Guided Tour of Machine Learning in Finance

Week 4: Reinforcement Learning

4-2-4-Bellman-equation-and-RL

Igor Halperin

NYU Tandon School of Engineering, 2017

The optimal value function

- The value function for policy π

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots \mid s_0 = s, \pi \right]$$

- The Bellman equation for value function

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

The optimal value function

- The value function for policy π

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots \mid s_0 = s, \pi \right]$$

- The Bellman equation for value function

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

- The **optimal value function:** $V^*(s) = \max_{\pi} V^{\pi}(s)$

The optimal value function

- The value function for policy π

$$V^{\pi}(s) = \mathbb{E}\Big[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots \mid s_0 = s, \pi\Big]$$

- The Bellman equation for value function

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

- The **optimal value function:** $V^*(s) = \max_{\pi} V^{\pi}(s)$

- The **Bellman equation for optimal value function** $V^*(s)$

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

The optimal value function

- The value function for policy π

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots \mid s_0 = s, \pi \right]$$

- The Bellman equation for value function

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

- The **optimal value function:** $V^*(s) = \max_{\pi} V^{\pi}(s)$
- The **Bellman equation for optimal value function** $V^*(s)$

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

- The **optimal policy:**

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s' \mid s, a) V^*(s')$$

The optimal value function

- The value function for policy π

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots \mid s_0 = s, \pi \right]$$

- The Bellman equation for value function

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

- The **optimal value function:** $V^*(s) = \max_{\pi} V^{\pi}(s)$

- The **Bellman equation for optimal value function** $V^*(s)$

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

- The **optimal policy:**

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s' \mid s, a) V^*(s')$$

- Optimality means that

$$V^*(s) = V^{\pi^*}(s) \geq V^{\pi}(s), \boxed{\forall \pi \neq \pi^*}$$

Value iteration

- The **Bellman equation for optimal value function**

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

- **Value Iteration** algorithm (for discrete state-action space)**:**
  - Initialize the value function for each state $V(s) = V^{(0)}(s)$
  - Repeat the update of the value function until convergence:

$$V^{(k+1)}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} p(s' | s, a) V^{(k)}(s')$$

Value iteration

- The **Bellman equation for optimal value function**

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

- **Value Iteration** algorithm (for discrete state-action space)**:**
  - Initialize the value function for each state $V(s) = V^{(0)}(s)$
  - Repeat the update of the value function until convergence:

$$V^{(k+1)}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} p(s' | s, a) V^{(k)}(s')$$

- Synchronous updates: finish until the end of iteration, then update the value function for all states at once
- Asynchronous updates: update the value function on the fly

Value iteration

- The **Bellman equation for optimal value function**

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

- **Value Iteration** algorithm (for discrete state-action space)**:**
  - Initialize the value function for each state $V(s) = V^{(0)}(s)$
  - Repeat the update of the value function until convergence:

$$V^{(k+1)}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} p(s' | s, a) V^{(k)}(s')$$

- Synchronous updates: finish until the end of iteration, then update the value function for all states at once
- Asynchronous updates: update the value function on the fly

- Optimal policy: $\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s' | s, a) V^*(s')$

Policy iteration

- The Bellman equation for the value function

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

- **Policy Iteration** algorithm**:**
  - Initialize policy randomly $\pi(s) = \pi^{(0)}(s)$
  - Repeat the update of the value function until convergence:
    - Policy evaluation: Compute $V^{\pi}(s)$ from the Bellman equation for the value function
    - Iterate policy $\pi^{(k+1)}(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s' \mid s, a) V^{(\pi)}(s')$

Policy iteration

- The Bellman equation for the value function

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

- **Policy Iteration** algorithm**:**
  - Initialize policy randomly $\pi(s) = \pi^{(0)}(s)$
  - Repeat the update of the value function until convergence:
    - Policy evaluation: Compute $V^{\pi}(s)$ from the Bellman equation for the value function
    - Iterate policy $\pi^{(k+1)}(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s' | s, a) V^{(\pi)}(s')$

- The policy iteration step can be done using standard optimization software
- The policy evaluation is critical, as it requires solving Bellman equation multiply times - can be costly for large state spaces

Control question

Select all correct answers

1. The optimal policy is a policy for which the Bellman equation is fastest to solve.
2. The optimal policy is a policy that maximizes the value function.
3. The most computationally heavy part of Policy Iteration algorithms is a policy iteration step, because it involves two non-linear operators “arg” and “max”.
4. Synchronous updates in Value Iteration algorithm amount to simultaneous policy iteration updates for all visited points.

Correct answers: 2