L2: Markov Decision Processes

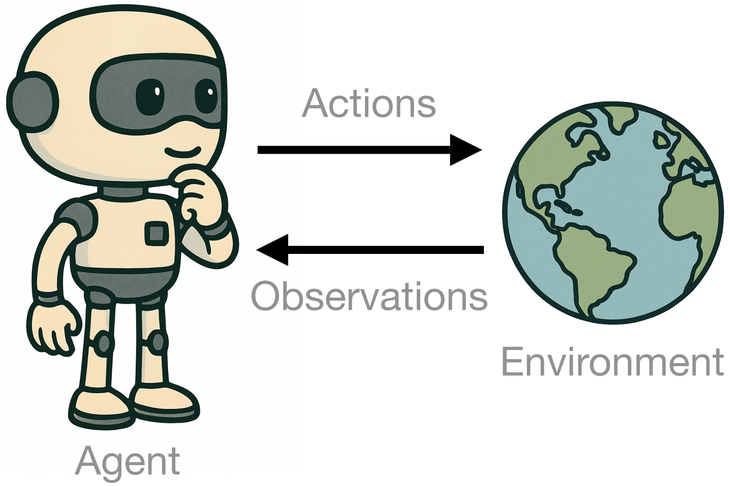
EECE 571N | Sequential Decision Making | Fall 2025

Cyrus Neary | cyrus.neary@ubc.ca

Learning Objectives

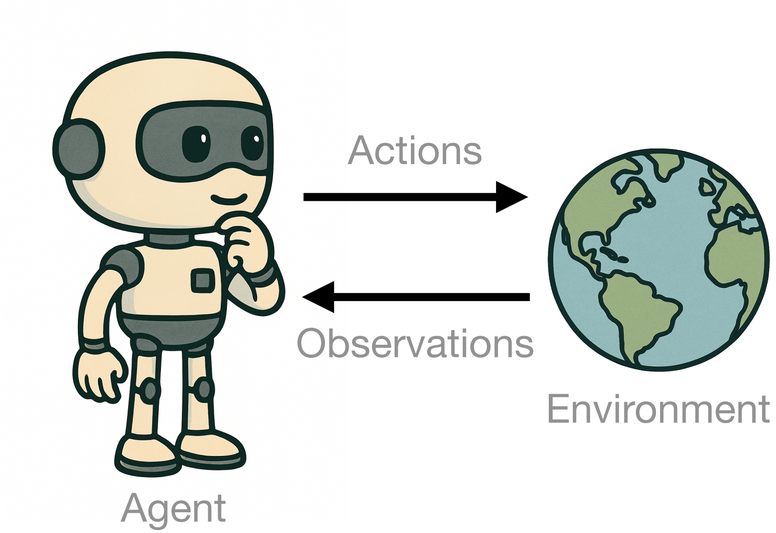
By the end of today's lecture, you should be able to…

- Explain the core modeling components of Markov decision processes (MDPs).
- Differentiate between classes of decision-making policies.
- Understand the components of the typical optimization objective in MDP problems.
- Define value functions and optimal value functions, and explain their role in sequential decision-making problems.



Sequential Decision-Making Problems

Objective: The decision-making *agent*  should implement a *policy* that selects actions a_t at every *decision epoch* t to maximize some function of the resulting rewards $r_0, r_1, \ldots$

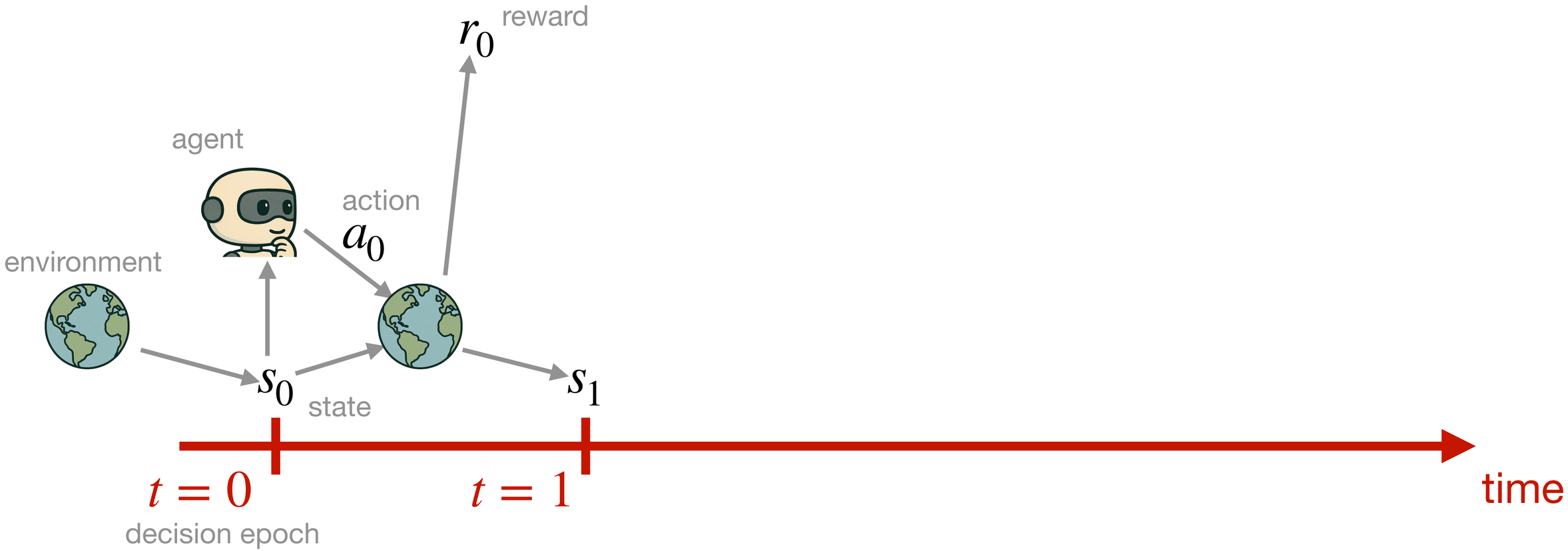


Sequential Decision-Making Problems

Objective: The decision-making *agent*  should implement a *policy* that selects actions a_t at every *decision epoch* t to maximize some function of the resulting rewards $r_0, r_1, \ldots$

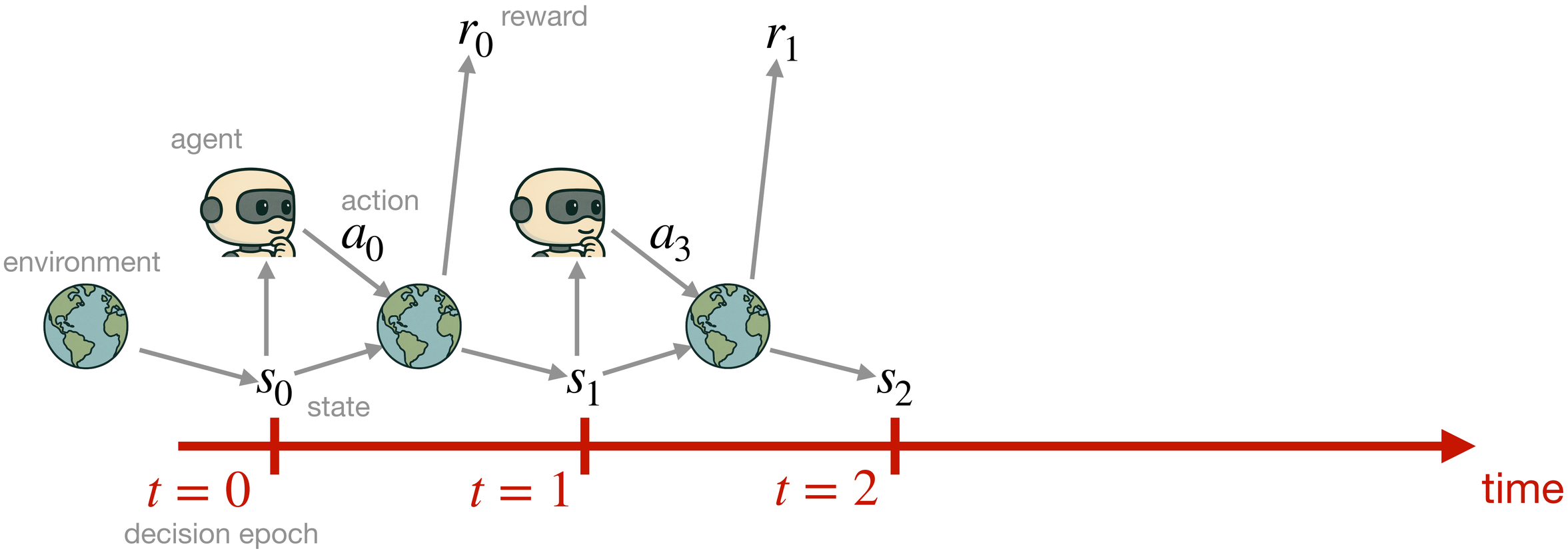
Sequential Decision-Making Problems

Objective: The decision-making *agent* should implement a *policy* that selects actions a_t at every *decision epoch* t to maximize some function of the resulting rewards $r_0, r_1, \ldots$



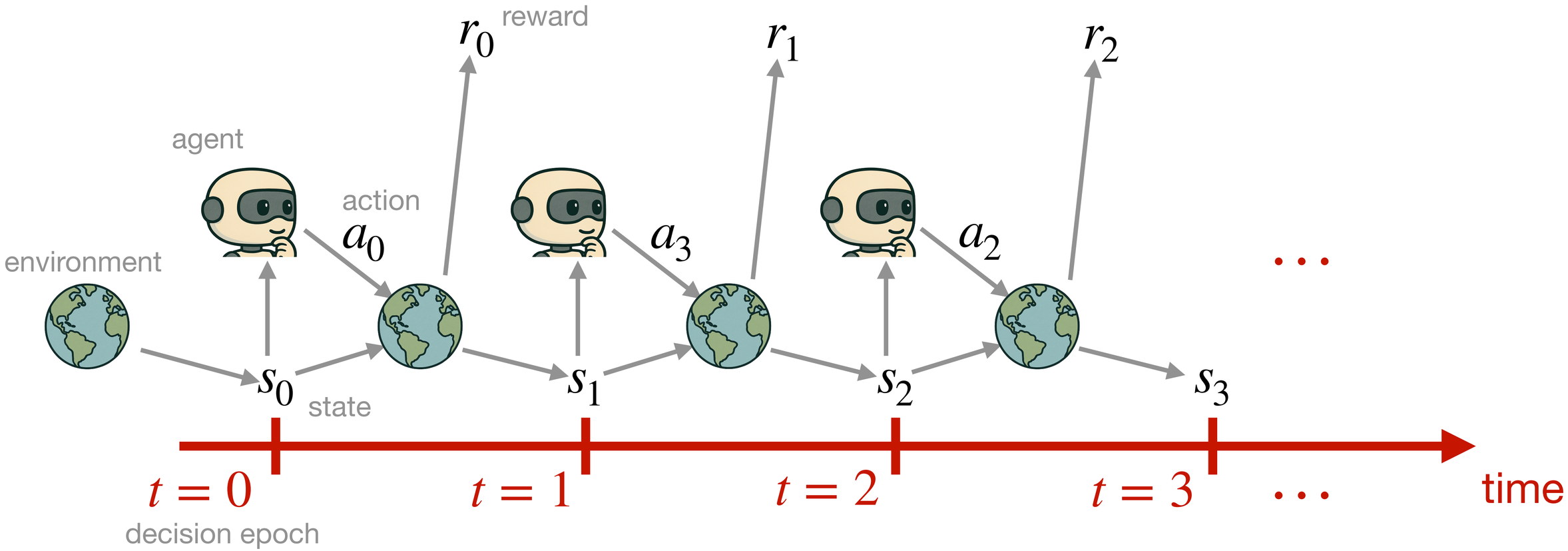
Sequential Decision-Making Problems

Objective: The decision-making *agent* should implement a *policy* that selects actions a_t at every *decision epoch* t to maximize some function of the resulting rewards $r_0, r_1, \ldots$

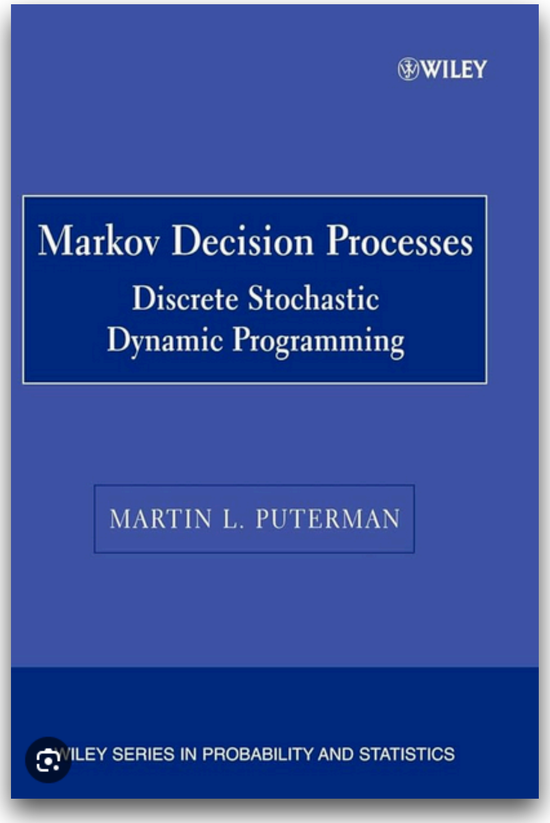


Sequential Decision-Making Problems

Objective: The decision-making *agent* should implement a *policy* that selects actions a_t at every *decision epoch* t to maximize some function of the resulting rewards $r_0, r_1, \ldots$



Modeling Sequential Decision-Making Problems



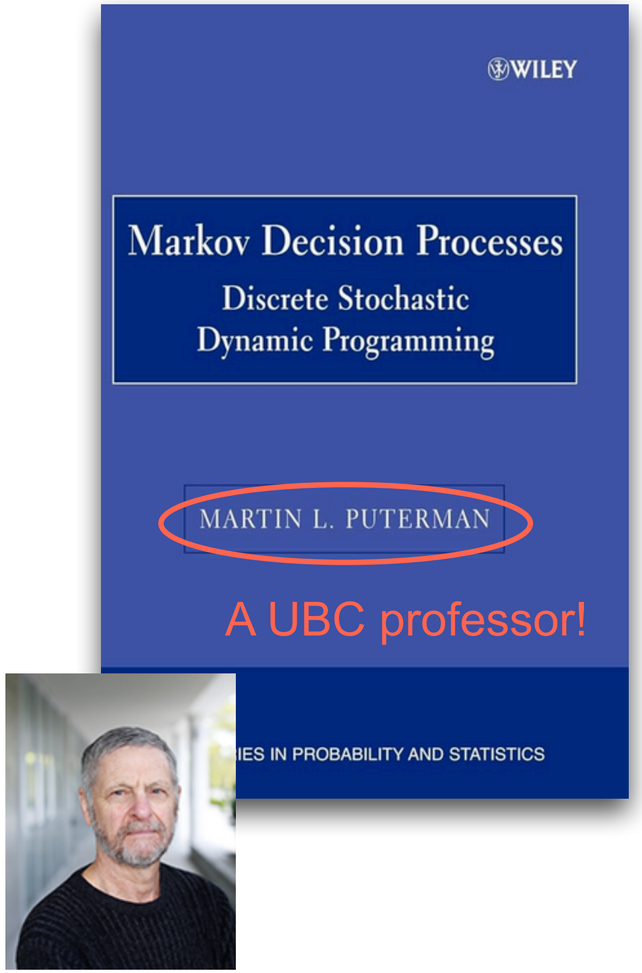
"Each day people make many decisions; decisions which have both immediate and long-term consequences. Decisions must not be made in isolation; today's decision impacts on tomorrow's and tomorrow's on the next day's…

This book presents and studies a model for sequential decision making under uncertainty, which takes into account both the outcomes of current decisions and future decision making opportunities. While this model may appear quite simple, it encompasses a wide range of applications and has generated a rich mathematical theory."

Puterman ML. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons; 2014 Aug 28.

Today's lecture draws from chapters 2, 3, and 5

Modeling Sequential Decision-Making Problems



"Each day people make many decisions; decisions which have both immediate and long-term consequences. Decisions must not be made in isolation; today's decision impacts on tomorrow's and tomorrow's on the next day's…

This book presents and studies a model for sequential decision making under uncertainty, which takes into account both the outcomes of current decisions and future decision making opportunities. While this model may appear quite simple, it encompasses a wide range of applications and has generated a rich mathematical theory."

Puterman ML. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons; 2014 Aug 28.

Today's lecture draws from chapters 2, 3, and 5

Markov Decision Processes

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

Markov Decision Processes

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

S - States

Set of all possible states from which the agent will make decisions.

Markov Decision Processes

A - Actions

Set of all possible actions the agent could take.

S - States

Set of all possible states from which the agent will make decisions.

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

Markov Decision Processes

R- Reward function

$R(s, a) \in \mathbb{R}$ — A function defining the reward the agent receives for transitioning from state $s \in S$ to $s' \in S$ under action $a \in A$.

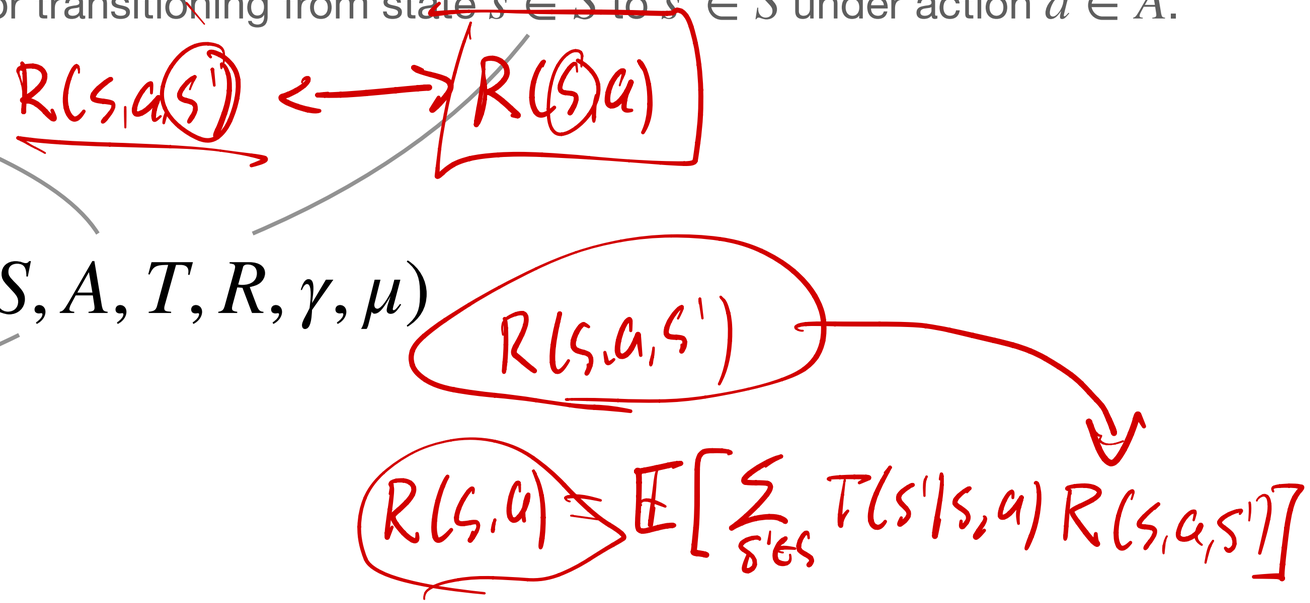
A - Actions

Set of all possible actions the agent could take.

S - States

Set of all possible states from which the agent will make decisions.

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$



Markov Decision Processes

R- Reward function

$R(s, a) \in \mathbb{R}$ — A function defining the reward the agent receives for transitioning from state $s \in S$ to $s' \in S$ under action $a \in A$.

A - Actions

Set of all possible actions the agent could take.

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

μ - Initial state distribution

$\mu(s)$ denotes the probability that the initial state is s.

S - States

Set of all possible states from which the agent will make decisions.

Markov Decision Processes

R- Reward function

$R(s, a) \in \mathbb{R}$ — A function defining the reward the agent receives for transitioning from state $s \in S$ to $s' \in S$ under action $a \in A$.

A - Actions

Set of all possible actions the agent could take.

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

μ - Initial state distribution

$\mu(s)$ denotes the probability that the initial state is s.

S - States

Set of all possible states from which the agent will make decisions.

γ - Discount factor

$\gamma \in [0,1]$ — a modeling parameter that reduces the impact of future rewards on decisions made now.

Markov Decision Processes

R- Reward function

$R(s, a) \in \mathbb{R}$ — A function defining the reward the agent receives for transitioning from state $s \in S$ to $s' \in S$ under action $a \in A$.

A - Actions

Set of all possible actions the agent could take.

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

μ - Initial state distribution

$\mu(s)$ denotes the probability that the initial state is s.

S - States

Set of all possible states from which the agent will make decisions.

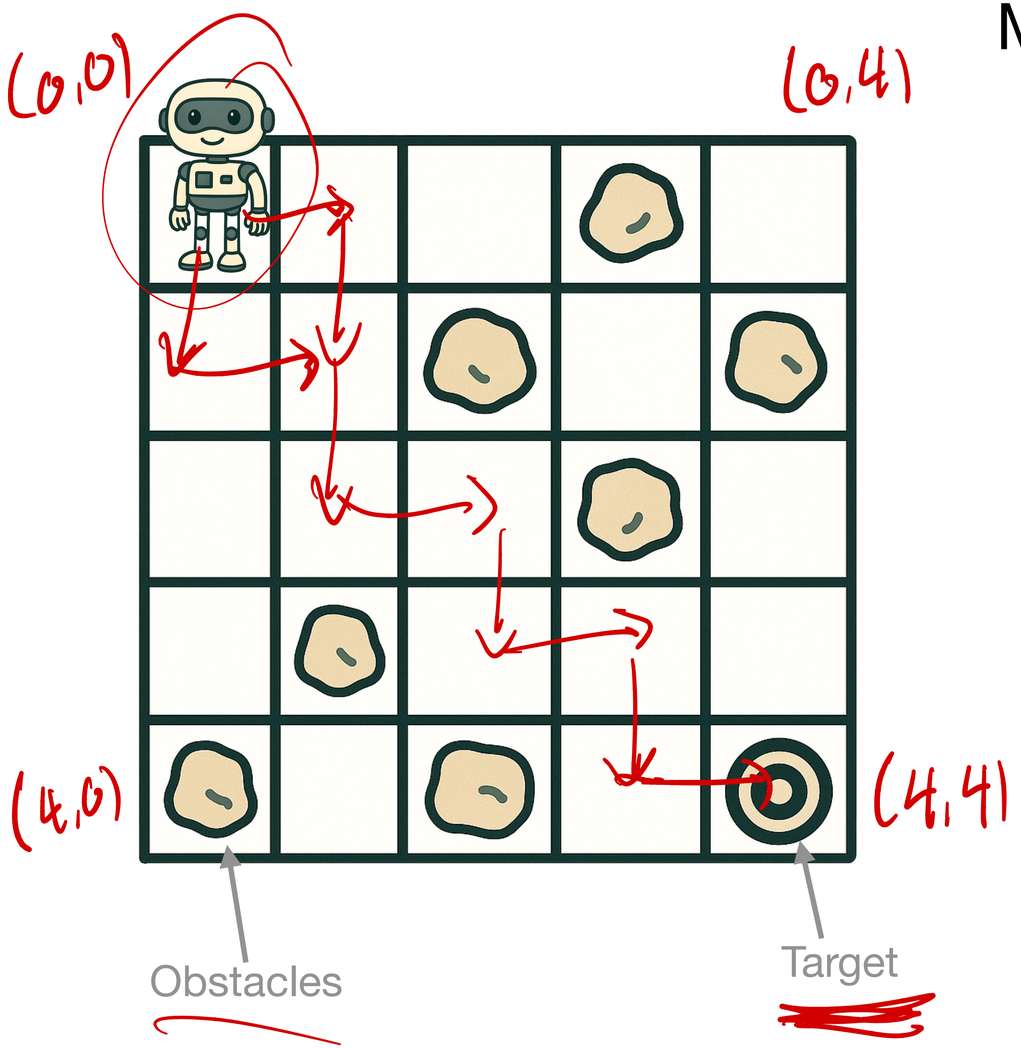
γ - Discount factor

$\gamma \in [0,1]$ — a modeling parameter that reduces the impact of future rewards on decisions made now.

T - Transition function

$s' \sim T(\cdot \mid s, a)$ a stochastic transition function defining the probability of the agent transitioning to next state $s' \in S$, given that it executed action $a \in A$ from state $s \in S$.

MDP Examples: Gridworld Navigation



MDP Components:

State space S — All possible locations of robot

$S = \{(x, y) \mid x, y \in \{0, 1, 2, 3, 4\}\}$

Action space A — Movement actions. Don't move

$A = \{\text{left}, \text{right}, \text{up}, \text{down}, \text{stay}\}$

Transition function T — Move as planned, but with prob p, "slip" to a random adjacent square.

$$T(s' \mid s, a) = \begin{cases} \text{Move in intended direction} & \text{w.p. } (1-p) \\ \text{Move in random direction} & \text{w.p. } p \end{cases}$$

agent takes $a = \text{"up"} \in A$

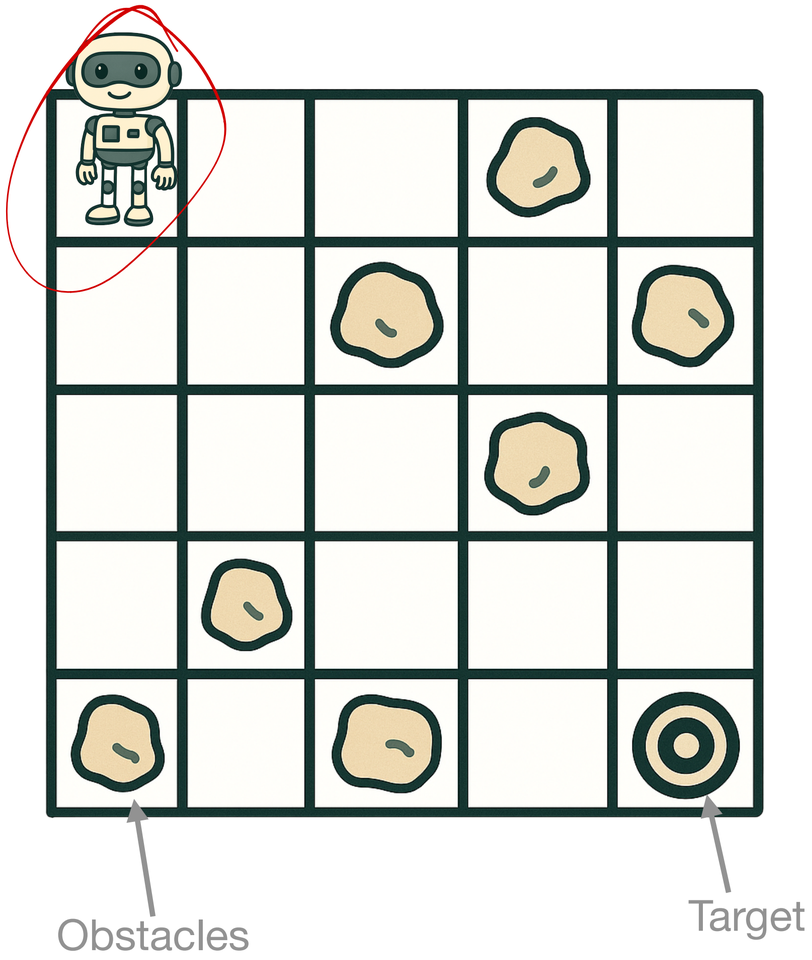
$s = (2,2)$

$$T(s'|s,a) = \begin{cases} (2,1) & \text{with prob } (1-p) \\ \left.\begin{array}{ll} (2,2) & \text{w.p. } p/3 \\ (1,2) & \text{w.p. } p/3 \\ (3,2) & \text{w.p. } p/3 \end{array}\right\} & \text{w.p. } p \\ \sout{(2,3)} \quad \sout{(2,1)} & \sout{\text{w.p. } 0} \end{cases}$$

what happens when taking an "impossible" action?

it's up to you!

MDP Examples: Gridworld Navigation



MDP Components:

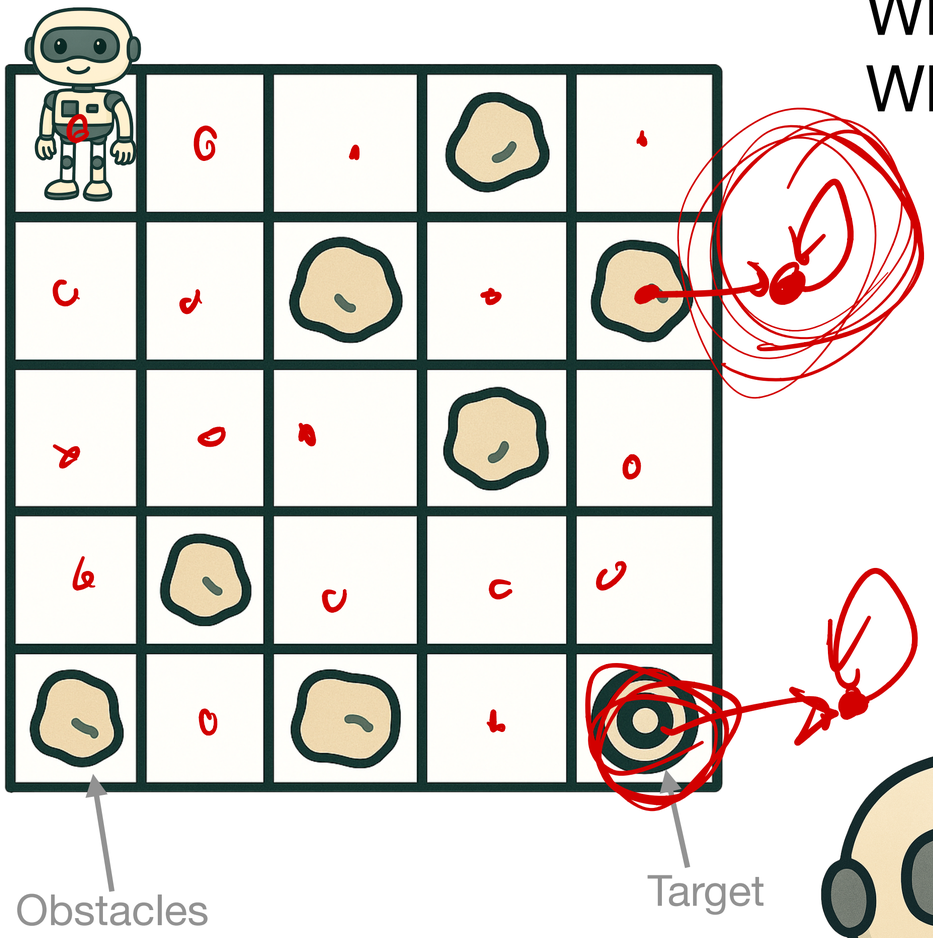
Reward function R

$$R(s,a) = \begin{cases} +1 & \text{if } s=(4,4) \\ -1 & \text{if } s \in \{(3,0),(2,1),(2,4),\} \\ -0.1 & \text{otherwise} \end{cases}$$

Initial distribution $\mu(s)$

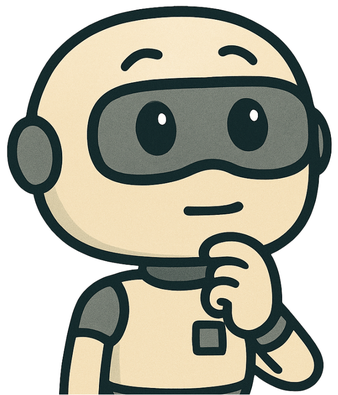
$$\mu(s) = \begin{cases} 1 & \text{if } s=(0,0) \\ 0 & \text{otherwise} \end{cases}$$

MDP Examples: Gridworld Navigation



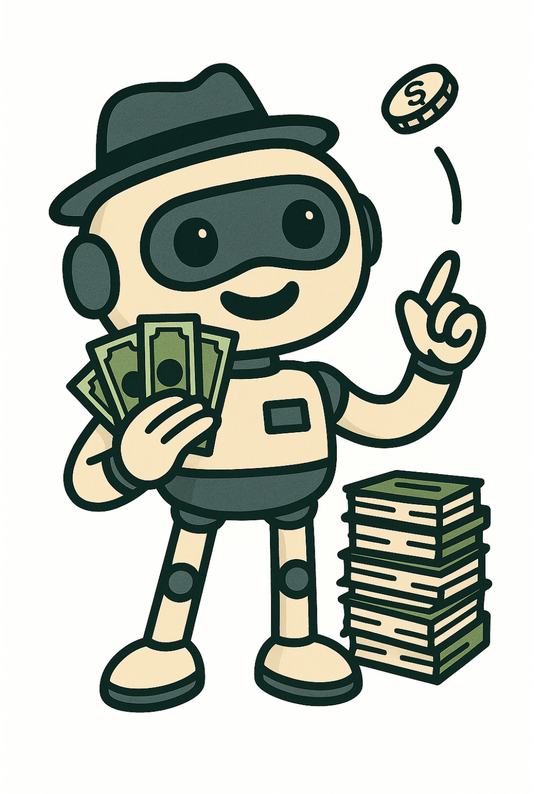
Why include stochastic transitions?
What could they model?

- Real world is imperfect.
  - ↳ our model is imperfect
  - ↳ Stochasticity introduces conservative behaviors.
- Imperfect observations.



MDP Examples: Gambler's Ruin

- A gambler plays a game in which they make a bet of B dollars at every turn.
- They then have a probability p of winning $2B$ dollars, and a probability $1-p$ of losing the B dollars that they bet.
- They start with 50 dollars, and their goal is to cash out once they have have reached 100 dollars.



MDP Components:

State space S

All possible amounts of money that the gambler has.

$S = \{0, 1, 2, \ldots, 100\}$

Action space A

Any bet amount below current money.

$A(s) = \{0, 1, 2, \ldots, s\}$

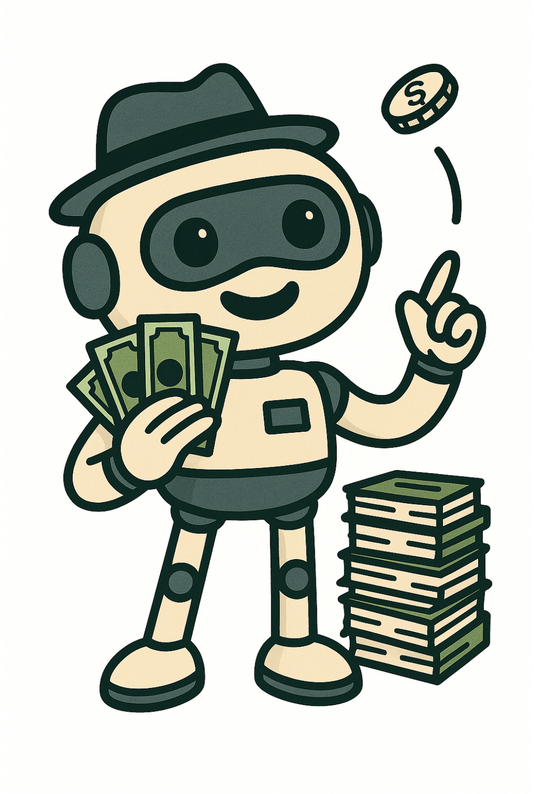
Transition function T

Double the bet w.p. p and loose it with probability $1-p$.

$$T(s'|s,a) = \begin{cases} p & \text{if } s' = 2a+s \\ 1-p & \text{if } s' = s-a \\ 0 & \text{otherwise} \end{cases}$$

MDP Examples: Gambler's Ruin

- A gambler plays a game in which they make a bet of B dollars at every turn.
- They then have a probability p of winning $2B$ dollars, and a probability $1-p$ of losing the B dollars that they bet.
- They start with 50 dollars, and their goal is to cash out once they have have reached 100 dollars.



MDP Components:

Reward function R

Since we want it to cash out at $s=100$,

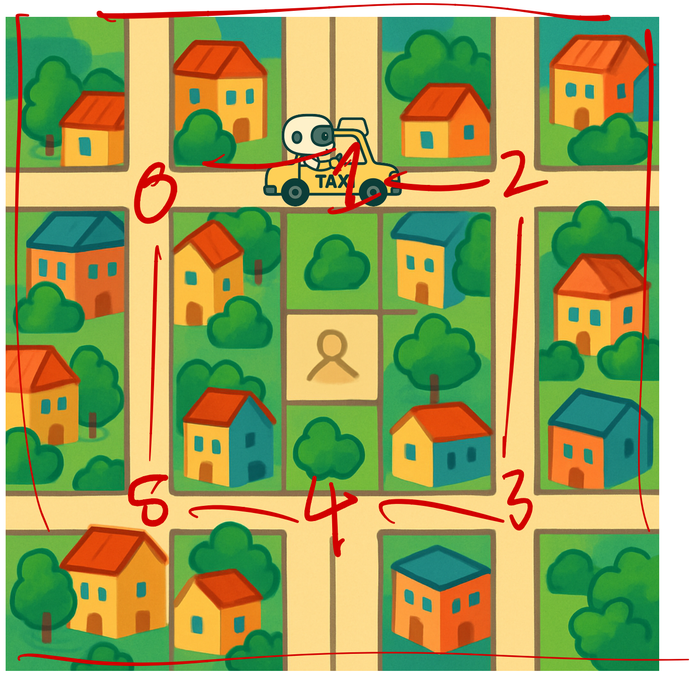
$$R(s,a) = \begin{cases} +1 & \text{if } s=100 \\ 0 & \text{otherwise.} \end{cases}$$

Initial distribution $\mu(s)$

$$\mu(s) = \begin{cases} 1 & \text{if } s=50 \\ 0 & \text{otherwise} \end{cases}$$

MDP Examples: Autonomous Taxi Example

- An autonomous taxi needs to pickup and dropoff passengers in a city.
- Passengers randomly spawn at locations throughout the city, and each has a different dropoff location.
- The taxi can only carry at most 1 passenger at a time.
- The taxi gets $5 each time they deliver a passenger to the dropoff point.



MDP Components:

State space S

$S_{pass_i}^{dest} = \{0,1,2,3,4,5\}$ for all $i \in \{1,2,\ldots,10\}$

All relevant locations for the taxi.

$S_{loc} = \{0,1,2,3,4,5\}$

Locations of passengers waiting for pickup.

$S_{pass} = \{0,1,2,3,4,5\}$ for passenger $i \in \{1,2,\ldots,10\}$

Is the taxi full?

$S_{bool} = \{T, 1\}$

Action space A

- Pickup/dropoff passenger
- Move between adjacent locations.

Transition function T

$T(s'|s,a) \quad s \in S$

a move between locations $\rightarrow$ $T(s'|s,a) = 1$ if s' contains that new location, 0 otherwise

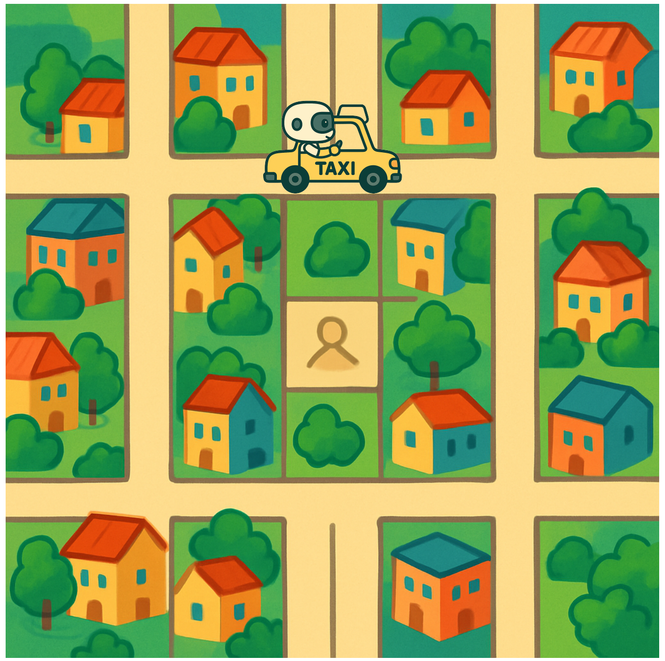
a pickup or dropoff. $\rightarrow$ Set the passenger location, and change Bool portion of state.

State space

$$s \in S = S_{loc} \times S_{bool} \times S_{pass,1} \times \cdots \times S_{pass,16} \times S^{dest}_{pass,1} \times \cdots \times S^{dest}_{pass,10}$$

MDP Examples: Autonomous Taxi Example

- An autonomous taxi needs to pickup and dropoff passengers in a city.
- Passengers randomly spawn at locations throughout the city, and each has a different dropoff location.
- The taxi can only carry at most 1 passenger at a time.
- The taxi gets $\$5$ each time they deliver a passenger to the dropoff point.



MDP Components:

Reward function R

$$R(s,a) = \begin{cases} +5 & \text{if successful dropoff} \\ 0 & \text{otherwise} \end{cases}$$

a = dropoff action

s = location of taxi
is the passenger's destination

Initial distribution $\mu(s)$

uniform distribution for where the taxi starts, where the passengers start etc.

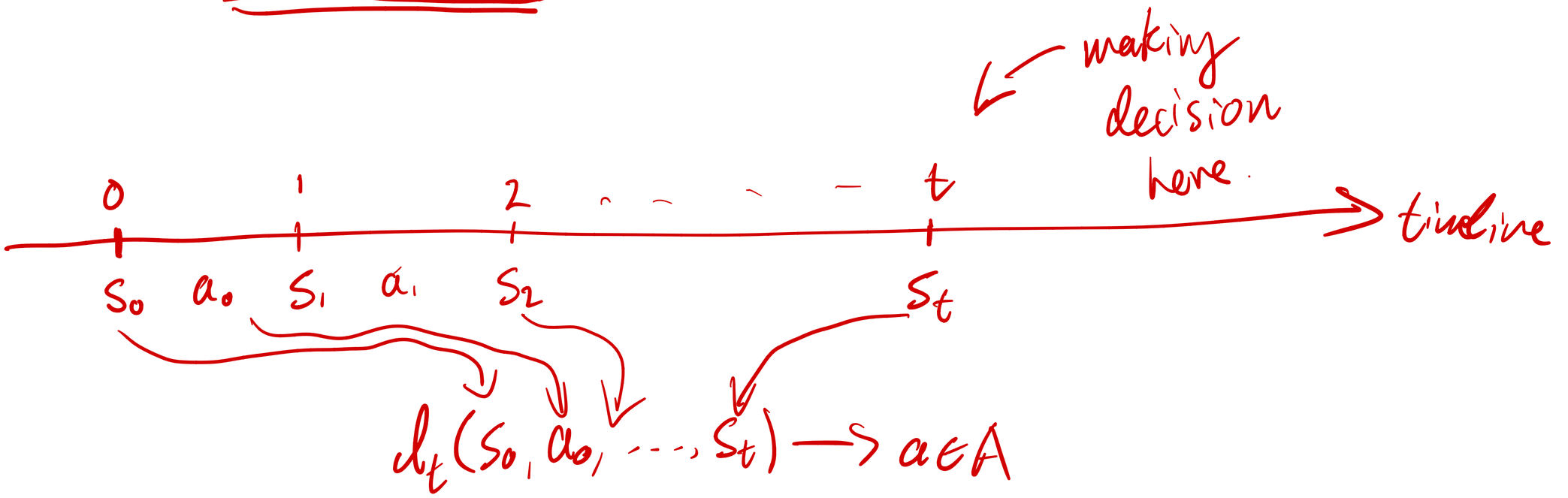
Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

A decision rule is *history-dependent* if it relies on previous states and actions: $d_t(s_0, a_0, \ldots, s_t)$

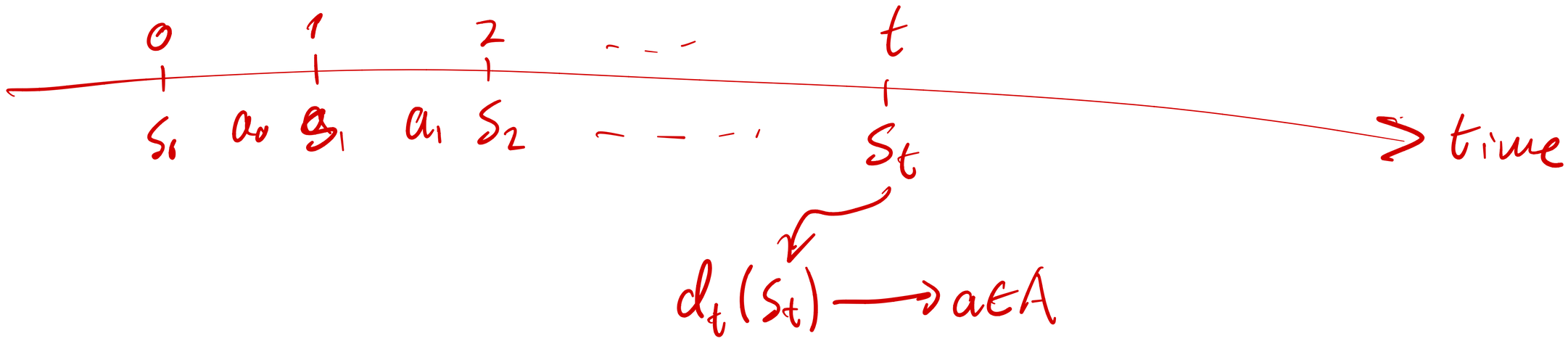


Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

A decision rule is *history-dependent* if it relies on previous states and actions: $d_t(s_0, a_0, \ldots, s_t)$

A decision rule is *Markovian* if it only relies on the current state: $d_t(s_t)$



Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

A decision rule is *history-dependent* if it relies on previous states and actions: $d_t(s_0, a_0, \ldots, s_t)$

A decision rule is *Markovian* if it only relies on the current state: $d_t(s_t)$

A decision rule is *deterministic* if it outputs a single action $d_t(s_t) = a \in A$

Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

A decision rule is *history-dependent* if it relies on previous states and actions: $d_t(s_0, a_0, \ldots, s_t)$

A decision rule is *Markovian* if it only relies on the current state: $d_t(s_t)$

A decision rule is *deterministic* if it outputs a single action $d_t(s_t) = a \in A$

A decision rule is *random* if it outputs a distribution over actions $d_t(a \mid s) = \mathbb{P}(a_t = a \,|\, s_t = s)$

Decision Rules

A *decision rule* d_t is a function specifying an action to be executed at an individual decision epoch t.

A decision rule is *history-dependent* if it relies on previous states and actions: $d_t(s_0, a_0, \ldots, s_t)$

A decision rule is *Markovian* if it only relies on the current state: $d_t(s_t)$

A decision rule is *deterministic* if it outputs a single action $d_t(s_t) = a \in A$

A decision rule is *random* if it outputs a distribution over actions $d_t(a \mid s) = \mathbb{P}(a_t = a \,|\, s_t = s)$

Decision rules can be:

	History-Dependent	**Markovian**
Deterministic	HD	MD
Random	HR	MR

When would we expect a decision rule to need history in order to make good decisions?

→ data about env. stochasticity.

Learning ← History is almost like "data" that we can use to build a model of what will happen under our next action.

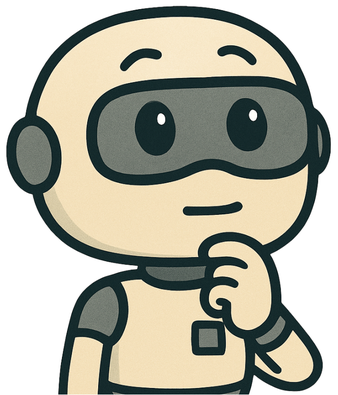
- Current observation is insufficient to fully encode the information needed to make good decisions.

Learning → - History to build a model of "what we know" and needs to be explored.

- when tasks have time limits or countdowns.

↳ "Finite horizon" decision problem.

$t \in [0, 1, 2, \ldots, T]$



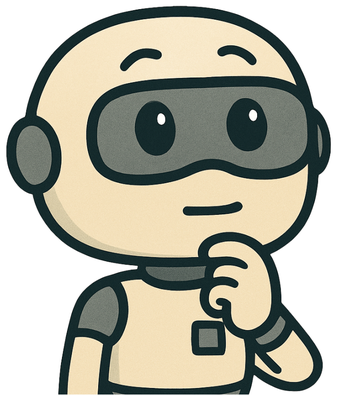
When might a random decision rule be helpful over a deterministic one?

- Adversarial setting.
  - ↳ Make the moves unpredictable.

Learning: Randomly try moves to "explore" and gather more data
or
  - ↳ Breaking out of local optima

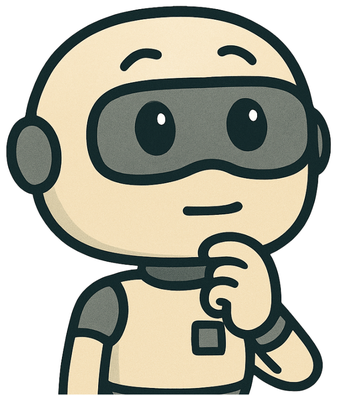
- Rewards stochastic

- there can be multiple optimal policies.
  - ↳ we can randomize between them and the result is still optimal.



More generally, why distinguish between different types of decision rules?

we need balance between the computational tractability and our ability to model what's going on in the problem.



Policies and Policy Classes

A *policy* $\pi = d_0, d_1, \ldots$ is a sequence of decision rules that specifies which action the agent should take at **every** decision epoch.

A policy belongs to a *policy class* $\Pi^{HD}, \Pi^{MD}, \Pi^{HR}, \Pi^{MR}$, if $\forall t,\ d_t \in D^{HD}, D^{MD}, D^{HR}, D^{MR}$, respectively.

e.g. $d_0 \in D^{MD}$, $d_1 \in D^{MD}$, ..., $d_t \in D^{MD}$

A policy is *stationary* if the decision rule is constant in time, i.e. $d_t = d_{t'}$ for every t, t'.

In this course, we will often consider *stationary Markovian policies.*

$\pi \in \Pi^{SM}$

What's the most general class of policies?

Stationary deterministic

History-dependent randomized

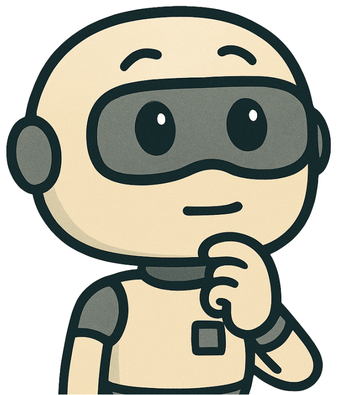
$$\Pi^{SD} \subset \Pi^{SR} \subset \Pi^{MR} \subset \Pi^{HR}$$

$$\Pi^{SD} \subset \Pi^{MD} \subset \Pi^{MR} \subset \Pi^{HR}$$

$$\Pi^{SD} \subset \Pi^{MD} \subset \Pi^{HD} \subset \Pi^{HR}$$

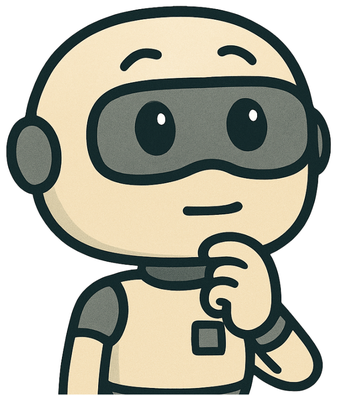
Thus Π^{HR} are the most general policies, and Π^{SD}

See Puterman's book, chapter 2.1.5 on policies.



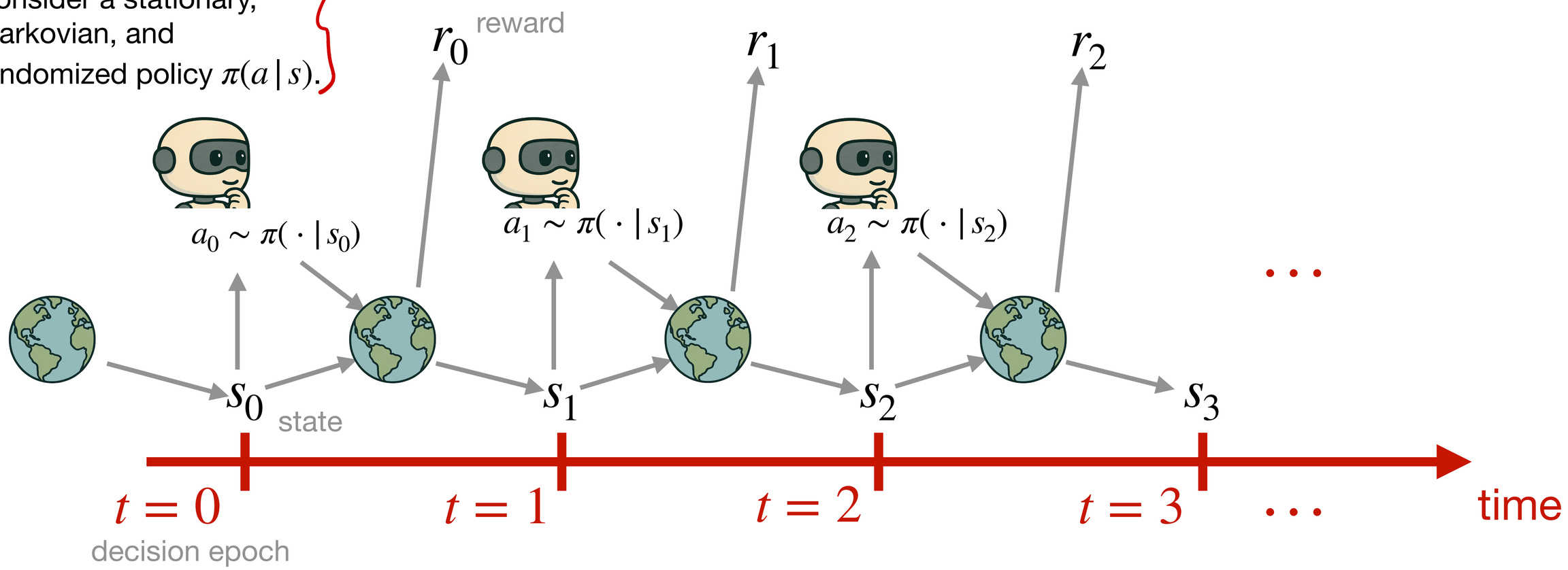
Why should we distinguish between stationary and non-stationary policies?

- Non-stationary policies are sometimes necessary for problems that involve finite horizons, or MDPs that change with time.
- Tractability, and availability of good algorithms.



Executing a Policy in an MDP

Consider a stationary, Markovian, and randomized policy $\pi(a \mid s)$.



Executing a policy induces stochastic "trajectories" $\tau = s_0, a_0, r_0, s_1, a_1, r_1, \ldots$

Objective: choose $\pi(a \mid s)$ to maximize expectation of some function of trajectories.

$$\pi^* \in \text{argmax}_\pi \mathbb{E}[f(s_0, a_0, r_0, s_1, a_1, r_1, \ldots) \mid \pi]$$

Objective: Design “Performant” Policies w.r.t. Reward

A common optimization problem: Infinite-Horizon Total Discounted Reward

$$\pi^* \in \arg\max_{\pi \in \Pi^{HR}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, \pi, s_0 \sim \mu\right]$$

Intuition: Find a policy that maximizes the expected sum of discounted rewards.

Objective: Design “Performant” Policies w.r.t. Reward

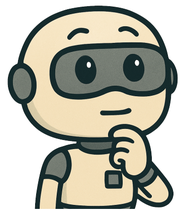
A common optimization problem: Infinite-Horizon Total Discounted Reward

$$\pi^* \in \arg\max_{\pi \in \Pi^{HR}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, \pi, s_0 \sim \mu\right]$$

Intuition: Find a policy that maximizes the expected sum of discounted rewards.

What does this optimization problem actually mean?

- What distribution is the expectation taken over?
- Why is discounting necessary?
- When does such an optimal policy exist (particularly in a manageable policy class)?



Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^{\pi})$ for **finite time horizons**?

Probability Space $(\Omega, \mathcal{F}, \mathbb{P})$

Ω sample space, set of all possible outcomes $\omega \in \Omega$

e.g. Dice $\omega \in \{1,2,3,4,5,6\} = \Omega$

$\mathcal{F} \subseteq 2^{\Omega}$: σ-algebra of events

event $A \in \mathcal{F}$ is a set of outcomes.

- $A = \{2,4,6\}$ = role # was even
- $A = \{1\}$ = rolled 1

$\mathbb{P} : \mathcal{F} \to [0,1]$ probability measure.

$$\mathbb{P}(A = \{2,4,6\}) = 3/6 = 1/2$$

$X : \Omega \to \mathbb{R}$ is a function that lets us assign values to the otherwise abstract outcomes.

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **finite time horizons**?

Sample space $\Omega = S \times A \times S \times A \times \ldots \times S = \{S \times A\}^N \times S$

$$\omega = (s_0, a_0, s_1, a_1, \ldots, s_N) \in \Omega$$

Each outcome ω is a potential *trajectory* of length N.

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **finite time horizons**?

Sample space $\Omega = S \times A \times S \times A \times \ldots \times S = \{S \times A\}^N \times S$

$$\omega = (s_0, a_0, s_1, a_1, \ldots, s_N) \in \Omega$$

Each outcome ω is a potential *trajectory* of length N.

σ-algebra $B(\Omega) = B(\{S \times A\}^{N-1} \times S)$

Events are sets of trajectories.

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **finite time horizons**?

Sample space $\Omega = S \times A \times S \times A \times \ldots \times S = \{S \times A\}^N \times S$

$$\omega = (s_0, a_0, s_1, a_1, \ldots, s_N) \in \Omega$$

Each outcome ω is a potential *trajectory* of length N.

σ-algebra $B(\Omega) = B(\{S \times A\}^{N-1} \times S)$

Events are sets of trajectories.

Probability measure

$$\begin{aligned}\mathbb{P}^\pi\left((s_0, a_0, s_1, a_1, \ldots, s_N)\right) &= \mu(s_0)\pi(a_0 \mid s_0)T(s_1 \mid s_0, a_0)\pi(a_1 \mid s_1)T(s_2 \mid s_1, a_1)\ldots \\ &= \mu(s_0)\prod_{t=0}^{N}\pi(a_t \mid s_t)T(s_{t+1} \mid s_t, a_t)\end{aligned}$$

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **infinite time horions**?

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **infinite time horions**?

Sample space $\Omega = S \times A \times S \times A \times \ldots$

$$\omega = (s_0, a_0, s_1, a_1, \ldots) \in \Omega$$

Each outcome ω is a potential infinite *trajectory.*

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **infinite time horions**?

Sample space $\Omega = S \times A \times S \times A \times \dots$

$$\omega = (s_0, a_0, s_1, a_1, \dots) \in \Omega$$

Each outcome ω is a potential infinite *trajectory.*

σ-algebra $B(\Omega) = B(\{S \times A\}^\infty)$

Events are so-called cylinder sets.

Fixing a Policy in an MDP Induces Stochastic Processes

What's the probability space $(\Omega, B(\Omega), \mathbb{P}^\pi)$ for **infinite time horions**?

Sample space $\Omega = S \times A \times S \times A \times \dots$

$$\omega = (s_0, a_0, s_1, a_1, \dots) \in \Omega$$

Each outcome ω is a potential infinite *trajectory.*

σ-algebra $B(\Omega) = B(\{S \times A\}^\infty)$

Events are so-called cylinder sets.

Probability measure

Cylinder sets have probability mass. Infinite trajectories have measure zero.

$$\mathbb{P}^\pi\left((s_0, a_0, s_1, a_1, \dots, s_N)\right) = \mu(s_0)\pi(a_0 \mid s_0)T(s_1 \mid s_0, a_0)\pi(a_1 \mid s_1)T(s_2 \mid s_1, a_1)\dots$$

$$= \mu(s_0)\prod_{t=0}^{N} \pi(a_t \mid s_t)T(s_{t+1} \mid s_t, a_t)$$

Optimality Criterion Revisited

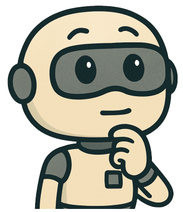
A common optimization problem: Infinite-Horizon Total Discounted Reward

$$\pi^* \in \arg\max_{\pi \in \Pi^{HR}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, \pi, s_0 \sim \mu\right]$$

Intuition: Find a policy that maximizes the expected sum of discounted rewards.

What does this optimization problem actually mean?

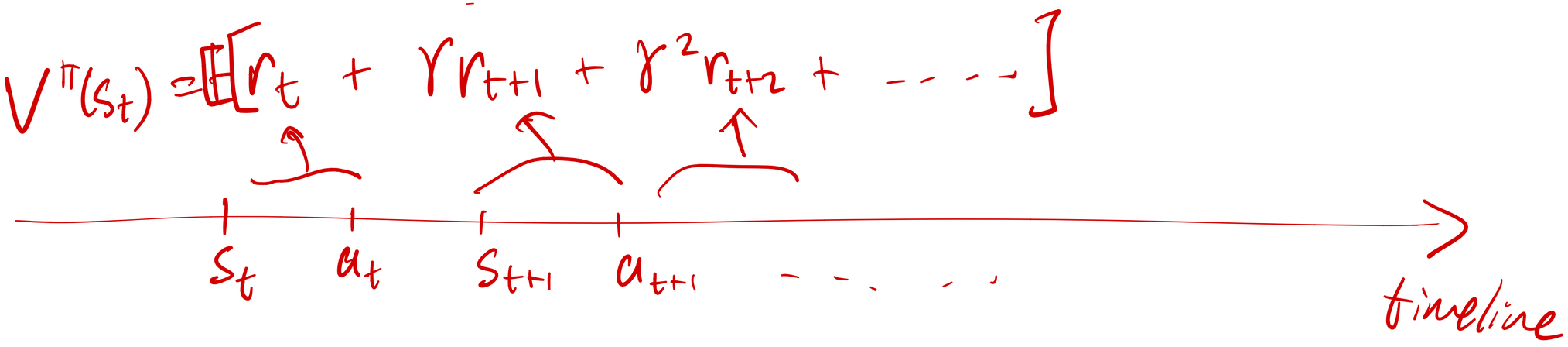
- What distribution is the expectation taken over?
- Why is discounting necessary?
- When does such an optimal policy exist (particularly in a manageable policy class)?



Defining the Value Function

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

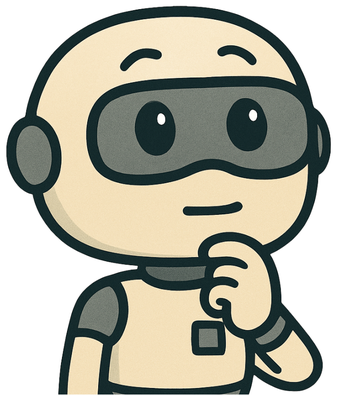
Intuition: How “valuable” is it for the agent to be in state s, if it follows policy π from that point onwards?



What Assumptions Did we Make When Defining the Value Function?

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

- We assuming this infinite sum converges.
- Assumed $V^{\pi}(s)$ doesn't depend on current time.
  - ↳ This only works for stationary policies and infinite time horizons.



The Mathematical Importance of Discounting

$$V^{\pi}(s) = \lim_{N \to \infty} \mathbb{E}\left[\sum_{t=0}^{N} \gamma^t r_t \mid \pi, s_0 = s\right]$$

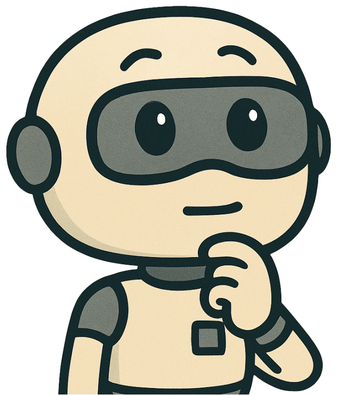
When does the limit exist?

Suppose $R(s,a)$ is bounded, $|R(s,a)| \leq M < \infty$ ~~$\forall s,a$~~

Consider $\lim_{N\to\infty} \mathbb{E}\left[\sum_{t=0}^{N} |\gamma^t r_t| \mid \pi, s_0 = s\right] \leq \lim_{N\to\infty} \sum_{t=0}^{\infty} \gamma^t M$

as long as $0 \leq \gamma < 1$, we have a geometric series converges to $\frac{M}{1-\gamma}$

Series is absolutely convergent $\Rightarrow$ convergent.



Why is this important?

- Essential for the convergence/math of infinite horizon problems.

Defining the Action-Value Function

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

Intuition: How “valuable” is it for the agent to be in state s, if it follows policy π from that point onwards?

Defining the Action-Value Function

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

Intuition: How “valuable” is it for the agent to be in state s, if it follows policy π from that point onwards?

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right]$$

Intuition: How “valuable” is it for the agent to take action a from state s, if it follows policy π from that point onwards?

Defining the Action-Value Function

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

Intuition: How “valuable” is it for the agent to be in state s, if it follows policy π from that point onwards?

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) Q^{\pi}(s, a) \quad \updownarrow \quad Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^{\pi}(s')$$

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s, a_0 = a\right]$$

Intuition: How “valuable” is it for the agent to take action a from state s, if it follows policy π from that point onwards?

An Optimality Criterion: Infinite-Horizon Total Discounted Reward

A policy $\pi^*(\cdot \mid s)$ is considered optimal whenever

$$V^{\pi^*}(s) \geq V^{\pi}(s), \quad \forall s \in S, \forall \pi \in \Pi^{HR}$$

An Optimality Criterion: Infinite-Horizon Total Discounted Reward

A policy $\pi^*(\cdot \mid s)$ is considered optimal whenever

$$V^{\pi^*}(s) \geq V^{\pi}(s), \quad \forall s \in S, \forall \pi \in \Pi^{HR}$$

The optimal value of an MDP is defined as:

$$V^*(s) := \sup_{\pi \in \Pi^{HR}} V^{\pi}(s)$$

An Optimality Criterion: Infinite-Horizon Total Discounted Reward

A policy $\pi^*(\cdot \mid s)$ is considered optimal whenever

$$V^{\pi^*}(s) \geq V^{\pi}(s), \quad \forall s \in S, \forall \pi \in \Pi^{HR}$$

The optimal value of an MDP is defined as:

$$V^*(s) := \sup_{\pi \in \Pi^{HR}} V^{\pi}(s)$$

An optimal policy $\pi^* \in \Pi^{\{HR,MR,HD,MD\}}$ exists when:

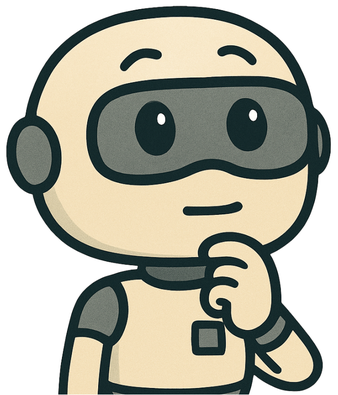
$$V^{\pi^*}(s) \geq V^*(s), \quad \forall s \in S$$

Optimality of Markovian Policies

What policy classes $\Pi^{HR}, \Pi^{HD}, \Pi^{MR}, \Pi^{MD}$ do we need to search to find an optimal policy?

- For any policy $\pi \in \Pi^{HR}$, for each state $s \in S$, there exists a policy $\pi' \in \Pi^{MR}$ for which $v^{\pi}(s) = v^{\pi'}(s)$, (see thm 5.5.3 of puterman),

- Also, there exists a policy $\pi \in \Pi^{MD}$ that is optimal. i.e. achieves <u>at least</u> as much value as <u>all</u> other policies.



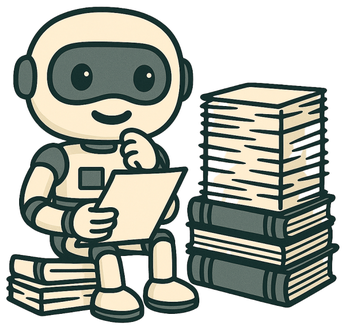
Other Optimality Criteria

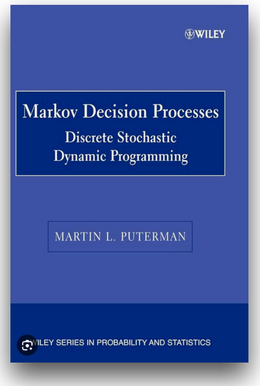
- So far, we've only presented one notion of optimality, assuming $0 \leq \gamma < 1$, and that our goal is to maximize the *sum* of future rewards.

- In this course, we will stick to discounted sums of reward over infinite horizons, due to the mathematical simplicity of this setting, and its dominance in the literature.

- Other notions of optimality in MDPs have been studied, and new ones can be defined (perhaps by you!).
  - E.g., Finite horizons, undiscounted rewards, average rewards, entropy-regularized value functions, risk-aware measures of value, etc.

- If you're interested in learning more, start by reading Puterman's book.





Chapters 4-10

Next Class: Solving for Optimal Policies