MDP Planning

DP

Dynamic Programming is a powerful technique for solving Markov Decision Processes (MDPs).

DP Algorithms

Value Iteration:

- **Goal:** Directly compute the optimal value function.
- **Process:**
  1. Initialize the value function arbitrarily.
  2. Iteratively update the value function using the Bellman equation until convergence.
  3. Once converged, extract the optimal policy from the value function.

Policy Iteration:

- **Goal:** Alternate between policy evaluation and policy improvement.
- **Process:**
  1. Initialize a policy arbitrarily.
  2. Evaluate the policy to compute the corresponding value function.
  3. Improve the policy by selecting the action with the highest Q-value for each state.
  4. Repeat steps 2 and 3 until convergence.

Example: Gridworld

Consider a simple gridworld where an agent can move up, down, left, or right. The goal is to reach a terminal state while maximizing the total reward.

- **States:** Grid cells.
- **Actions:** Move up, down, left, or right.
- **Transition Probabilities:** Assume deterministic transitions.
- **Rewards:** Positive reward for reaching the goal, negative reward for falling off the grid.

Using Value Iteration:

1. Initialize the value function for all states to 0.
2. Iteratively update the value function using the Bellman equation. For example, for a state , the updated value would be:

   where
   `a` is an action, `s'` is the next state, and `γ` is the discount factor.

   $$V'(s) = max(R(s,a) + \gamma V(s'))$$

3. Continue updating until the value function converges.
4. Once converged, extract the optimal policy by selecting the action with the highest Q-value for each state.

Using Policy Iteration:

1. Initialize a policy arbitrarily (e.g., always move right).
2. Evaluate the policy to compute the corresponding value function using policy evaluation.
3. Improve the policy by selecting the action with the highest Q-value for each state.
4. Repeat steps 2 and 3 until convergence.

Advantages of DP

- **Guaranteed convergence:** DP algorithms are guaranteed to converge to the optimal solution under certain conditions.
- **Efficient for small state spaces:** DP can be very efficient for problems with small state spaces.
- **Foundation for other algorithms:** DP serves as a foundation for many other reinforcement learning algorithms.

Disadvantages of Planning Methods

1. **Computational Complexity:** Planning methods can be computationally expensive, especially for large state spaces. This is because they require iteratively updating the value function or policy for all possible states.
2. **Need for Complete Knowledge:** Planning methods assume that the agent has complete knowledge of the environment, including the transition probabilities and rewards. In real-world scenarios, this assumption may not hold, as the environment may be partially observable or stochastic.
3. **Limited Applicability to Continuous State Spaces:** While planning methods can be extended to handle continuous state spaces using function approximation techniques, they may still be computationally challenging and require careful discretization or feature engineering.
4. **Lack of Online Learning:** Planning methods are typically offline, meaning they require pre-processing the environment and computing the optimal policy before interacting with it. This can be limiting in situations where the environment changes over time or where the agent needs to learn online.