10-301/601: Introduction to Machine Learning Lecture 21: Value and Policy Iteration

Henry Chai & Matt Gormley

11/13/23

Front Matter

- Announcements
  - HW7 released 11/10, due 11/20 at 11:59 PM
    - Please be mindful of your grace day usage (see the course syllabus for the policy)

Recall: Reinforcement Learning Objective Function

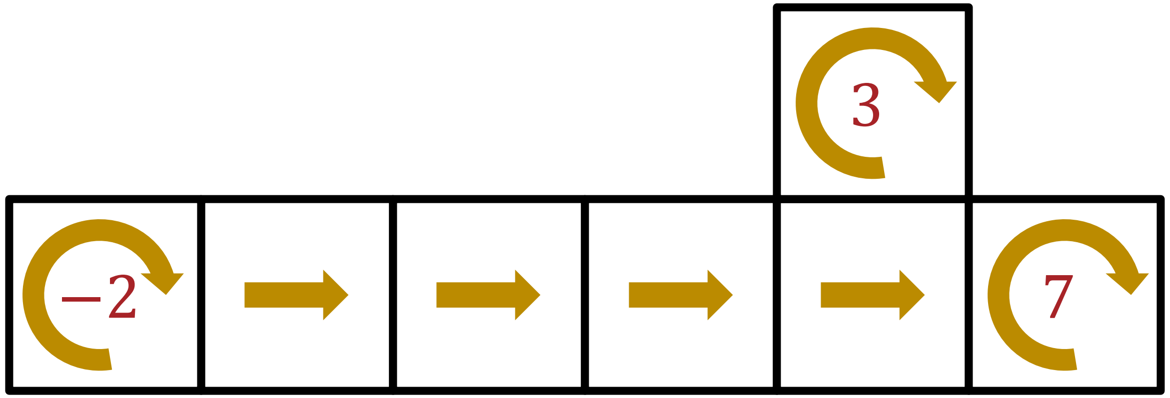
- Find a policy $\pi^* = \underset{\pi}{\operatorname{argmax}}\ V^\pi(s)\ \forall\, s \in \mathcal{S}$
- Assume stochastic transitions and deterministic rewards
- $V^\pi(s) = \mathbb{E}[$*discounted* total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}_{p(s' \mid s,a)}\left[R\big(s_0 = s, \pi(s_0)\big) + \gamma R\big(s_1, \pi(s_1)\big) + \gamma^2 R\big(s_2, \pi(s_2)\big) + \cdots\right]$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p(s' \mid s,a)}\left[R\big(s_t, \pi(s_t)\big)\right]$$

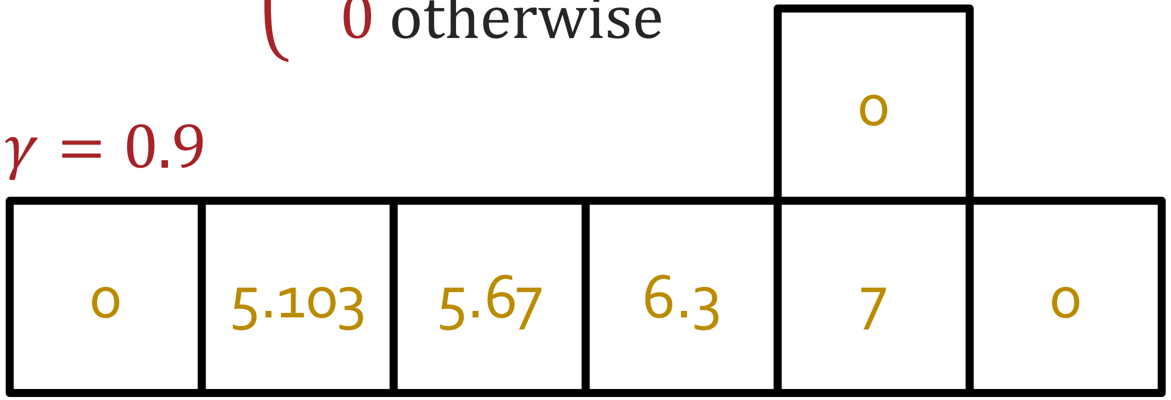
where $0 \leq \gamma < 1$ is some discount factor for future rewards

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

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) \Big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \cdots \mid s_1] \Big)$$

Value Function

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \textstyle\sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) \Big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}\big[R(s_2, \pi(s_2)) + \cdots \mid s_1\big] \Big)$$

$$\underbrace{\mathrm{V}^{\pi}(\mathrm{s}) = R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) V^{\pi}(s_1)}_{\text{Bellman equations}}$$

Optimality

- Optimal value function:

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

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

- Optimal policy:

$$\pi^*(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \underbrace{R(s,a)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^*(s')}_{\text{(Discounted) Future reward}}$$

- Insight: if you know the optimal value function, you can solve for the optimal policy!

Fixed Point Iteration

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

$$x_1 = f_1(x_1, \dots, x_n)$$
$$\vdots$$
$$x_n = f_n(x_1, \dots, x_n)$$
$$x_1^{(0)}, \dots, x_n^{(0)}$$

- While not converged, do

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

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

Value Iteration

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

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

  - $t = t + 1$
- For $s \in \mathcal{S}$

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

- Return π^*

Synchronous Value Iteration

- Inputs: $R(s,a)$, $p(s' \mid s,a)$
- 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 $a \in \mathcal{A}$

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

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

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

- Return π^*

Asynchronous Value Iteration

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

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

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

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

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

- 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

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

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

- **Theorem 3**: Policy convergence

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

Policy Iteration

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

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

  - Update π

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

- 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!

- Policy iteration takes $O(|\mathcal{S}|^2|\mathcal{A}| + |\mathcal{S}|^3)$ time / iteration
  - However, empirically policy iteration requires fewer iterations to converge than value iteration

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?

MDP and Value/Policy Iteration Learning Objectives

You should be able to...

- Compare reinforcement learning to other learning paradigms
- Cast a real-world problem as a Markov Decision Process
- Depict the exploration vs. exploitation tradeoff via MDP examples
- Explain how to solve a system of equations using fixed point iteration
- Define the Bellman Equations
- Show how to compute the optimal policy in terms of the optimal value function
- Explain the relationship between a value function mapping states to expected rewards and a value function mapping state-action pairs to expected rewards
- Implement value iteration and policy iteration
- Contrast the computational complexity and empirical convergence of value iteration vs. policy iteration
- Identify the conditions under which the value iteration algorithm will converge to the true value function
- Describe properties of the policy iteration algorithm