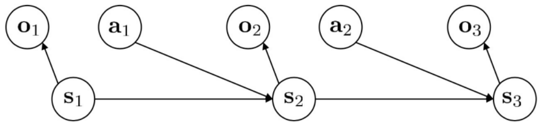
Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

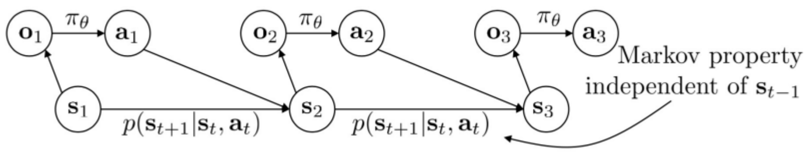
Summary of Lecture 3: Value-Based Methods

Summarized By: Amirhossein Asadi

- In the graphical model of an **MDP**, the environment is modeled as a directed graph where nodes represent states and edges represent probabilistic transitions between states influenced by actions. Each edge is associated with a transition probability and a **reward**. This representation helps in understanding the dynamic structure of the problem and analyzing optimization methods.



- A **policy** can be defined as a function of the **state** ($\pi(S_t)$) in fully observable environments (MDPs) or the observation ($\pi(O_t)$) in partially observable environments **(POMDPs)**. In the latter case, since the agent lacks full state information, it must rely on observations to make decisions.



- The **optimal value function** $V^*(s)$ represents the maximum expected return from state s under the **optimal policy** :

$$V^*(s) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t, S_{t+1}) \mid S_0 = s, \pi\right]$$

  Similarly, the **optimal action-value function** $Q^*(s, a)$ gives the maximum return when taking action a in state s.

- **Value Iteration** is an algorithm used to find the optimal value function in an **MDP**. It repeatedly updates the value of each state based on the expected rewards and the value of neighboring states, using the **Bellman optimality equation**. This process continues until the values converge, allowing the optimal policy to be derived from the final value function.

- **Policy Iteration** is an algorithm for finding the optimal policy in an **MDP**. It alternates between **policy evaluation**, where the value function for the current policy is calculated, and **policy improvement**, where the policy is updated based on the current value function. This process repeats until the policy converges to the optimal one.

- **Policy Evaluation** is the process of calculating the value function for a given **policy**. It updates the value of each state iteratively until it converges, showing the expected return for each state under the current policy.

- **Policy Improvement** is the process of updating a **policy** by selecting actions that maximize the value function for each state. The policy is improved iteratively until no further improvements can be made, leading to the optimal policy.

- **Planning vs. Learning: Planning** uses a model of the environment to simulate actions and improve decision-making, while **Learning** involves interacting with the environment to update policies based on experience. The main difference is that planning relies on a model, while learning uses real-world data.