July 19, 2025

# Introduction to Week 2 - Markov Decision Processes (MDPs)

## Overview

In this week, we will delve into the foundational concepts of Markov Decision Processes (MDPs), a crucial framework in decision-making and reinforcement learning. MDPs assist in modeling complex problems where outcomes are partly random and partly under the control of a decision-maker.

# Key Concepts of MDPs

1. **Definition of MDP:**
   - A tuple $(S, A, P, R, \gamma)$ where:
     - $S$: Finite set of states.
     - $A$: Finite set of actions available at each state.
     - $P$: Transition probability function.
     - $R$: Reward function for actions taken in states.
     - $\gamma$: Discount factor (0 to 1) for future rewards.

# Examples and Importance of MDPs

## Importance of MDPs

MDPs provide a formal framework for modeling decision-making in uncertain outcomes and are essential for developing algorithms in reinforcement learning.

## Example Scenario

Consider a robot navigating through a grid world:

- **States (S)**: Each cell in the grid represents a state.
- **Actions (A)**: The robot can move Up, Down, Left, or Right.
- **Transition (P)**: Probability of slipping to a different cell.
- **Rewards (R)**: Positive for reaching a goal, negative for hitting obstacles, and a small penalty for moves.
- **Discount Factor ($\gamma$)**: Values immediate rewards versus future rewards.

## Understanding MDPs Through Formulas

The **Bellman Equation** is central to solving MDPs:

$$V(s) = \max_{a \in A} \left( R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V(s') \right) \tag{1}$$

- $V(s)$: Value function representing the maximum expected reward from state $s$.
- The equation indicates that the value of a state is the maximum expected reward obtainable by choosing an action $a$ and subsequently following the optimal policy.

# Key Points to Emphasize about MDPs

- MDPs are integral to robotics, economics, and artificial intelligence.
- Understanding MDP structure aids in developing efficient algorithms for decision-making problems.
- Real-world applications include autonomous navigation, resource management, and game playing.

In the next slide, we will provide an overview of the key concepts related to MDPs, including their characteristics, value iteration, and policy iteration methods.

# What is a Markov Decision Process (MDP)?

- **Definition**: An MDP is a mathematical framework for modeling decision-making situations where outcomes are partly random and partly under the control of a decision-maker.
- **Key Components**:
  - **States (S)**: Set of all possible states. For example, in a game, this could represent the current position of a player.
  - **Actions (A)**: Set of all possible actions the agent can take, like moving left, right, or jumping.
  - **Transition Function (P)**: Probabilities of moving from one state to another given a particular action, denoted as $P(s'|s, a)$.
  - **Reward Function (R)**: Assigns numerical values (rewards) to state-action pairs, symbolized as $R(s, a)$.
  - **Discount Factor ($\gamma$)**: A value between 0 and 1 that determines the importance of future rewards.

# Key Concepts and Terminology

- **Policy ($\pi$)**: A strategy that the agent uses to decide actions based on the current state, represented as $\pi(a|s)$.
- **Value Function (V)**: Measures the goodness of a state under a specific policy:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \middle| s_0 = s, \pi\right]$$

- **Optimal Policy ($\pi^*$)**: The policy that maximizes the expected sum of rewards.

# Example Scenario and Key Points

- **Example Scenario**:
  - A grid world agent aims to reach a goal while avoiding obstacles. Each cell is a state, allowing movements of up, down, left, or right.
- **Key Points**:
  - MDPs are foundational for reinforcement learning algorithms.
  - Components of MDPs motivate solution techniques, including:
    - **Dynamic Programming**: Uses Bellman equations for value functions and optimal policies.
    - **Monte Carlo Methods**: Estimates value functions through averaging returns from sampled episodes.
    - **Temporal Difference Learning**: Combines ideas from both dynamic programming and Monte Carlo methods.
- **Conclusion**: Understanding MDPs facilitates learning about reinforcement learning algorithms and their application to real-world decisions.

# Conclusion - Summary of Key Concepts

In this chapter, we explored **Markov Decision Processes (MDPs)**, a mathematical framework for sequential decision-making under uncertainty. MDPs model environments where outcomes are partly random and partly controlled.

The key components of MDPs are:

- **States (S)**: Various configurations of the environment.
- **Actions (A)**: Choices available to the decision-maker.
- **Transition Function (P)**: Probability of moving from one state to another after taking an action.
- **Reward Function (R)**: Feedback in terms of rewards for each transition.
- **Discount Factor ($\gamma$)**: Represents the importance of future rewards.

# Mathematical Representation of MDPs

The objective in an MDP is to find a policy $\pi$ that maximizes the expected sum of rewards over time. The **Optimal Value Function ($V^*$)** is defined as:

$$V^*(s) = \max_{\pi} \sum_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a) + \gamma V^*(s')] \tag{2}$$

# Example and Applications

**Example:** Consider a grid world where an agent can move in four directions (up, down, left, right). The agent receives a reward for reaching a goal state (e.g., +10) and a penalty for falling into a trap (e.g., -10).

- The agent's goal is to find the optimal policy that maximizes its expected rewards while avoiding traps.
- **Key Points:**
  - Understanding MDPs is crucial for fields like machine learning and robotics.
  - **Value Iteration** and **Policy Iteration** are two primary algorithms used to find optimal policies.
  - MDPs are applicable in various domains such as robotics (path planning), finance (investment decisions), and healthcare (treatment planning).

# Conclusion

By mastering MDPs, you are prepared to tackle complex decision-making problems involving uncertainty.

### Next Steps

The next steps will involve implementing algorithms based on MDPs and applying them to practical scenarios for better understanding and real-world application.