# Reinforcement Learning: Markov Decision Processes

BIOE 498/598 PJ

Spring 2022

# Supervised learning vs. Reinforcement learning (RL)

**Supervised Learning**

- Learning from data that has already been collected
- Examples: Linear models, Gaussian Process Regression, Neural Networks

## Supervised learning vs. Reinforcement learning (RL)

**Supervised Learning**

- Learning from data that has already been collected
- Examples: Linear models, Gaussian Process Regression, Neural Networks

**Reinforcement Learning**

- Learning from trial and error
- Examples: Animals, computer chess, self-driving cars

# RL is structured randomness

- Many RL algorithms rely on random processes to generate data.
- RL needs structure to learn from these data.
- The most common framework is the *Markov Decision Process* (MDP).

# Markov Decision Processes

MDPs describe how an *agent* interacts with its environment.

- At any time, the agent and environment are described by a **state**.
- The agent selects an **action** to move between states.
- Every action and state produce a **reward**.
- The agent's goal is to maximize the total reward it collects.

# Markov Decision Processes

MDPs describe how an *agent* interacts with its environment.

- At any time, the agent and environment are described by a **state**.
- The agent selects an **action** to move between states.
- Every action and state produce a **reward**.
- The agent's goal is to maximize the total reward it collects.

MDPs have the *Markov Property*:

- All decisions depend only on the current state.
- Each state includes all of the relevant history.

## Markov Decision Processes (continued)

- We denote a state as $s$.
- The actions $a \in \mathcal{A}$ available to the agent can depend on the state, so $\mathcal{A} = \mathcal{A}(s)$.
- A **policy** $\pi$ is a function that maps states to actions. The value $\pi(s, a)$ is the probability that the agent will select action $a$ in state $s$.

# Markov Decision Processes (continued)

- We denote a state as $s$.
- The actions $a \in \mathcal{A}$ available to the agent can depend on the state, so $\mathcal{A} = \mathcal{A}(s)$.
- A **policy** $\pi$ is a function that maps states to actions. The value $\pi(s, a)$ is the probability that the agent will select action $a$ in state $s$.
- MDPs can be *deterministic* or *stochastic*.
  - Deterministic: Actions always determine the next state.
  - Stochastic: Actions change the probability that any other state will be the next state.
- We will focus on *finite horizon* or *episodic* MDPs.
  - Finite horizon MDPs stop (terminate) after a finite number of actions.
  - A *trajectory* is a single pass through a finite horizon MDP.

# Gridworld

Imagine a simple maze on a $4 \times 4$ grid.

- Each square is a state.
- The walls determine the available actions at each state.
- The agent starts in the bottom left and must reach the top right.
- The objective is to finish the maze in as few steps as possible.

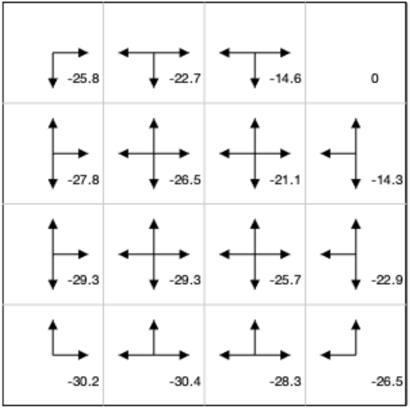
| | | | |
|---|---|---|---|
| | | | end |
| | | | |
| | | | |
| start | | | |

## A Monte Carlo approach

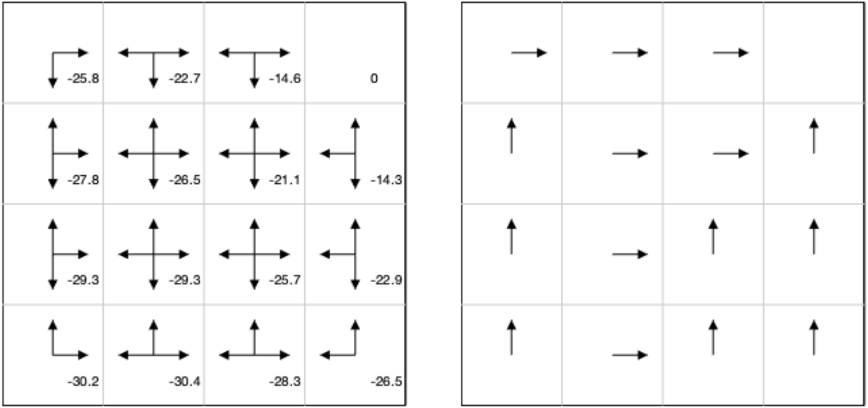
- Each grid square is a state.
- Actions: move up, down, left, or right, but the agent cannot leave the grid.
- Reward: $-1$ for each step.
- Policy: Random.

Starting from a random state, make random moves until the agent reaches the end.

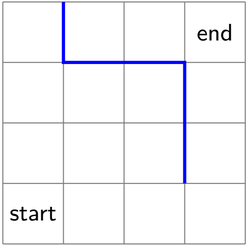
Repeat may times and average the total rewards from each trajectory.



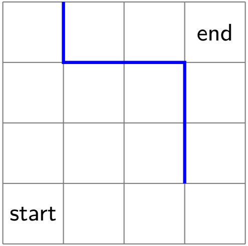
# From randomness to a better policy (policy improvement)

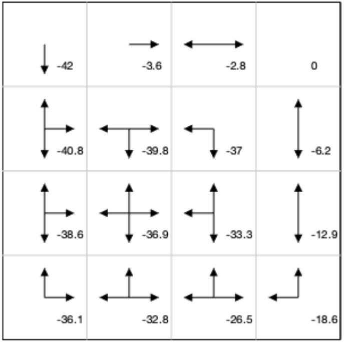


Let's add an internal wall for the agent to navigate.

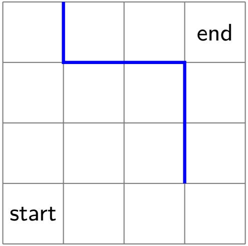


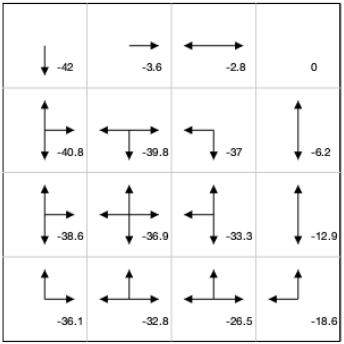
Let's add an internal wall for the agent to navigate.

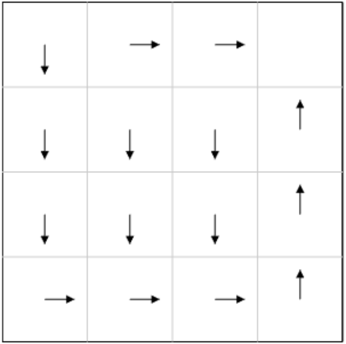


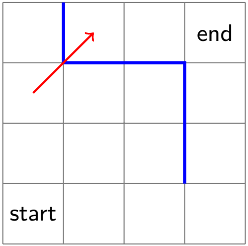
Let's add an internal wall for the agent to navigate.



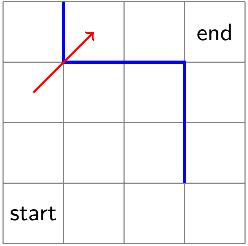



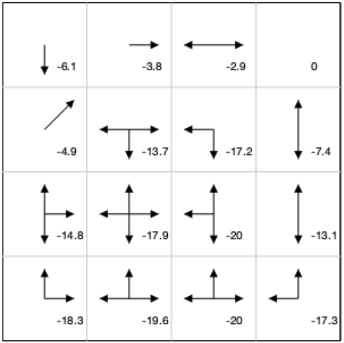



Can the agent learn a shortcut?

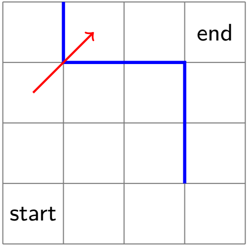


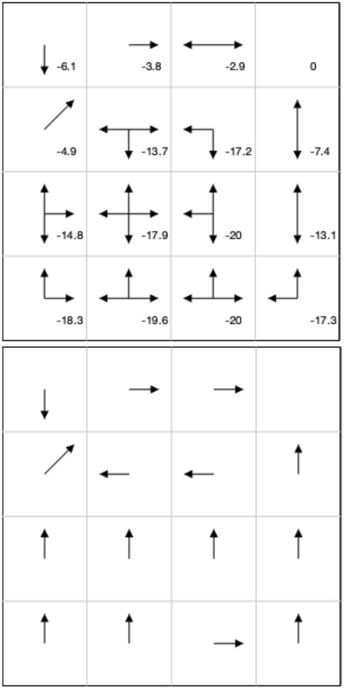
Can the agent learn a shortcut?



Can the agent learn a shortcut?



# Summary

- RL agents can learn by trial and error.
- MDPs provide a mathematical structure for RL problems.
- The choice of states, actions, and rewards is critical.

## Summary

- RL agents can learn by trial and error.
- MDPs provide a mathematical structure for RL problems.
- The choice of states, actions, and rewards is critical.
- **Next time:** What are we learning from our random maze walks?