Markov Decision Process

By:

Bayat Vahabodin

Contents

Introduction

Markov Decision Processes (MDPs) are mathematical models used to represent decision-making problems in situations where outcomes are partially random and partially under the control of a decision maker.

The basic idea behind an MDP is to represent a decision-making problem as a set of states, actions, and rewards. The states represent the possible configurations of the system being modeled, the actions represent the decisions that the decision maker can take, and the rewards represent the benefits (or costs) associated with taking each action in each state. The key assumption of an MDP is that the probability of transitioning from one state to another depends only on the current state and the action taken, and not on the history of the system.

The solution to an MDP involves finding a policy that maximizes the expected total reward over time. A policy is a function that maps each state to an action, and the expected total reward is the sum of the rewards received at each time step, discounted by a factor that represents the degree of importance placed on future rewards.

Step 1: Displaying MDP graph

we displayed the MDP graph using the graphviz library. The MDP consisted of three states (s0, s1, s2), two actions (a0, a1), and transition probabilities for each action from one state to another. Additionally, we had two reward values for two specific transitions.

Step 2: Solving MDP using Value Iteration

There are several algorithms that can be used to solve an MDP, including value iteration and policy iteration. Value iteration is an iterative algorithm that involves repeatedly updating an estimate of the value of each state based on the values of its neighboring states.

We are using value iteration to solve an MDP with three states, two actions, and a set of transition probabilities and rewards. The goal is to find the optimal policy and the corresponding value function. The optimal policy is the policy that maximizes the expected total reward over time, and the value function is a function that assigns a value to each state based on the expected total reward that can be obtained from that state under the optimal policy.

We are also varying the values of some of the parameters, such as the discount factor gamma and the convergence threshold delta, to see how they affect the results. By comparing the results obtained with different parameter settings, you can gain insights into how the behavior of the system changes as these parameters are varied.

Models

Model 1: max_iterations=100, gamma = 0.8, and delta = 1e-6

Out:

- Optimal value function:
  {'s0': 1.4889091050931198, 's1': 5.034315652083617, 's2': 1.8611371937942818}
- Optimal policy:
  {'s0': 'a1', 's1': 'a0', 's2': 'a1'}
- Average reward:
  0.5333333333333333

Model 2: max_iterations=100, gamma = 0.9, and delta = 1e-6

Out:

- Optimal value function:
  {'s0': 3.789830037441072, 's1': 7.30280158776066, 's2': 4.21093543912048}
- Optimal policy:
  {'s0': 'a1', 's1': 'a0', 's2': 'a1'}
- Average reward:
  0.5333333333333333

Model 3: max_iterations=100, gamma = 0.9, and delta = 1e-9

Out:

- Optimal value function:
  {'s0': 3.789830037441072, 's1': 7.30280158776066, 's2': 4.21093543912048}
- Optimal policy:
  {'s0': 'a1', 's1': 'a0', 's2': 'a1'}
- Average reward:
  0.5333333333333333

Model 4: max_iterations=1000, gamma = 0.9, and delta = 1e-9

Out:

- Optimal value function:
  {'s0': 3.789948606985037, 's1': 7.302920157304626, 's2': 4.211054008664446}
- Optimal policy:
  {'s0': 'a1', 's1': 'a0', 's2': 'a1'}
- Average reward:
  0.5333333333333333

Conclusion

All of these four models are based on the same MDP problem and have the same reward structure. We set the maximum number of iterations to 100 and 100, the termination threshold (delta) to 1e-6 and 1e-9, and experimented with two different values of the discount factor (gamma), 0.8 and 0.9.

The results showed that the optimal policy and the average reward was the same for all values, and only the optimal value function differed for the two values of gamma. A higher value of gamma means that the agent puts more weight on future rewards and is more patient. The optimal value function for gamma=0.9 was higher than that for gamma=0.8 for all states, indicating that the agent is willing to wait longer for higher rewards in the future.

In conclusion, results show that the choice of gamma can have a significant impact on the optimal policy and the expected long-term reward. Therefore, it is important to carefully choose the discount factor depending on the application and the agent's preferences. Model 4 is the most accurate model due to its high number of maximum iterations and small delta value. However, it is also the least efficient model as it requires more iterations to converge. Model 2 and 3 offer a balance between accuracy and efficiency, while Model 1 is the least accurate model due to its small number of maximum iterations and low discount factor.