

10-301/601: Introduction to Machine Learning

Lecture 22: Value and Policy Iteration

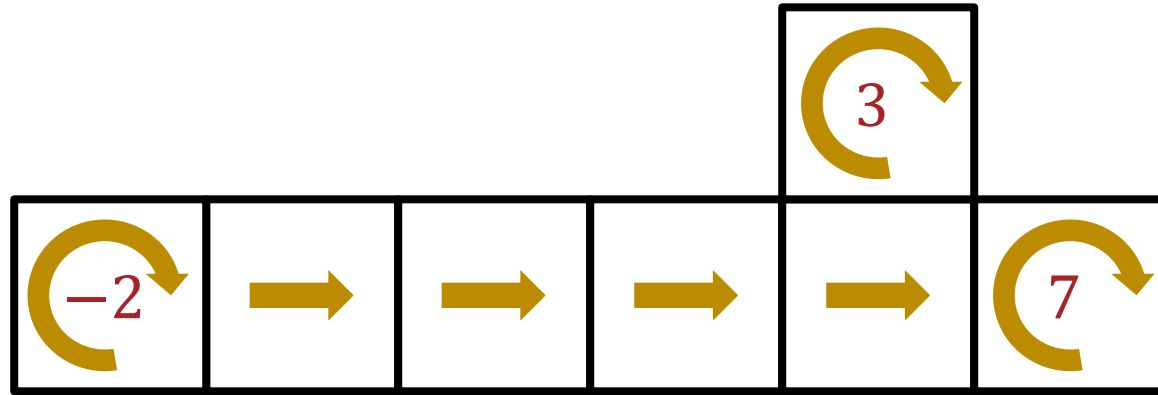
Henry Chai

7/9/24

Front Matter

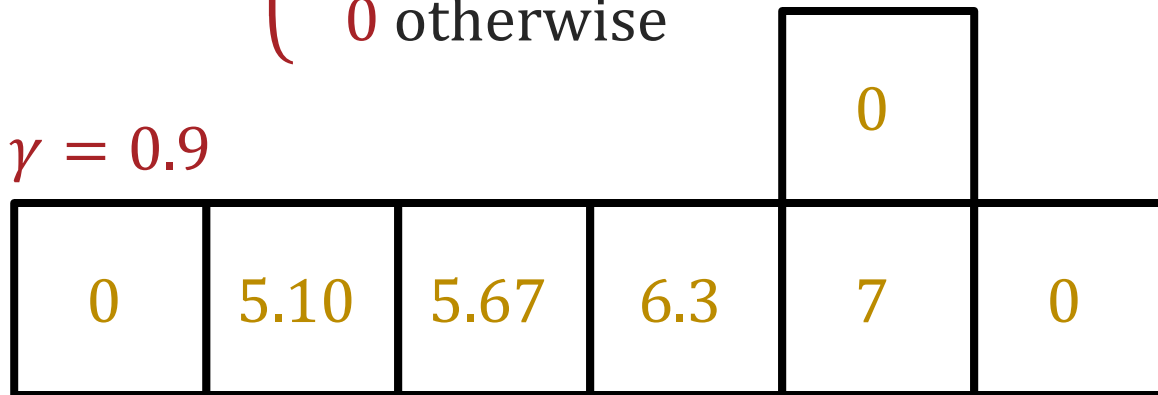
- Announcements
 - HW7 released 7/3, due 7/11 at 11:59 PM
- Recommended Readings
 - Mitchell, Chapter 13

Recall: Value Function Example



$$R(s, a) = \begin{cases} -2 & \text{if entering state 0 (safety)} \\ 3 & \text{if entering state 5 (field goal)} \\ 7 & \text{if entering state 6 (touch down)} \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma = 0.9$$



Value Function

- $V^\pi(s) = \mathbb{E}[\text{discounted total reward of starting in state } s \text{ and executing policy } \pi \text{ forever}]$
$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$$
$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$$
$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) (R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \dots \mid s_1])$$

Value Function

- $V^\pi(s) = \mathbb{E}[\text{discounted total reward of starting in state } s \text{ and executing policy } \pi \text{ forever}]$
 $= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$
 $= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$
 $= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) (R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \dots \mid s_1])$

Value Function

- $V^\pi(s) = \mathbb{E}[\text{discounted total reward of starting in state } s \text{ and executing policy } \pi \text{ forever}]$
 $= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$
 $= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \dots \mid s_0 = s]$
 $= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) (R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \dots \mid s_1])$

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) V^\pi(s_1)$$

Bellman equations

Optimality

- Optimal value function:

$$V^*(s) = \max_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^*(s')$$

- System of $|\mathcal{S}|$ equations and $|\mathcal{S}|$ variables

- Optimal policy:

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^*(s')$$

Immediate
reward

(Discounted)
Future reward

Fixed Point Iteration

- Iterative method for solving a system of equations
- Given some equations and initial values

$$x_1 = f_1(x_1, \dots, x_n)$$

⋮

$$x_n = f_n(x_1, \dots, x_n)$$

$$x_1^{(0)}, \dots, x_n^{(0)}$$

- While not converged, do

$$x_1^{(t+1)} \leftarrow f_1(x_1^{(t)}, \dots, x_n^{(t)})$$

⋮

$$x_n^{(t+1)} \leftarrow f_n(x_1^{(t)}, \dots, x_n^{(t)})$$

Fixed Point Iteration: Example

$$x_1 = x_1 x_2 + \frac{1}{2}$$

$$x_2 = -\frac{3x_1}{2}$$

$$x_1^{(0)} = x_2^{(0)} = 0$$

$$\hat{x}_1 = \frac{1}{3}, \hat{x}_2 = -\frac{1}{2}$$

t	$x_1^{(t)}$	$x_2^{(t)}$
0	0	0
1	0.5	0
2	0.5	-0.75
3	0.125	-0.75
4	0.4063	-0.1875
5	0.4238	-0.6094
6	0.2417	-0.6357
7	0.3463	-0.3626
8	0.3744	-0.5195
9	0.3055	-0.5616
10	0.3284	-0.4582
11	0.3495	-0.4926
12	0.3278	-0.5243
13	0.3281	-0.4917
14	0.3386	-0.4922
15	0.3333	-0.5080

Value Iteration

- Inputs: $R(s, a), p(s' | s, a)$
- Initialize $V^{(0)}(s) = 0 \forall s \in \mathcal{S}$ (or randomly) and set $t = 0$
- While not converged, do:

- For $s \in \mathcal{S}$

$$V^{(t+1)}(s) \leftarrow \max_{a \in \mathcal{A}} R(s, a) + \underbrace{\gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')}_{Q(s, a)}$$

- $t = t + 1$

- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')$$

- Return π^*

Value Iteration

- Inputs: $R(s, a), p(s' | s, a)$
- Initialize $V^{(0)}(s) = 0 \forall s \in \mathcal{S}$ (or randomly) and set $t = 0$
- While not converged, do:

- For $s \in \mathcal{S}$

- For $a \in \mathcal{A}$

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')$$

- $V^{(t+1)}(s) \leftarrow \max_{a \in \mathcal{A}} Q(s, a)$

- $t = t + 1$

- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')$$

- Return π^*

What is the runtime of one iteration of value iteration?

$$O(|\mathcal{S}||\mathcal{A}|)$$

0%

$$O(|\mathcal{S}|^2|\mathcal{A}|)$$

0%

$$O(|\mathcal{S}||\mathcal{A}|^2)$$

0%

$$O(|\mathcal{S}|^2|\mathcal{A}|^2)$$

0%

Synchronous Value Iteration

- Inputs: $R(s, a), p(s' | s, a)$
- Initialize $V^{(0)}(s) = 0 \forall s \in \mathcal{S}$ (or randomly) and set $t = 0$
- While not converged, do:

- For $s \in \mathcal{S}$

- For $a \in \mathcal{A}$

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')$$

- $V^{(t+1)}(s) \leftarrow \max_{a \in \mathcal{A}} Q(s, a)$

- $t = t + 1$

- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^{(t)}(s')$$

- Return π^*

Asynchronous Value Iteration

- Inputs: $R(s, a), p(s' | s, a)$
- Initialize $V^{(0)}(s) = 0 \forall s \in \mathcal{S}$ (or randomly)
- While not converged, do:

- For $s \in \mathcal{S}$

- For $a \in \mathcal{A}$

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V(s')$$

- $V(s) \leftarrow \max_{a \in \mathcal{A}} Q(s, a)$

- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V(s')$$

- Return π^*

Value Iteration Theory

- **Theorem 1:** Value function convergence

V will converge to V^* if each state is “visited”
infinitely often (Bertsekas, 1989)

- **Theorem 2:** Convergence criterion

$$\text{if } \max_{s \in \mathcal{S}} |V^{(t+1)}(s) - V^{(t)}(s)| < \epsilon,$$

then $\max_{s \in \mathcal{S}} |V^{(t+1)}(s) - V^*(s)| < \frac{2\epsilon\gamma}{1-\gamma}$ (Williams & Baird, 1993)

- **Theorem 3:** Policy convergence

The “greedy” policy, $\pi(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q(s, a)$, converges to the optimal π^* in a finite number of iterations, often before the value function has converged! (Bertsekas, 1987)

Policy Iteration

- Inputs: $R(s, a), p(s' | s, a)$
- Initialize π randomly
- While not converged, do:
 - Solve the Bellman equations defined by policy π

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, \pi(s)) V^\pi(s')$$

- Update π

$$\pi(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' | s, a) V^\pi(s')$$

- Return π

Policy Iteration Theory

- In policy iteration, the policy improves in each iteration.
- Given finite state and action spaces, there are finitely many possible policies
- Thus, the number of iterations needed to converge is bounded!
- Value iteration takes $O(|S|^2|A|)$ time / iteration
- Policy iteration takes $O(|S|^2|A| + |S|^3)$ time / iteration
 - However, empirically policy iteration requires fewer iterations to converge

Two big Q's

1. What can we do if the reward and/or transition functions/distributions are unknown?
2. How can we handle infinite (or just very large) state/action spaces?

Key Takeaways

- If the reward and transition functions are known, we can solve for the optimal policy (and value function) using value or policy iteration
 - Both algorithms are instances of fixed point iteration and are guaranteed to converge (under some assumptions)