}$,

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s).$$

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all $s \in \mathcal{S}$

Proof.

$$\begin{aligned}
V^{\pi}(s) &\leq Q^{\pi}(s, \pi'(s)) = \mathbb{E}_{\pi'}(r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma Q^{\pi}(s_{t+1}, \pi'(s_{t+1})) | s_t = s) \\
&= \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 V^{\pi}(s_{t+2}) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 Q^{\pi}(s_{t+2}, \pi'(s_{t+2})) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s) = V^{\pi'}(s)
\end{aligned}$$

□

Policy Improvement

- Now consider the greedy policy $\pi' = \text{greedy}(V^\pi)$.
- Then, $\pi' \geq \pi$. That is, $V^{\pi'}(s) \geq V^\pi(s)$ for all $s \in \mathcal{S}$.
  - ★ By defintion of π', at state s, the action chosen by policy π' is given by the greedy operator
  $$\pi'(s) = \arg\max_a Q^\pi(s,a)$$
  - ★ This improves the value from any state s over one step
  $$Q^\pi(s, \pi'(s)) = \max_a Q^\pi(s,a) \geq Q^\pi(s, \pi(s)) = V^\pi(s)$$
  - ★ It therefore improves the value function, $V^{\pi'}(s) \geq V^\pi(s)$
- Policy π' is at least as good as policy π

Figure Source: Refer to David Silver Lecture 3 slides for a more detailed proof

Policy Improvement

- If improvements stop,

$$Q^{\pi}(s, \pi'(s)) = \max_a Q^{\pi}(s, a) = Q^{\pi}(s, \pi(s)) = V^{\pi}(s)$$

- Bellman optimality equation is satisfied as,

$$V^{\pi}(s) = \max_a Q^{\pi}(s, a)$$

- The policy π for which the improvement stops is the optimal policy.

$$V^{\pi}(s) = V_*(s) \quad \forall s \in \mathcal{S}$$

Policy Iteration : Algorithm

Algorithm Policy Iteration

1: Start with an initial policy π_1
2: **for** $i = 1, 2, \cdots, N$ **do**
3: Evaluate $V^{\pi_i}(s) \quad \forall s \in \mathcal{S}$. That is,
4: **for** $k = 1, 2, \cdots, K$ **do**
5: For all $s \in \mathcal{S}$ calculate

$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V_k^{\pi_i}(s')\right]$$

6: **end for**
7: Perform policy Improvement

$$\pi_{i+1} = \text{greedy}(V^{\pi_i})$$

8: **end for**

Update Rule :

$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V_k^{\pi_i}(s')\right]$$

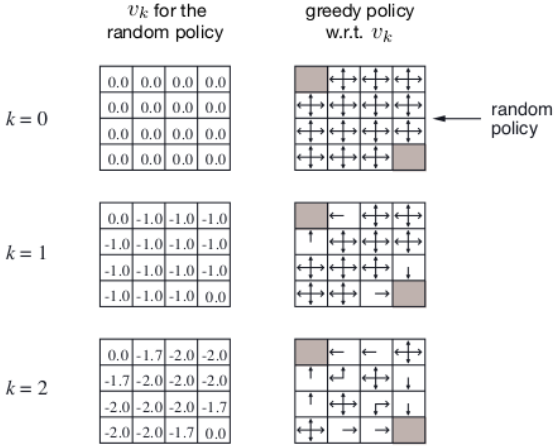


Figure Source: David Silver's UCL course

Policy Iteration : Example

$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

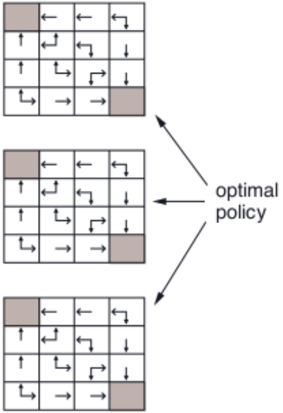
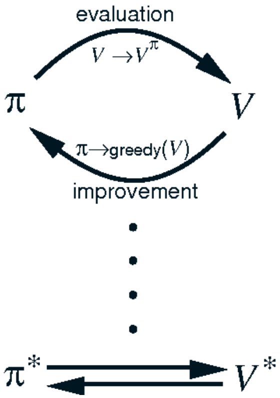
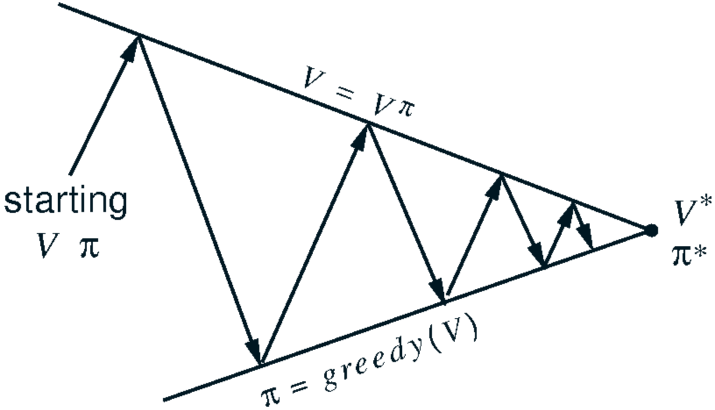


Figure Source: David Silver's UCL course

Policy Iteration : Schematic Representation





- The sequence $\{\pi_1, \pi_2, \cdots,\}$ is guaranteed to converge.
- At convergence, both current policy and the value function associated with the policy are optimal.

Figure Source: David Silver's UCL course

Modified Policy Iteration

Can we computationally simplify policy iteration process ?

- We need not wait for policy evaluation to converge to V^π
- We can have a stopping criterion like ϵ-convergence of value function evaluation or K iterations of policy evaluation
- Extreme case of $K = 1$ is **value iteration**. We update the policy every iteration

Value Iteration

Value Iteration

Question : Is there a way to arrive at V_* starting from an arbitrary value function V_0 ?

Answer : Value Iteration

Optimality Equation for State Value Function

Recall the Bellman Evaluation Equation for an MDP with policy π

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

<u>Question</u> : 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

Bellman's Optimality Principle

Principle of Optimality

The tail of an optimal policy must be optimal

- 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 constructed from optimal solutions to subproblems

Overlapping Subproblems : Problem can be broken down into subproblems and can be reused several times

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

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

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**