Lecture 1

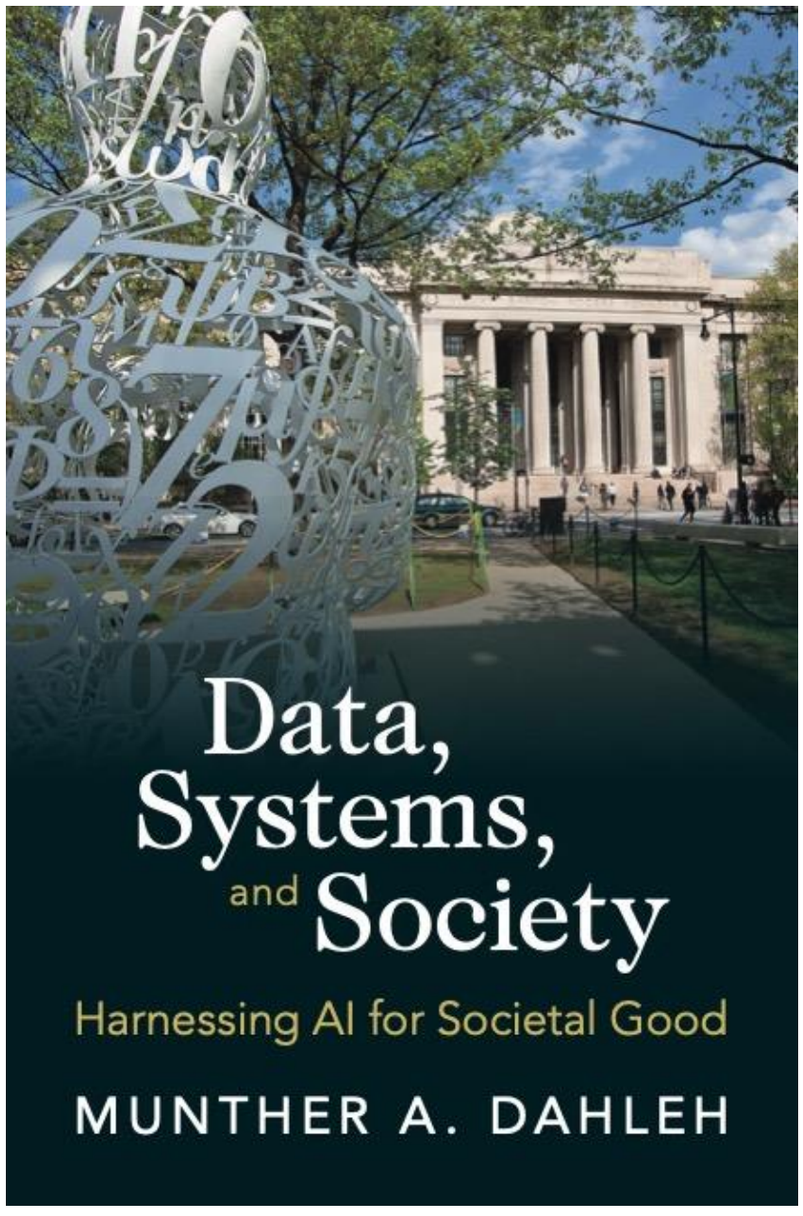
Introduction to Sequential Decision Making (Dynamic Programming (DP))

Introduction

Prof. Munther (Munzer) Dahleh
https://dahleh.lids.mit.edu

Podcast on Societal impact of Data Science and AI

Cambridge University Press



Readings

1. Dynamic Programming and Optimal Control (2007), Vol. I, 4th Edition by Dimitri P. Bertsekas. **[DPOC]**

2. Dynamic Programming and Optimal Control (2005), Vol. II, 4th Edition by Dimitri P. Bertsekas. **[DPOC2]**

3. Neuro-Dynamic Programming (1996) by Dimitri P. Bertsekas and John N. Tsitsiklis. **[NDP]** Available online: https://web.mit.edu/dimitrib/www/NDP.pdf

Announcements: Infomation

- Make sure you are registered and on the Canvas page, as Gradescope + Piazza are both linked there, and it is where homework will be posted
- Course policy, schedule, and lectures will be on the course website (https://web.mit.edu/6.7920/www/)
- PSET 0 out, not graded, for mathematical refresher

Announcements: Policy

- Grade Distribution:
  - Completion grade for eight problem sets (15%)
  - Eight in-class PSet checkpoints (20%)
  - One in-class quiz (35%)
  - Class project (30%)

- Homework is due before lecture at 2PM on assigned dates

- PSet Checkpoints are short 10-minute quizzes on the Pset

- Late policy is 4 late days across all homeworks; after that, it will be penalized 10% every 24 hours

- AI use is permitted as long as fully disclosed

-

Lecture 1

Optimal Control and Reinforcement Learning

Bicycle balancing

Reward: Direct and immediate

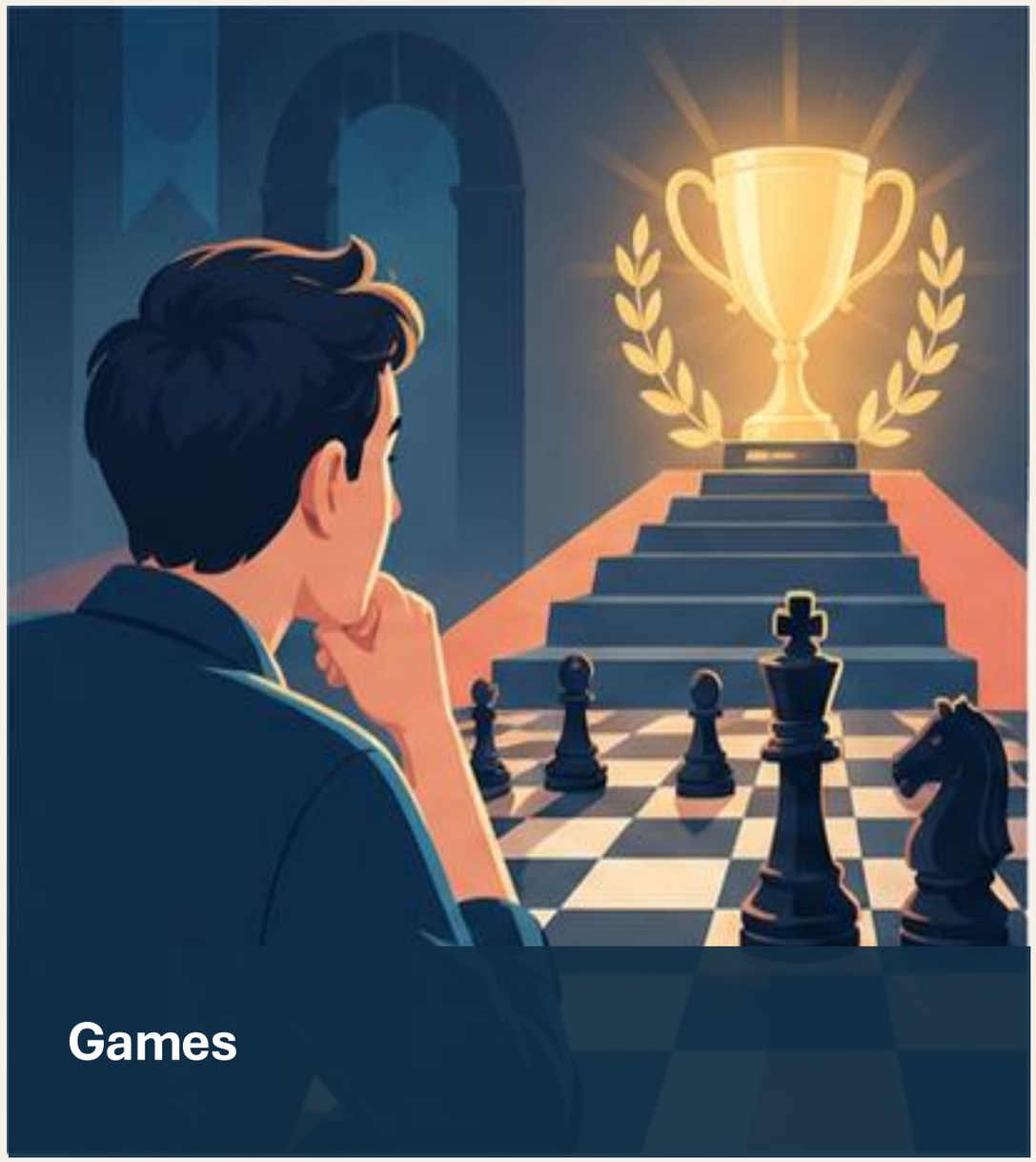
INTENDED REWARD

Remain upright

WHY IT BECOMES DIFFICULT

Each steering action quickly changes the bicycle’s measurable tilt angle.

Can the agent learn the corrective action from immediate feedback?



Games

Reward: Direct and delayed

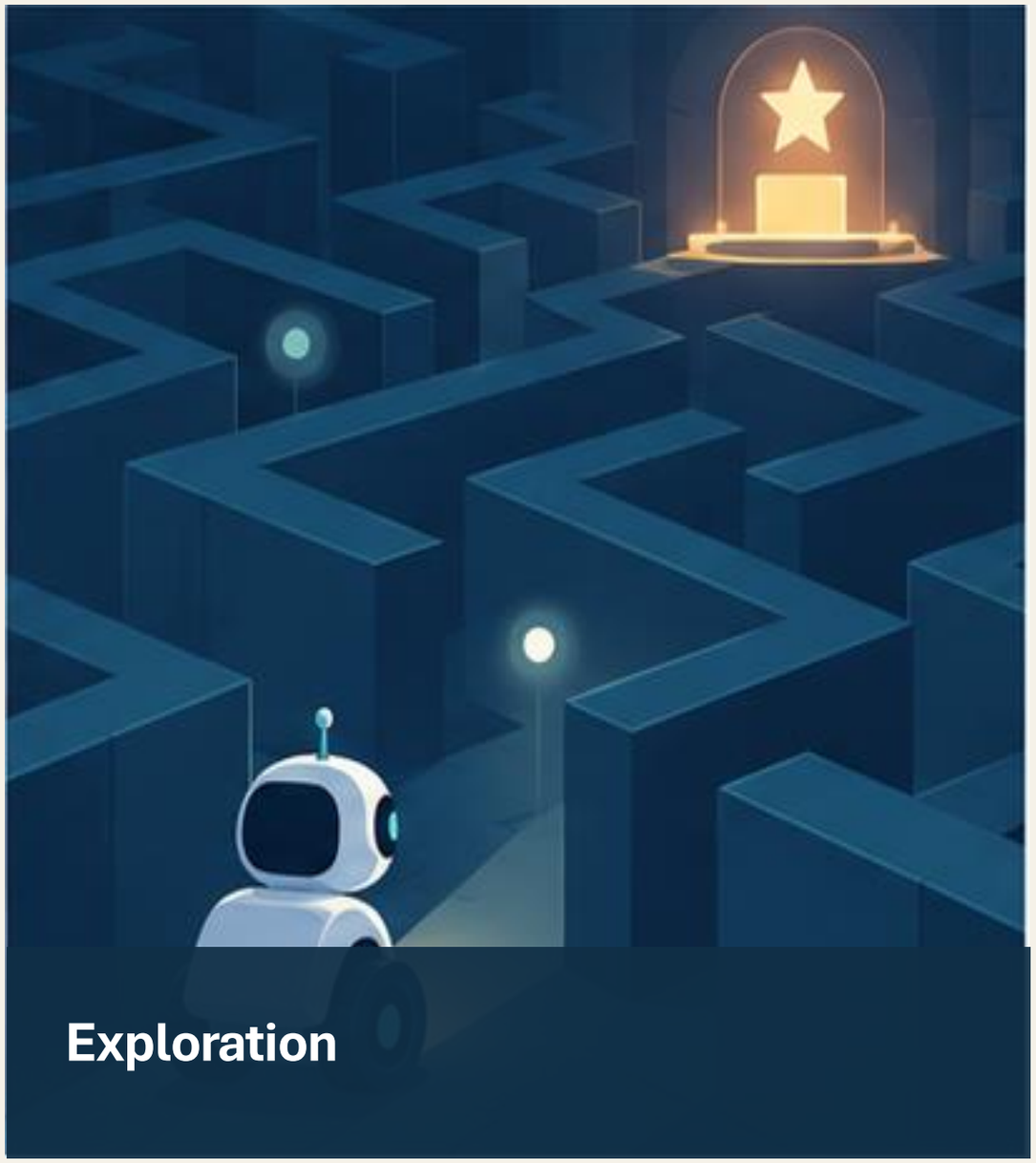
INTENDED REWARD

Win the game

WHY IT BECOMES DIFFICULT

The objective is explicit and the outcome is observed immediately.

Can the agent discover the winning action?



Exploration

Sparse

INTENDED REWARD

Reward: Reach the distant goal

WHY IT BECOMES DIFFICULT

The signal is meaningful but appears so rarely that the agent may never find it.

How does the agent learn before it succeeds?

Autonomous driving

Reward: Engineered

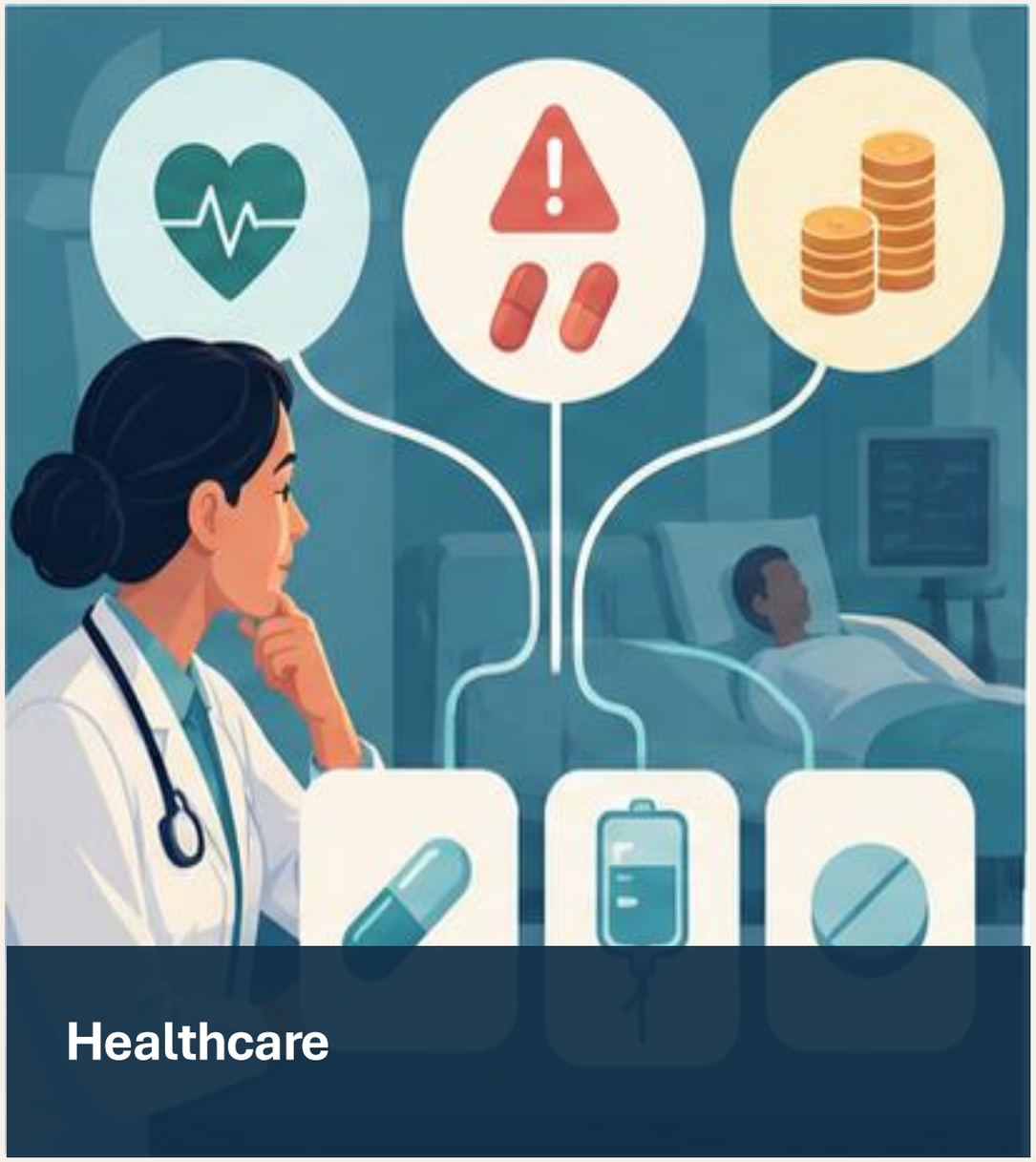
INTENDED REWARD

Arrive safely and comfortably

WHY IT BECOMES DIFFICULT

A designer must convert safe driving into measurable terms without creating shortcuts.

Does the numerical reward capture what we truly want?



Reward: Multiple objectives

INTENDED REWARD

Improve health with acceptable burden

WHY IT BECOMES DIFFICULT

Recovery, side effects, cost, and quality of life can point toward different actions.

How should competing outcomes be balanced?

Language models

Reward: Preference based

INTENDED REWARD

Produce the response people prefer

WHY IT BECOMES DIFFICULT

No direct metric defines a good answer, so the system learns from human comparisons.

Whose preferences should determine the reward?

Personalized education

Reward: Individual and heterogeneous

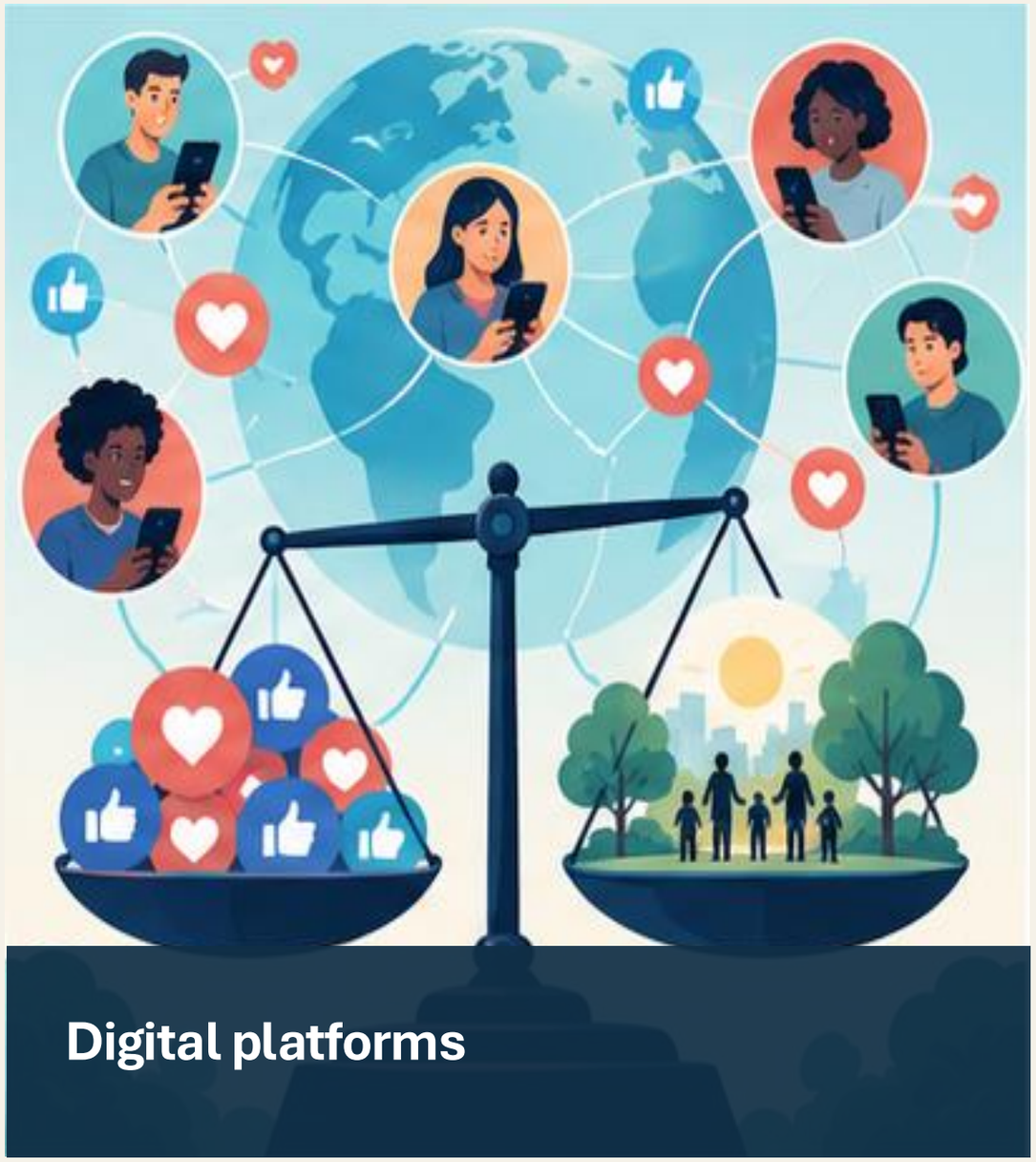
INTENDED REWARD

Help each student learn

WHY IT BECOMES DIFFICULT

The best outcome differs across students and may change as each student progresses.

Can one reward represent every learner?



Digital platforms

Reward: Strategic and societal

INTENDED REWARD

Create value without harming collective welfare

WHY IT BECOMES DIFFICULT

Engagement is easy to measure, while trust, welfare, and externalities are not.

What happens when the measurable proxy becomes the target?

Two distinct sources of difficulty

An application can be difficult because the policy is hard to learn, because the reward is hard to specify, or both.

EASIER TO SPECIFY

HARDER TO SPECIFY

1 2 3 4 5 6 7 8

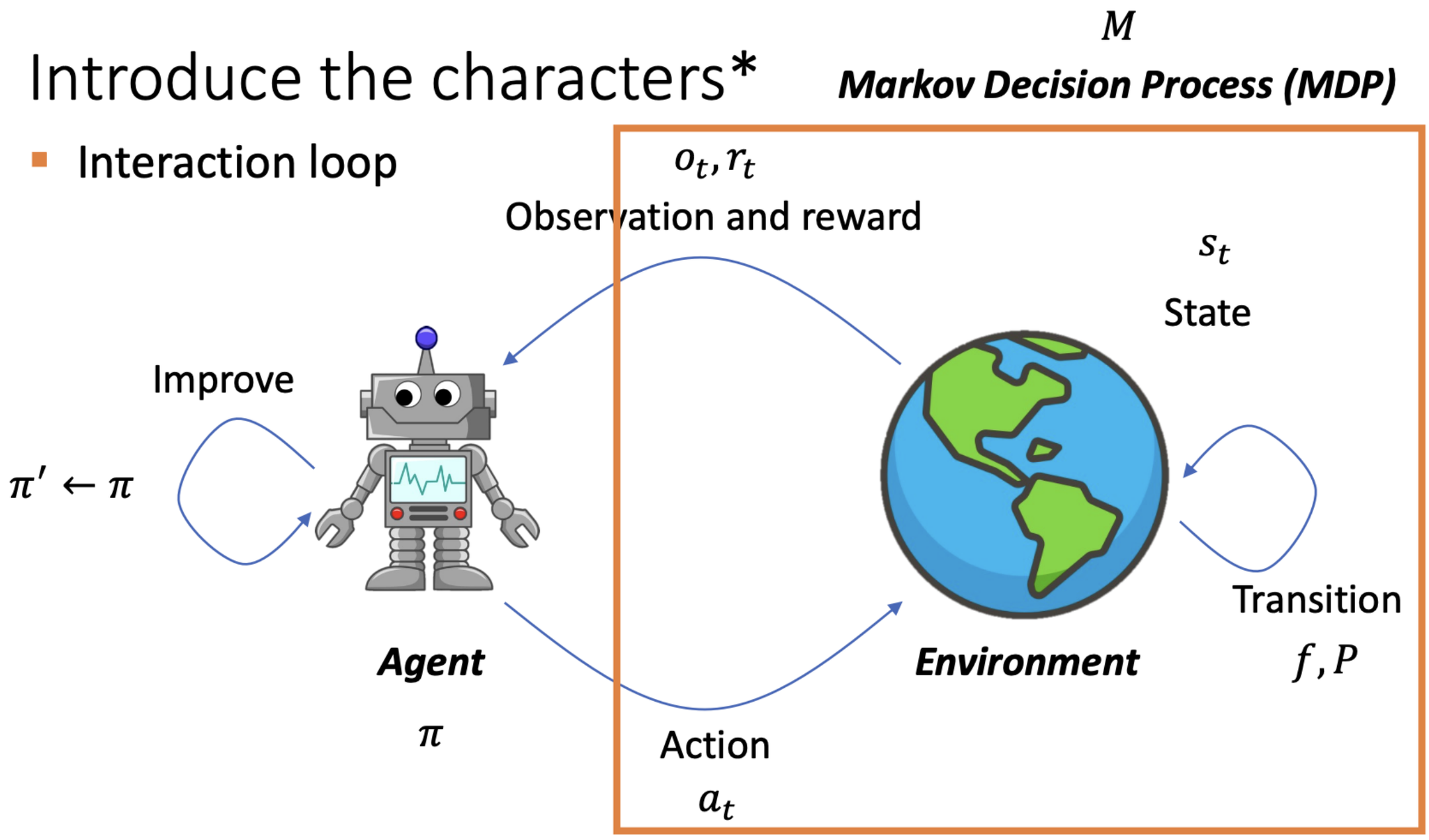
The central challenge

RL can optimize the reward we provide. It cannot guarantee that we provided the right reward.

General Problem

Introduce the characters*

- Interaction loop



Goal: maximize reward over time (returns, cumulative reward)

* pun intended

Classifications

- We have explicit models of the environment, actions, and rewards
  - Dynamic Programming and Optimal Control

- Environment is unknown but we measure the actions and state Evolution
  - Reinforcement learning with full information

- Environment is unknown but we measure the actions and a function of the state Evolution
  - Partially observed problem

Dynamic Systems, States, MDP

Outline

- State Representation
- Definitions
- Continuous vs Discrete MDP
- Control/Decisions
- Stability
- Linear Systems
- Optimal Control

State Space

Dynamic Systems and States

General Dynamic System

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, w_t)$
3. Model of uncertainty: e.g., $\mathbb{P}(\cdot)$ of w_t

Dynamic Systems and States

General Dynamic System

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, w_t)$
3. Model of uncertainty: e.g., $\mathbb{P}(\cdot)$ of w_t

State: A set of variables that summarizes all the necessary information about the system's current and past conditions to predict its future behavior (given past exogenous disturbances), without needing to know the past history of the system.

Dynamic Systems and States

General Dynamic System

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, w_t)$
3. Model of uncertainty: e.g., $\mathbb{P}(\cdot)$ of w_t

State: A set of variables that summarizes all the necessary information about the system's current and past conditions to predict its future behavior (given past exogenous disturbances), without needing to know the past history of the system.

$$\text{Past} \rightarrow \text{s}_\text{t} \rightarrow \text{Future} \quad \text{(Markov property)}$$

Independent Noise

- w_t are independent random variables.
- The system dynamics:

$$s_{t+1} = \frac{1}{2} s_t + w_t$$

Independent Noise

- w_t are independent random variables.
- The system dynamics:

$$s_{t+1} = \frac{1}{2} s_t + w_t$$

Response: Assuming s_0 is fixed,

$$s_t = \left(\frac{1}{2}\right)^t s_0 + \sum_{\tau=0}^{t-1} \left(\frac{1}{2}\right)^{t-\tau-1} w_\tau$$

Independent Noise

- w_t are independent random variables.
- The system dynamics:

$$s_{t+1} = \frac{1}{2} s_t + w_t$$

Response: Assuming s_0 is fixed,

$$s_t = \left(\frac{1}{2}\right)^t s_0 + \sum_{\tau=0}^{t-1} \left(\frac{1}{2}\right)^{t-\tau-1} w_\tau$$

Convolution

Discrete State

Let s_t be the number of jobs in queue at time t. w_t models the arrival of jobs, IID process.

Then the queue dynamics follow:

$$s_{t+1} = \max\{0, s_t + w_t - 1\}$$

Assume $w(t)$ are IID

Is this a state-space representation?

Discrete Example: State dependent noise

Suppose:

$$w_t = g(s_t, \bar{w}_t), \quad \bar{w}_t \text{ is i.i.d.}$$

Then the system evolves as:

$$s_{t+1} = f(s_t, w_t) = f\left(s_t, g(s_t, \bar{w}_t)\right)$$

Converted to Independent Noise Case

We do not need to find g

Example: State dependent noise

Arrival process z_t:

$$\text{If } s_t < 10, \text{ the person joins.} \quad \text{If } s_t \geq 10, \text{ the person flips a coin.}$$

The queue update equation is:

$$s_{t+1} = \max\{0,\ s_t + z_t - 1\}$$

Example: State dependent noise

Arrival process z_t:

If $s_t < 10$, the person joins. If $s_t \geq 10$, the person flips a coin.

The queue update equation is:

$$s_{t+1} = \max\{0,\ s_t + z_t - 1\}$$

where z_t is a state-dependent random variable defined by:

$$z_t = g(s_t, \bar{w}_t)$$

with $\bar{w}_t$ being i.i.d. noise.

Kernel Representations

- Let $\mathcal{S}$ be a finite state space, with $|\mathcal{S}| = n$
- Define transition probabilities $P(s'; s)$

Given:

$$s_{t+1} = f(s_t, w_t)$$

The transition probability is:

$$\begin{aligned} P(s'; s) &= \mathbb{P}(s_{t+1} = s' \mid s_t = s) \\ &= \mathbb{P}\left(f(s, w_t) = s'\right) \end{aligned}$$

Is the converse true?

Kernel Representations

$$\mathbb{P}(s';s) \quad \rightsquigarrow \quad s_{t+1} = f(s_t, w_t)$$

Hint: Simulate using a uniform distribution on [0,1].
Divide the access to reflect the probabilities!

Markov Decision Processes (MDP)

Markov Decision Process (MDP) (control systems)

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, a_t, w_t)$
3. Stage cost: $c_t(s_t, a_t, w_t)$
4. Control constraints: $a_t \in A_t(s_t)$
5. Model of uncertainty: e.g., $\mathbb{P}(\cdot \mid s_t, a_t)$ of w_t

A feedback policy π, is a sequence of functions: $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$
where $\mu_t : s_t \mapsto a_t = \mu_t(s_t) \in A_t(s_t)$

Memoryless Feedback

Is it Markov?

MDP (general strategies)

Control Strategy

$$\mu_t(s_\tau, a_\tau, s_{\tau+1}, a_{\tau+1} \cdots s_t)$$

Is there a problem with this?

$$\text{Dynamics: } s_{t+1} = f_t(s_t, \mu_t(s_\tau, a_\tau \ldots, s_t), w_t)$$

Is it Markov? What is the state?

Transition Kernel Representation

$$s_{t+1} = f_t(s_t, a_t, w_t) \quad \Longleftrightarrow \quad \mathbb{P}(s' \mid s, a)$$

$$\mathbb{P}(f_t(s, a, w_t) = s' \mid s, a) = p_t(s'; s, a)$$

If $|\mathcal{S}| = n$ (finite state space), we write the kernel as:

$$p_t(j; i, u)$$

To compute expectations:

$$\mathbb{E}\left[g(s_{t+1}) \mid s_t = s,\, a_t = a\right] = \sum_{s'} g(s')\, p_t(s'; s, a)$$

Example: Autoregressive Models with Inputs

$$y_t = \sum_{i=1}^{p} d_i \, y_{t-i} + b \, a_{t-1} + \varepsilon_{t-1},$$

$$\text{state:} \quad S_t = \begin{bmatrix} y_{t-p} \\ \vdots \\ y_t \end{bmatrix},$$

$$S_{t+1} = \begin{bmatrix} y_{t-p+1} \\ \vdots \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ d_p & d_{p-1} & \cdots & d_2 & d_1 \end{bmatrix} S_t + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix} a_t + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \varepsilon_t.$$

Example: Autoregressive with moving average models

$$y_t = d_1 y_{t-1} + \cdots + d_p y_{t-p} + b_1 a_{t-1} + \cdots + b_{m_1} a_{t-m_1}$$
$$+ \epsilon_t + c_1 \epsilon_{t-1} + \cdots + c_{m_2} \epsilon_{t_{m_2}}$$

Use superposition: Linearity allows the realization of each separate system

Dynamic Programming and Optimal Control

Introduction to Dynamic Programming

A mathematical framework of **sequential** decision making under uncertainty

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, a_t, w_t)$
3. Stage cost: $c_t(x_t, a_t, w_t)$
4. Control constraints: $a_t \in A_t(x_t)$
5. Model of uncertainty: e.g., $\mathbb{P}(\cdot \mid s_t, a_t)$ of w_t

Introduction to Dynamic Programming

A mathematical framework of **sequential** decision making under uncertainty

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, a_t, w_t)$
3. Stage cost: $c_t(x_t, a_t, w_t)$
4. Control constraints: $a_t \in A_t(x_t)$
5. Model of uncertainty: e.g., $\mathbb{P}(\cdot \mid s_t, a_t)$ of w_t

$$\mathbb{E}\left[c_N(s_N) + \sum_{t=0}^{N-1} c_t(s_t, a_t, w_t)\right]$$

Introduction to Dynamic Programming

A mathematical framework of **sequential** decision making under uncertainty

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, a_t, w_t)$
3. Stage cost: $c_t(x_t, a_t, w_t)$
4. Control constraints: $a_t \in A_t(x_t)$
5. Model of uncertainty: e.g., $\mathbb{P}(\cdot \mid s_t, a_t)$ of w_t

$$\mathbb{E}\left[c_N(s_N) + \sum_{t=0}^{N-1} c_t(s_t, a_t, w_t)\right]$$

■ A feedback policy π, is a sequence of functions: $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$
where $\mu_t : s_t \mapsto a_t = \mu_t(s_t) \in A_t(s_t)$

$$V_N^\pi(s_0) = \mathbb{E}\left[c_N(s_N) + \sum_{t=0}^{N-1} c_t(s_t, \mu_t(s_t), w_t)\right]$$

Introduction to Dynamic Programming

A mathematical framework of **sequential** decision making under uncertainty

1. State: $s_t \in S$
2. Dynamics: $s_{t+1} = f_t(s_t, a_t, w_t)$
3. Stage cost: $c_t(x_t, a_t, w_t)$
4. Control constraints: $a_t \in A_t(x_t)$
5. Model of uncertainty: e.g., $\mathbb{P}(\cdot \mid s_t, a_t)$ of w_t

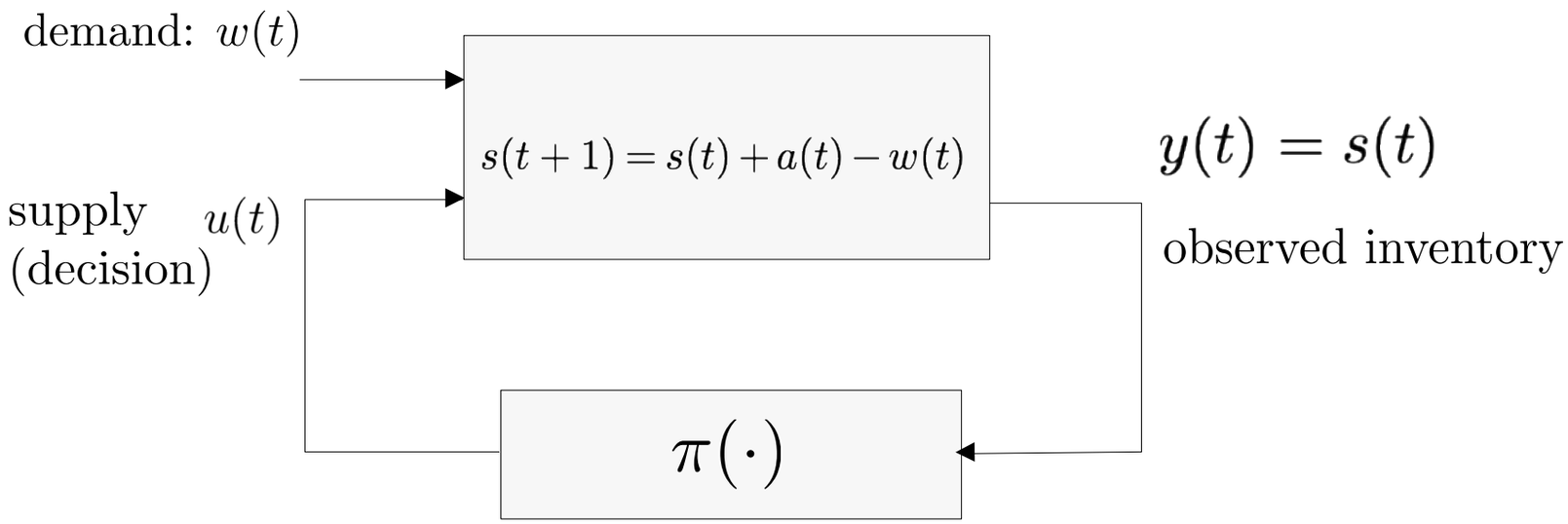
$$\mathbb{E}\left[c_N(s_N) + \sum_{t=0}^{N-1} c_t(s_t, a_t, w_t)\right]$$

- A feedback policy π, is a sequence of functions: $\pi = \{\mu_0, \ldots, \mu_{N-1}\}$ where $\mu_t : s_t \mapsto a_t = \mu_t(s_t) \in A_t(s_t)$

$$V_N^\pi(s_0) = \mathbb{E}\left[c_N(s_N) + \sum_{t=0}^{N-1} c_t(s_t, \mu_t(s_t), w_t)\right]$$

- $V^*(s_0) = \min_\pi V^\pi(s_0)$
- $\pi^* \in \arg\min_\pi V^\pi(s_0)$ i.e., $V^{\pi^*}(s_0) = V^*(s_0)$

Motivation: Inventory control



Cumulative cost over N periods:

$$\mathbb{E}\left[c(s(N)) + \sum_{t=0}^{N-1} \{c(s_t, a_t\}\right]$$

How should we make supply order decisions to minimize this cost / maximize profit?

Motivation: Inventory control

$$\min_{\pi} \quad \mathbb{E}\left[r(s_N) + \sum_{t=0}^{N-1} da_t + g(s_t)\right]$$

$$\text{s.t.} \quad s_{t+1} = s_t + a_t - w_t, \quad a_t \geq 0, \quad w_t \sim \text{i.i.d.}$$

$$g(s) = p\max(hs, -qs)$$

Questions