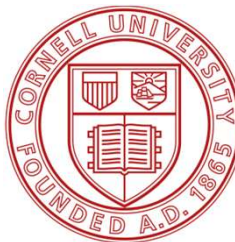
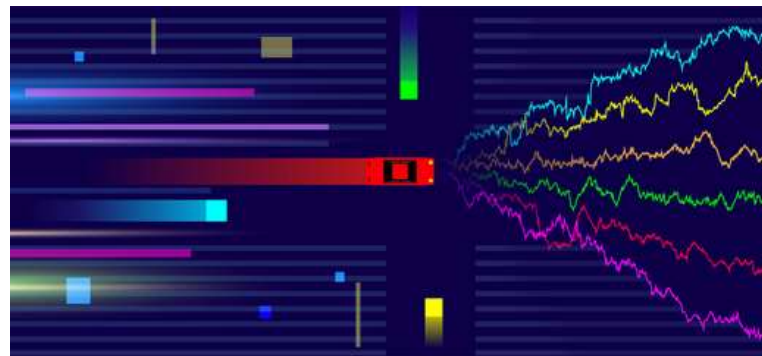


RL for Operations

Day 1: MDP Basics, VI+PI, Deep RL

Sean Sinclair, Sid Banerjee, Christina Yu
Cornell University



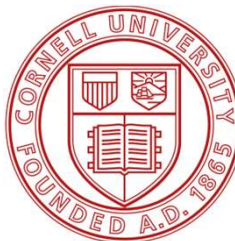
RL for Operations

Day 1: MDP Basics, VI+PI, Deep RL

Sean Sinclair, Sid Banerjee, Christina Yu
Cornell University



<https://github.com/seanrsinclair/RLinOperations>



main 1 branch 0 tags

Go to file Code

Sean Robert Sinclair add code demo slides

adaptive_discretization	finalize adaptive disc demo
custom_simulator	remove figures
exo_mdp	finalize exo mdp demo
slides	add code demo slides
windy_grid_world	finalize adaptive disc demo
.gitignore	update
README.md	update readme

5 days ago

Clone

HTTPS GitHub CLI

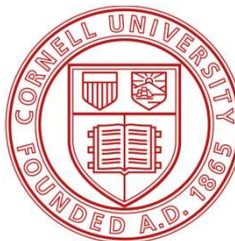
<https://github.com/seanrsinclair/RLinOperations>

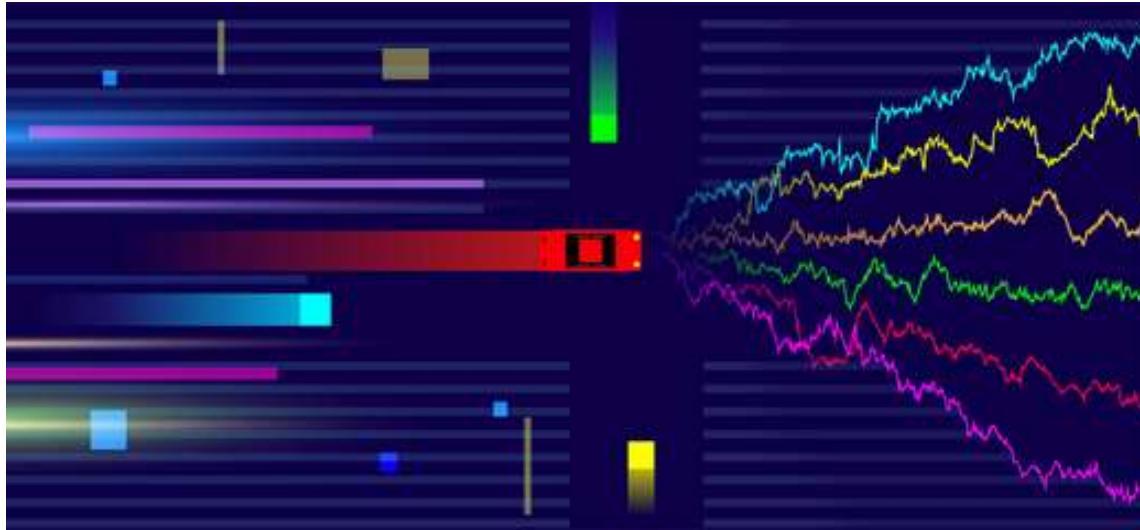
Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

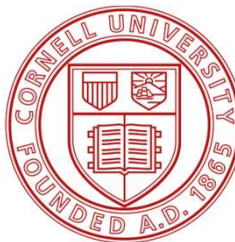
<https://github.com/seanrsinclair/RLinOperations>





Data Driven Decision Making at Simons

- First bootcamp August 22nd to August 26th
- Live streamed on youtube + zoom
- <https://simons.berkeley.edu/workshops/datadriven-2022-bc>



Plan for Today

MDP Basics

- Basic framework for Markov Decision Processes
- Tabular RL Algorithms with policy iteration + value iteration
- DeepRL algorithms (and their “tabular” counterparts)

Simulation Packages

- OpenAI Framework for simulation design
- Existing packages and code-bases for RL algorithm development

Simulation Implementation

- Develop simulator for problem using OpenAI Gym API

Tabular RL Algorithms

- Implement basic tabular RL algorithms to understand key algorithmic design aspects of *value estimates + value iteration*, *policy iteration*

Plan for Today

MDP Basics

- Basic framework for Markov Decision Processes
- Tabular RL Algorithms with policy iteration + value iteration
- DeepRL algorithms (and their “tabular” counterparts)

Simulation Packages

- OpenAI Framework for simulation design
- Existing packages and code-bases for RL algorithm development

Simulation Implementation

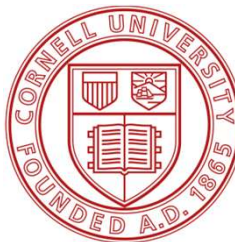
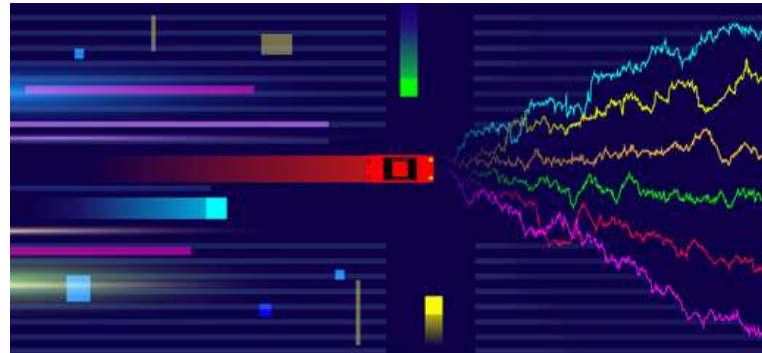
- Develop simulator for problem using OpenAI Gym API

Tabular RL Algorithms

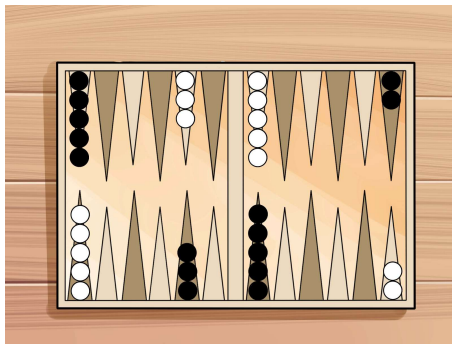
- Implement basic tabular RL algorithms to understand key algorithmic design aspects of *value estimates + value iteration*, *policy iteration*

MDP Basics

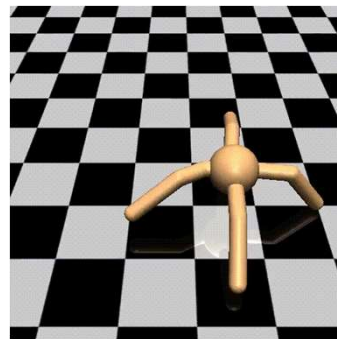
Siddhartha Banerjee
Cornell University



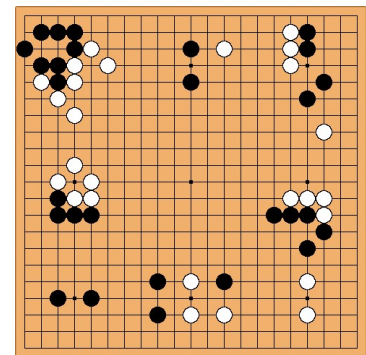
Success of RL



Backgammon



MuJoCo Simulator



AlphaGo Zero

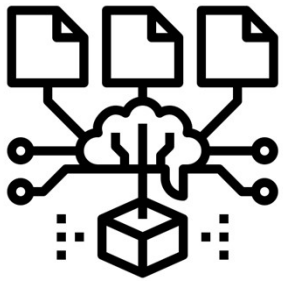
Focused on game playing + robotics

[Silver2017, Tesauro1995]

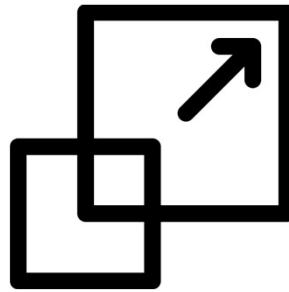
This Workshop

This workshop focuses on **RL for Operations**

We care about:



OR Models



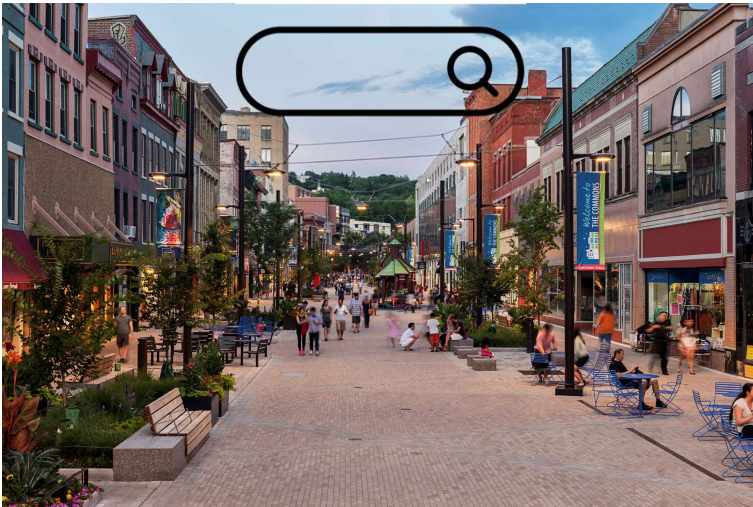
Computation + Scale



Impact

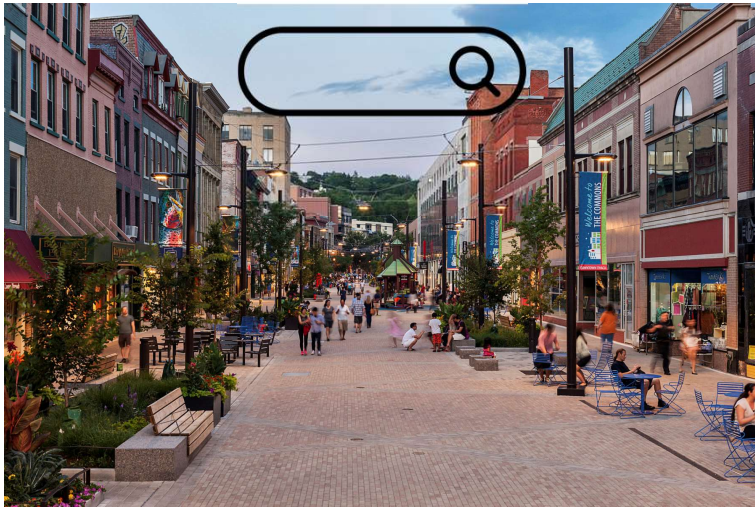
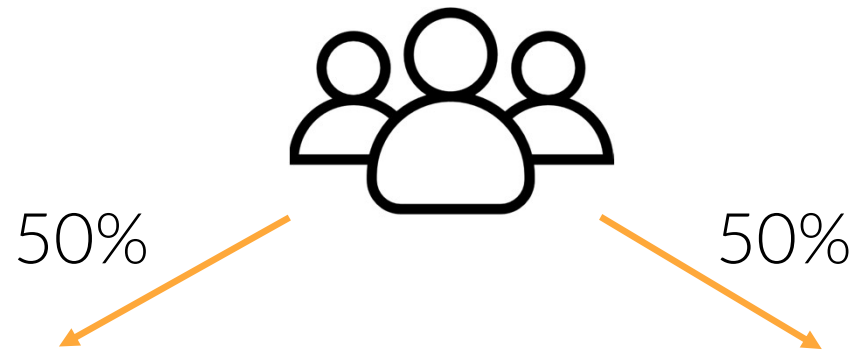
A Story

Typical question: “which decision is better”



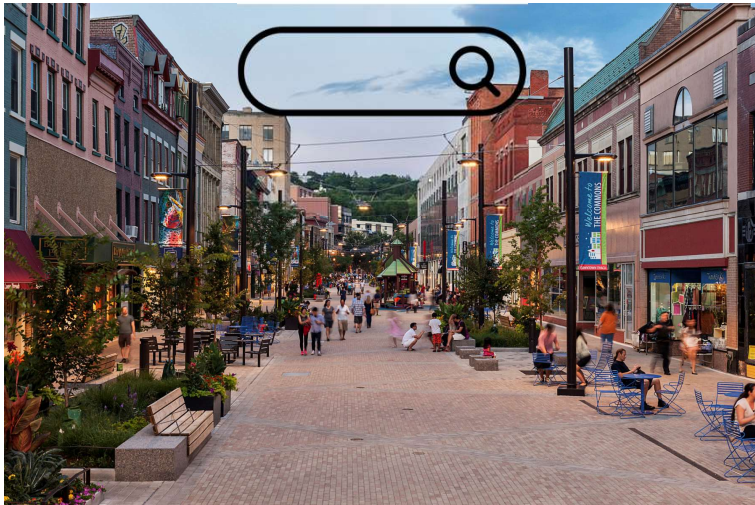
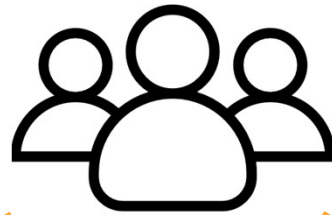
City of Ithaca homepage photo

A/B Test



Take users, divide randomly, observe
which has longer visit times

Bandit Algorithms



Adaptively partition users based on
observed feedback thus far

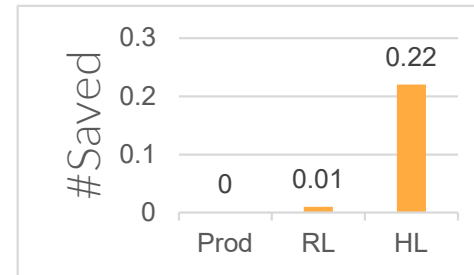
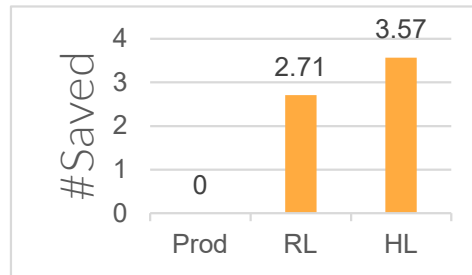
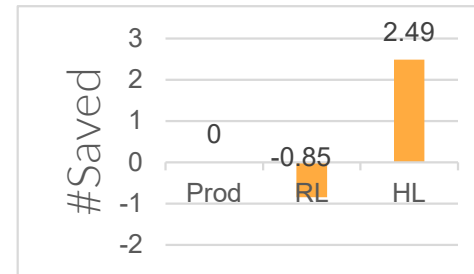
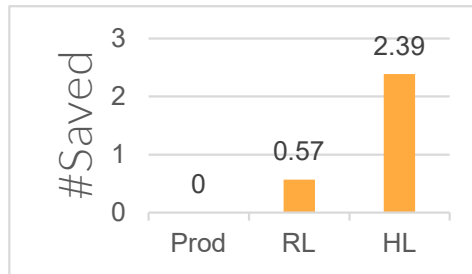
Bandit Algorithms



Renaissance of use in industry, motivated
new theory + practice research

Bandit Algorithms

Bandit problems are a dynamic “supervised learning” model, no feedback effects





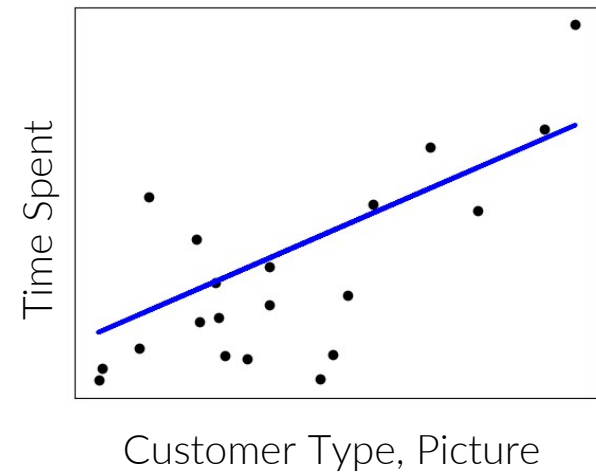
Developing algorithms for RL in Operations can be the *next big success* in industry

Supervised Learning

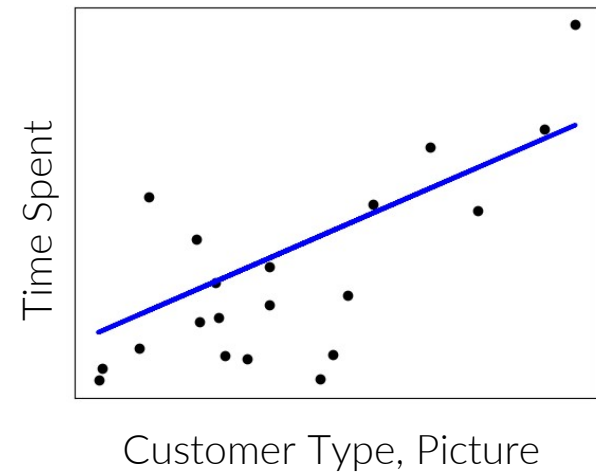
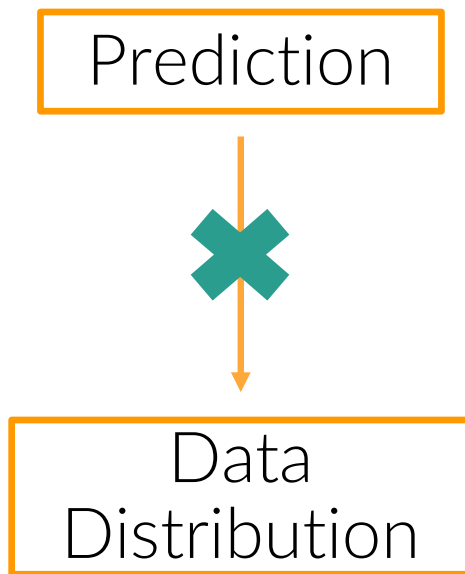
( ,  , 14 mins)

( ,  , 8 mins)

( ,  , 24 mins)

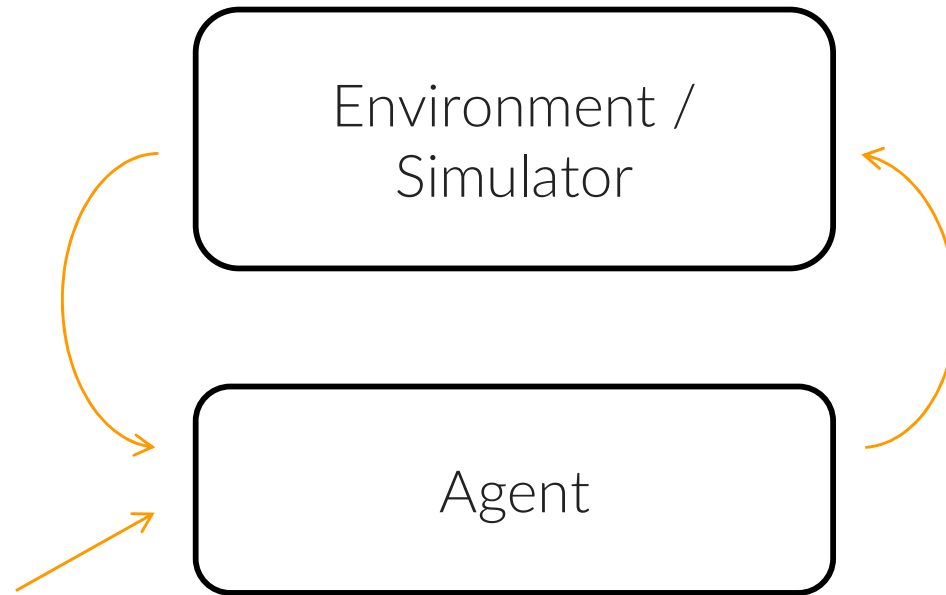


Supervised Learning

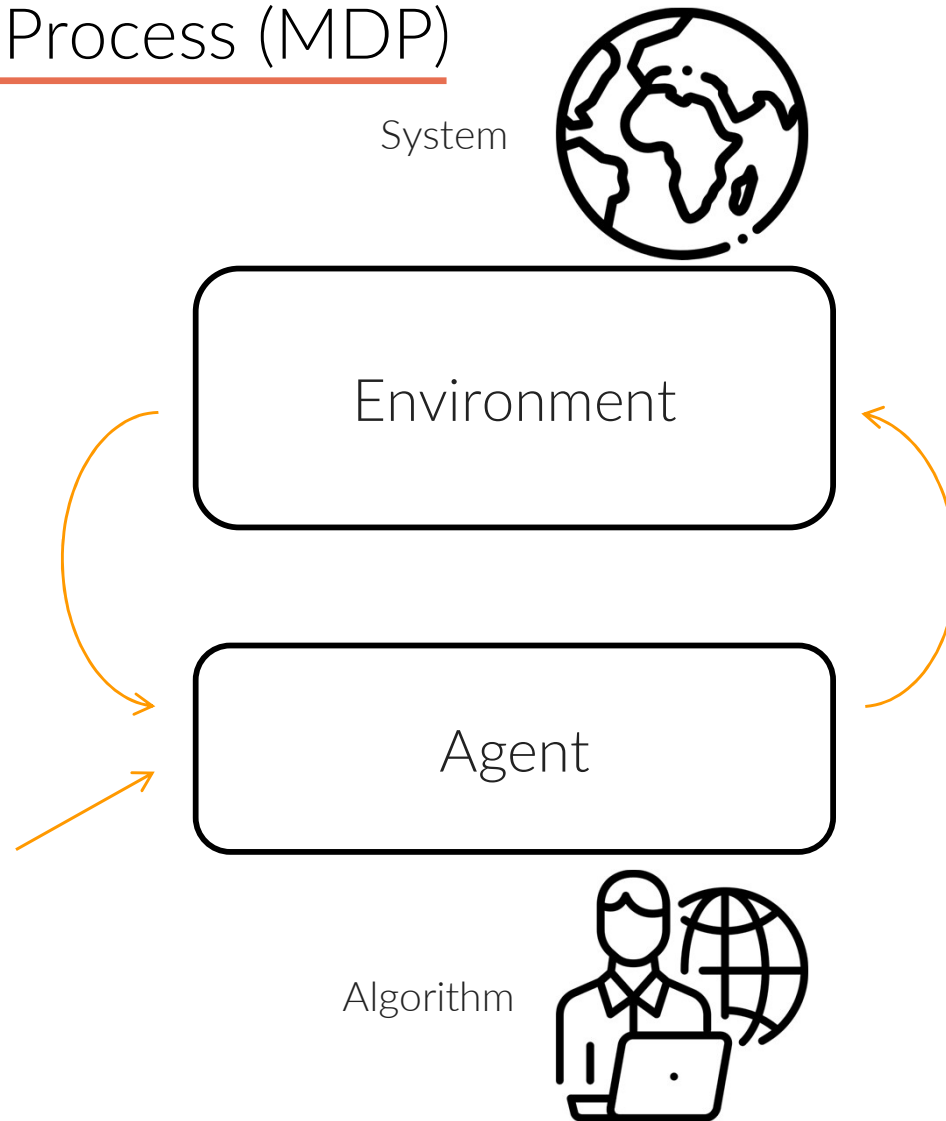


Theory and practice relies on **prediction not affecting data distribution**

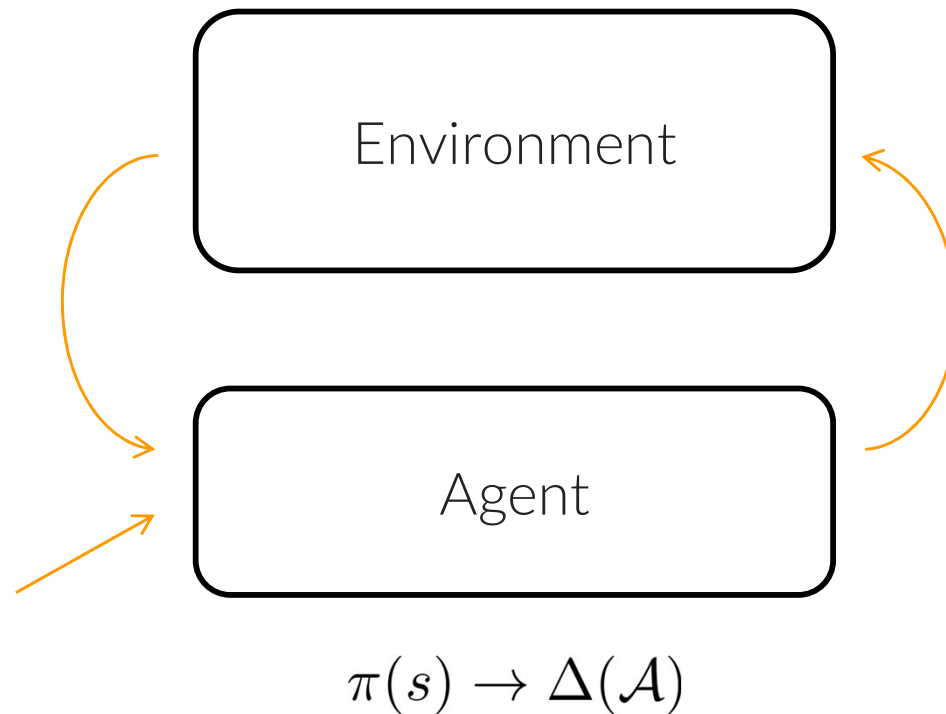
Markov Decision Process (MDP)



Markov Decision Process (MDP)



Markov Decision Process (MDP)



Policy: Determine **action** based on **state**

Markov Decision Process (MDP)

Environment: Determine **reward** and new **state**

$$r(s, a), s' \sim T(\cdot \mid s, a)$$



$$\pi(s) \rightarrow \Delta(\mathcal{A})$$

Policy: Determine **action** based on **state**

Markov Decision Process (MDP)

Environment: Determine **reward** and new **state**



Policy: Determine **action** based on **state**

MDP vs Supervised Learning

	Learn from Experience	Generalize	Interactive	Exploration	Credit Assignment
Supervised Learning	✓	✓	✗	✗	✗
Reinforcement Learning	✓	✓	✓	✓	✓

MDP vs Supervised Learning

	Learn from Experience	Generalize	Interactive	Exploration	Credit Assignment
Supervised Learning	✓	✓	✗	✗	✗
Reinforcement Learning	✓	✓	✓	✓	✓

Infinite Horizon Discounted

A **MDP** is defined by: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, T, s_0, \gamma\}$

\mathcal{S} State space

\mathcal{A} Action space

$r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ Reward

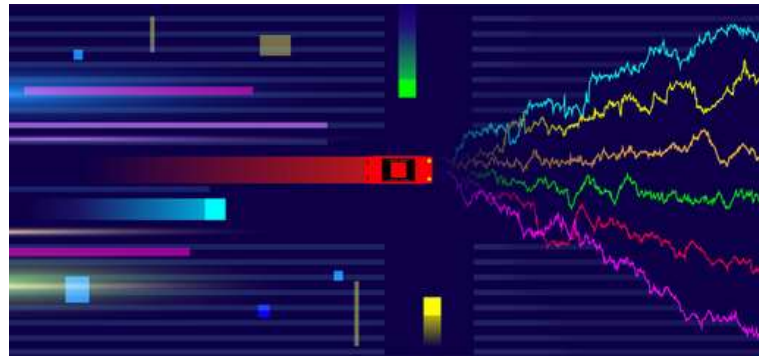
$T : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ Transitions

$\gamma \in [0, 1)$ Discount

Infinite Horizon Discounted

A **MDP** is defined by: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, T, s_0, \gamma\}$

$$r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 \dots$$



$\gamma \in [0, 1)$ Discount

Infinite Horizon Discounted

A **MDP** is defined by: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, T, s_0, \gamma\}$

\mathcal{S} State space

\mathcal{A} Action space

$r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ Reward

$T : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ Transitions

$\gamma \in [0, 1)$ Discount

$\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ Policy

Policies are
non-stationary

Finite Horizon

A **MDP** is defined by: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, T, s_0, \gamma\}$

\mathcal{S} State space

\mathcal{A} Action space

$r_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ Reward

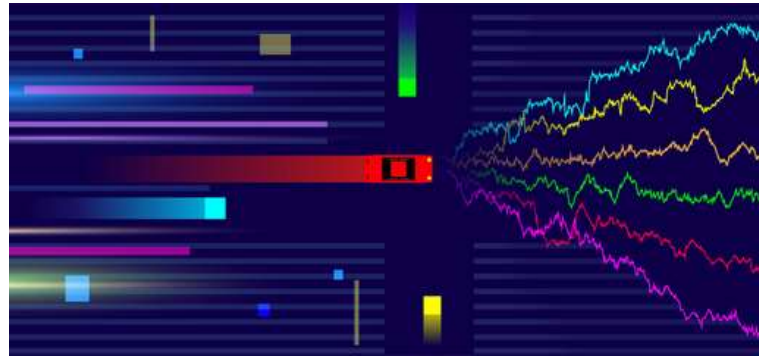
$T_h : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ Transitions

H Time horizon

$\pi_h : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ Policy

Finite Horizon

$$r_1 + r_2 + r_3 + \dots + r_H$$



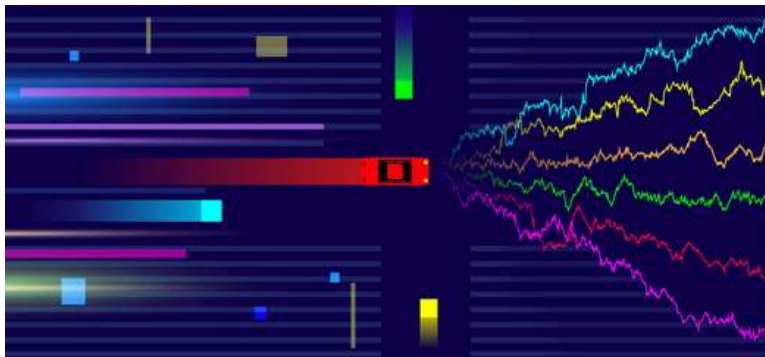
H Time horizon

Discounted?

Infinite Horizon

- Future rewards discounted at rate of γ
- “Less” importance on future rewards
- Transition, rewards, policy, not allowed to depend on timestep

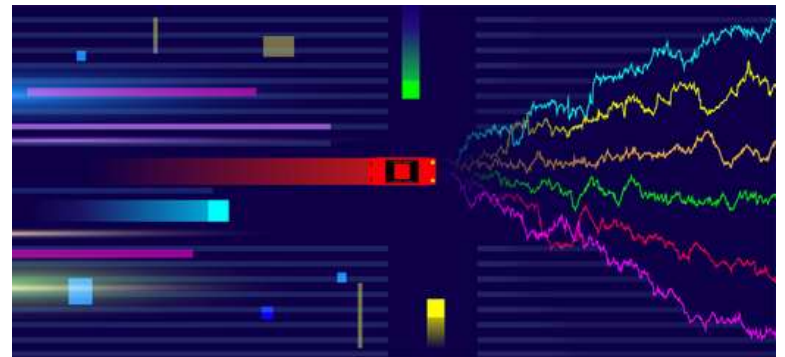
$$r_1 + \gamma r_2 + \gamma^3 r_3 + \gamma^4 r_4 \dots$$



Finite Horizon

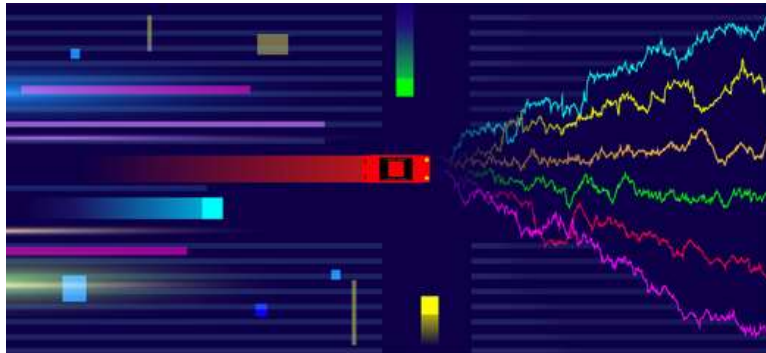
- Future rewards are *not* discounted
- “More” importance on future rewards
- Transition, rewards, policy *allowed* to depend on timestep

$$r_1 + r_2 + r_3 + \dots + r_H$$



Discounted?

$$\lim_{T \rightarrow \infty} \frac{1}{T} (r_1 + r_2 + \dots + r_T)$$

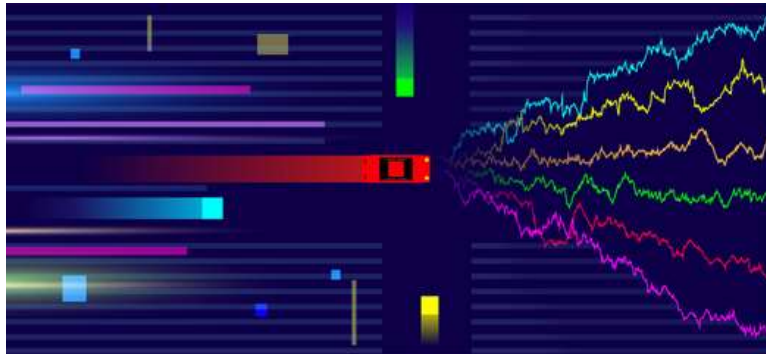


Average Cost

- Future rewards are *not* discounted
- Transition, rewards, policy, not allowed to depend on timestep
- Typically studied in queueing + scheduling literature

Discounted?

$$\lim_{T \rightarrow \infty} \frac{1}{T} (r_1 + r_2 + \dots + r_T)$$



People **tune**
discount factor in
practice

Average Cost

- Future rewards are *not* discounted
- Transition, rewards, policy, not allowed to depend on timestep
- Typically studied in pure OR literature

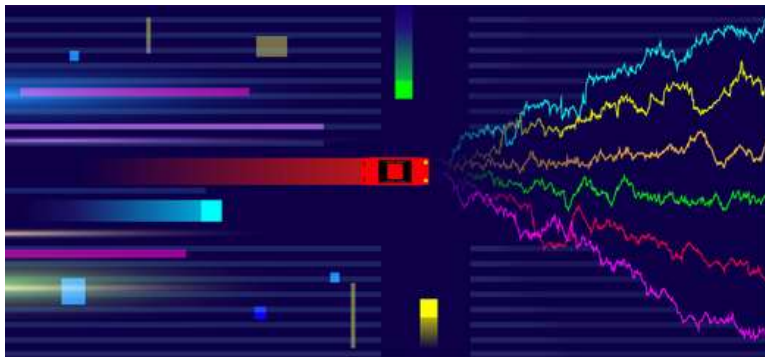
Informal Theorem: There exists a discount such that the optimal policy in infinite horizon discounted problem is optimal under average cost

Discounted?

Infinite Horizon

- Future rewards discounted at rate of γ
- “Less” importance on future rewards
- Transition, rewards, policy, not allowed to depend on timestep

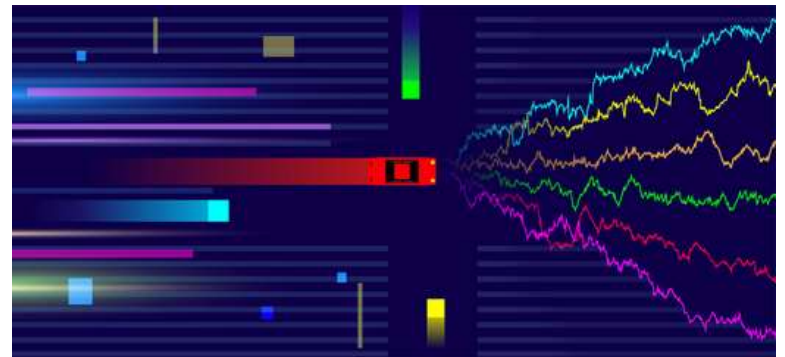
$$r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 \dots$$



Finite Horizon

- Future rewards are *not* discounted
- “More” importance on future rewards
- Transition, rewards, policy *allowed* to depend on timestep

$$r_1 + r_2 + r_3 + \dots + r_H$$



Value Function

The **Value Function** is expected return for policy

$$V^\pi(s) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid S_0 = s, A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$
$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid (S_0, A_0) = (s, a), A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$

Expectation over randomness in policy and transitions

Value Function

The **Value Function** is expected return for policy

$$V^\pi(s) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid S_0 = s, A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$
$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid (S_0, A_0) = (s, a), A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$

Starting State Actions by policy Next state by environment

Expectation over randomness in policy and transitions

Bellman Equation

$$V^\pi(s) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid S_0 = s, A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$
$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid (S_0, A_0) = (s, a), A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$

The Bellman Equations note that:

$$V^\pi(s) = \mathbb{E}_{A \sim \pi(s)} [r(s, A) + \gamma \mathbb{E}_{S' \sim T(\cdot \mid s, A)} [V^\pi(S')]]$$

$$Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{S' \sim T(\cdot \mid s, a)} [V^\pi(S')]$$

Optimal Policy

For an infinite horizon discounted MDP, there exists a deterministic stationary policy:

$$\pi^* : \mathcal{S} \rightarrow \mathcal{A}, \text{ s. t. } V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s, \pi$$

See [Puterman1994]

Our goal is to find this policy, either looking at:

- Sample complexity (statistics)
- Optimization complexity

Optimal Policy

For an infinite horizon discounted MDP, there exists a deterministic stationary policy:

$$\pi^* : \mathcal{S} \rightarrow \mathcal{A}, \text{ s. t. } V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s, \pi$$

See [Puterman1994]

Denote $V^* = V^{\pi^*}, Q^* = Q^{\pi^*}$

Bellman Optimality

The optimal policy satisfies Bellman Optimality equation:

$$V^*(s) = \max_{a \in \mathcal{A}} r(s, a) + \gamma \mathbb{E}_{S' \sim T(\cdot | s, a)} [V^*(S')]$$

Q-greedy policy: $\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$

See [Puterman1994]

Bellman Optimality

The optimal policy satisfies Bellman Optimality equation:

$$V^*(s) = \max_{a \in \mathcal{A}} r(s, a) + \gamma \mathbb{E}_{S' \sim T(\cdot | s, a)} [V^*(S')]$$

Q-greedy policy: $\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$

See [Puterman1994]

$$V^\pi(s) = \mathbb{E}_{A \sim \pi(s)} [r(s, A) + \gamma \mathbb{E}_{S' \sim T(\cdot | s, A)} [V^\pi(S')]]$$

Fixed Point Uniqueness

$$\text{If } V(s) = \max_{a \in \mathcal{A}} r(s, a) + \gamma \mathbb{E}_{S' \sim T(\cdot | s, a)} [V(S')]$$

$$\text{then } V(s) = V^*(s) \forall s$$

See [Puterman1994]

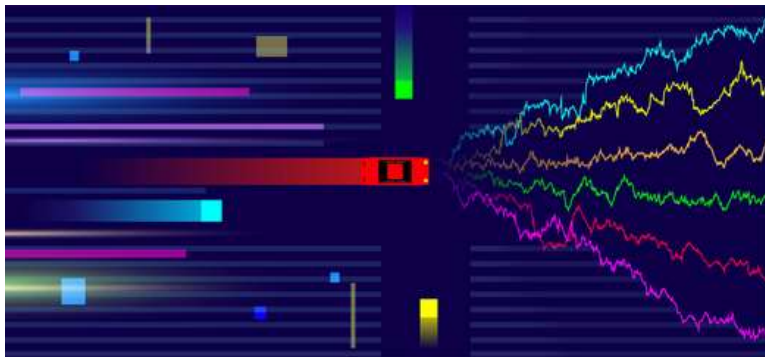
Highlights uniqueness of fixed point of
the Bellman equation

Discounted?

Infinite Horizon

- Future rewards discounted at rate of γ
- “Less” importance on future rewards
- Transition, rewards, policy, not allowed to depend on timestep

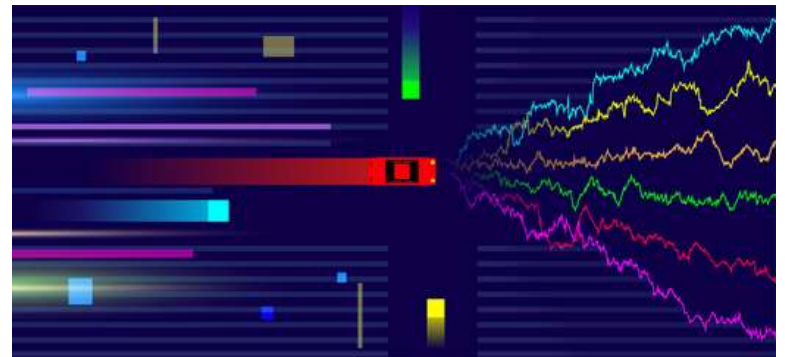
$$r_1 + \gamma r_2 + \gamma^3 r_3 + \gamma^4 r_4 \dots$$



Finite Horizon

- Future rewards are *not* discounted
- “More” importance on future rewards
- Transition, rewards, policy *allowed* to depend on timestep

$$r_1 + r_2 + r_3 + \dots + r_H$$



Value Function

The **Value Function** is expected return for policy

$$V_h^\pi(s) = \mathbb{E} \left[\sum_{h'=h}^H r_{h'}(S_{h'}, A_{h'}) \mid S_h = s, A_{h'} \sim \pi(S_{h'}), S_{h'+1} \sim T_{h'}(\cdot \mid S_{h'}, A_{h'}) \right]$$
$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{h'=h}^H r_{h'}(S_{h'}, A_{h'}) \mid (S_h, A_h) = (s, a), A_{h'} \sim \pi(S_{h'}), S_{h'+1} \sim T_{h'}(\cdot \mid S_{h'}, A_{h'}) \right]$$

Expectation over randomness in policy and transitions

Bellman Equation

$$V^\pi(s) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid S_0 = s, A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$
$$Q^\pi(s, a) = \mathbb{E} \left[\sum_{h=0}^{\infty} \gamma^h r(S_h, A_h) \mid (S_0, A_0) = (s, a), A_h \sim \pi(S_h), S_{h+1} \sim T(\cdot \mid S_h, A_h) \right]$$

The Bellman Equations note that:

$$V_h^\pi(s) = \mathbb{E}_{A \sim \pi_h(s)} [r_h(s, A) + \mathbb{E}_{S' \sim T_h(\cdot \mid s, A)} [V_{h+1}^\pi(S')]]$$

$$Q_h^\pi(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot \mid s, a)} [V_{h+1}^\pi(S')]$$

Optimal Policy

For a finite horizon MDP, there exists a deterministic (potentially nonstationary) policy:

$$\pi_h^* : \mathcal{S} \rightarrow \mathcal{A}, \text{ s. t. } V_h^{\pi^*}(s) \geq V_h^{\pi}(s) \forall s, \pi$$

See [Puterman1994]

Our goal is to find this policy, either looking at:

- Sample complexity (statistics)
- Optimization complexity

Optimal Policy

For a finite horizon MDP, there exists a deterministic (potentially nonstationary) policy:

$$\pi_h^* : \mathcal{S} \rightarrow \mathcal{A}, \text{ s. t. } V_h^{\pi^*}(s) \geq V_h^\pi(s) \forall s, \pi$$

See [Puterman1994]

Denote $V_h^* = V_h^{\pi^*}, Q_h^* = Q^{\pi^*}$

Bellman Optimality

The optimal policy satisfies Bellman Optimality equation:

$$V_h^*(s) = \max_{a \in \mathcal{A}} r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot | s, a)} [V_{h+1}^*(S')]$$

Q-greedy policy: $\pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$

See [Puterman1994]

Fixed Point Uniqueness

If $V_h(s) = \max_{a \in \mathcal{A}} r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot | s, a)} [V_{h+1}(S')]$
then $V(s) = V^*(s) \forall s$

See [Puterman1994]

Highlights uniqueness of fixed point of
the Bellman equation

References

- [Puterman1994] Martin Puterman. “Markov Decision Processes: Discrete Stochastic Dynamic Programming”. *John Wiley + Sons*, 1994.
- [Sutton2018] Richard Sutton. “Reinforcement Learning: An Introduction.” *MIT Press*, 2018.
- [Agarwal2021] Alekh Agarwal, Nan Jiang, Sham M. Kakade, Wen Sun. “Reinforcement Learning: Theory and Algorithms”. 2021.
- [Slivkins2019] Aleksandrs Slivkins. “Introduction to Multi-Armed Bandits.” *Foundations and Trends in ML*, 2019.
- [Powell2021] Warren Powell. “Reinforcement Learning and Stochastic Optimization.” 2021.
- [Meyn2021] Sean Meyn. “Control Systems and Reinforcement Learning”. *Cambridge University Press*, 2021.

Course Slides

Cornell CS6789: Foundations of Reinforcement Learning

https://wensun.github.io/CS6789_fall_2021.html

Stanford CS 234: Reinforcement Learning

<https://web.stanford.edu/class/cs234/>

UCL COMPM050: Course on RL

<https://www.davidsilver.uk/teaching/>