Policy vs Value Iteration for Markov Decision Processes

Anna Androvitsanea

May 10, 2024

Policy Iteration

- Method for solving Markov Decision Processes (MDPs)
- Consists of two main steps:
  - Policy Evaluation
  - Policy Improvement

Problem Setup

- States: A, B
- Actions: X, Y
- Rewards:
  - State A: X (+1), Y (+0)
  - State B: X (+0), Y (+2)
- Transitions:
  - A $\xrightarrow{X}$ B (prob. 1), A $\xrightarrow{Y}$ A (prob. 1)
  - B $\xrightarrow{X}$ A (prob. 1), B $\xrightarrow{Y}$ B (prob. 1)
- Discount factor $\gamma = 0.9$

Objective

Find an optimal policy π that maximizes the expected return from any starting state.

Step 1: Initialization

Assume an initial policy π:

- $\pi(A) = X$
- $\pi(B) = X$

Step 2: Policy Evaluation

Compute the value of each state under the policy π:

$$v_\pi(A) = R(A, X) + \gamma v_\pi(B)$$
$$v_\pi(B) = R(B, X) + \gamma v_\pi(A)$$

Solving the system of linear equations:

- $v_\pi(A) \approx 5$
- $v_\pi(B) \approx 4.5$

Step 3: Policy Improvement

For each state, select the action that maximizes the expected return:

- State A: $Q(A, X) > Q(A, Y) \Rightarrow \pi(A) = X$
- State B: $Q(B, Y) > Q(B, X) \Rightarrow \pi(B) = Y$

Updated policy π:

- $\pi(A) = X$
- $\pi(B) = Y$

Step 4: Repeat Evaluation and Improvement

Continue evaluating and improving the policy until it stabilizes. In this case, the new policy would need to be evaluated again, and if no further improvements can be made, it becomes the optimal policy.

Value Iteration

- Method for solving Markov Decision Processes (MDPs)
- Combines policy evaluation and policy improvement into a single update process
- Iteratively improves the value function until convergence

MDP Setup Recap

- States: A, B
- Actions: X, Y
- Rewards:
  - A, X: $+1$
  - A, Y: $+0$
  - B, X: $+0$
  - B, Y: $+2$
- Transitions:
  - A, X $\rightarrow$ B
  - A, Y $\rightarrow$ A
  - B, X $\rightarrow$ A
  - B, Y $\rightarrow$ B
- Discount factor γ: 0.9

Objective

To find the optimal value function $V^*(s)$ and derive an optimal policy from it.

Step 1: Initialization

Initialize the value function for all states to 0:

- $V(A) = 0$
- $V(B) = 0$

Step 2: Iterative Update

Perform the update rule:

$$V_{k+1}(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V_k(s') \right]$$

Where:

- s' are possible next states
- $P(s'|s,a)$ is the transition probability
- $R(s,a)$ is the reward for taking action a in state s
- $V_k(s')$ is the value of state s' at iteration k

Step 3: Iterative Calculation Example

Compute a few iterations:

- Iteration 1:
  $V_1(A) = \max(1 + 0.9 \times V_0(B), 0 + 0.9 \times V_0(A)) =$
  $\max(1 + 0, 0) = 1$
  $V_1(B) = \max(0 + 0.9 \times V_0(A), 2 + 0.9 \times V_0(B)) = \max(0, 2) = 2$
- Iteration 2:
  $V_2(A) = \max(1 + 0.9 \times V_1(B), 0 + 0.9 \times V_1(A)) =$
  $\max(1 + 0.9 \times 2, 0 + 0.9 \times 1) = 2.8$
  $V_2(B) = \max(0 + 0.9 \times V_1(A), 2 + 0.9 \times V_1(B)) =$
  $\max(0 + 0.9 \times 1, 2 + 0.9 \times 2) = 3.8$
- Iteration 3:
  $V_3(A) = \max(1 + 0.9 \times V_2(B), 0 + 0.9 \times V_2(A)) =$
  $\max(1 + 0.9 \times 3.8, 0 + 0.9 \times 2.8) = 4.42$
  $V_3(B) = \max(0 + 0.9 \times V_2(A), 2 + 0.9 \times V_2(B)) =$
  $\max(0 + 0.9 \times 2.8, 2 + 0.9 \times 3.8) = 5.42$

Step 4: Convergence

Continue updates until the change in value function between iterations is below a small threshold ϵ, indicating convergence.

Step 5: Derive Policy

Once the value function has converged to V^*, derive the optimal policy π^* by choosing the action that maximizes the value at each state:

$$\pi^*(s) = \arg\max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]$$

Result

The resulting values and policy provide the optimal strategy for the agent in the defined MDP, where the actions chosen maximize the expected return from each state under the model constraints. Value Iteration is generally simpler to implement than Policy Iteration.

RETRO: BOTH METHODS

- Objective: Solve an MDP with states A and B, and actions X and Y.
- Actions:
  - A, X: Transition to B, Reward $+1$
  - A, Y: Stay in A, Reward 0
  - B, X: Transition to A, Reward 0
  - B, Y: Stay in B, Reward $+2$
- Discount factor $\gamma = 0.9$.
- Goal: Find an optimal policy that maximizes the expected return.

Methods Overview

Policy Iteration

1. Initialize a policy arbitrarily.
2. Policy Evaluation: Solve v_π.
3. Policy Improvement: Update policy.
4. Iterate until policy is stable.

Value Iteration

1. Initialize $V(s) = 0$ for all s.
2. Update $V(s)$ using the Bellman optimality equation.
3. Iterate until V converges.
4. Derive policy from V^*.

Policy Iteration: Example

- Initial policy $\pi(A) = X$, $\pi(B) = X$
- Iteration 1: Evaluate v_π, improve to $\pi(A) = X$, $\pi(B) = Y$
- Policy stabilizes after one improvement.
- Optimal values: $v_\pi(A) \approx 5$, $v_\pi(B) \approx 4.5$

Value Iteration: Example

- Initial values: $V(A) = 0,\ V(B) = 0$
- Iteration 1: $V(A) = 1,\ V(B) = 2$
- Iteration 2: $V(A) = 2.8,\ V(B) = 3.8$
- Convergence after a few iterations.
- Optimal values: $V^*(A) \approx 4.42,\ V^*(B) \approx 5.42$

Comparison and Conclusion

Aspect	Policy Iteration	Value Iteration
Complexity	Lower (fewer iterations)	Higher (more iterations)
Convergence	Fast (policy stable)	Gradual (value convergence
Implementation	Complex (two phases)	Simpler (single update)

Table: Comparative analysis of Policy Iteration and Value Iteration

- Both methods arrive at the same optimal values.
- Policy Iteration is generally faster but requires solving systems of equations.
- Value Iteration is straightforward but might require more iterations.