

Reinforcement Learning

Lecture 1: Foundations of Reinforcement Learning

Chengchun Shi

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Lecture Outline (Cont'd)

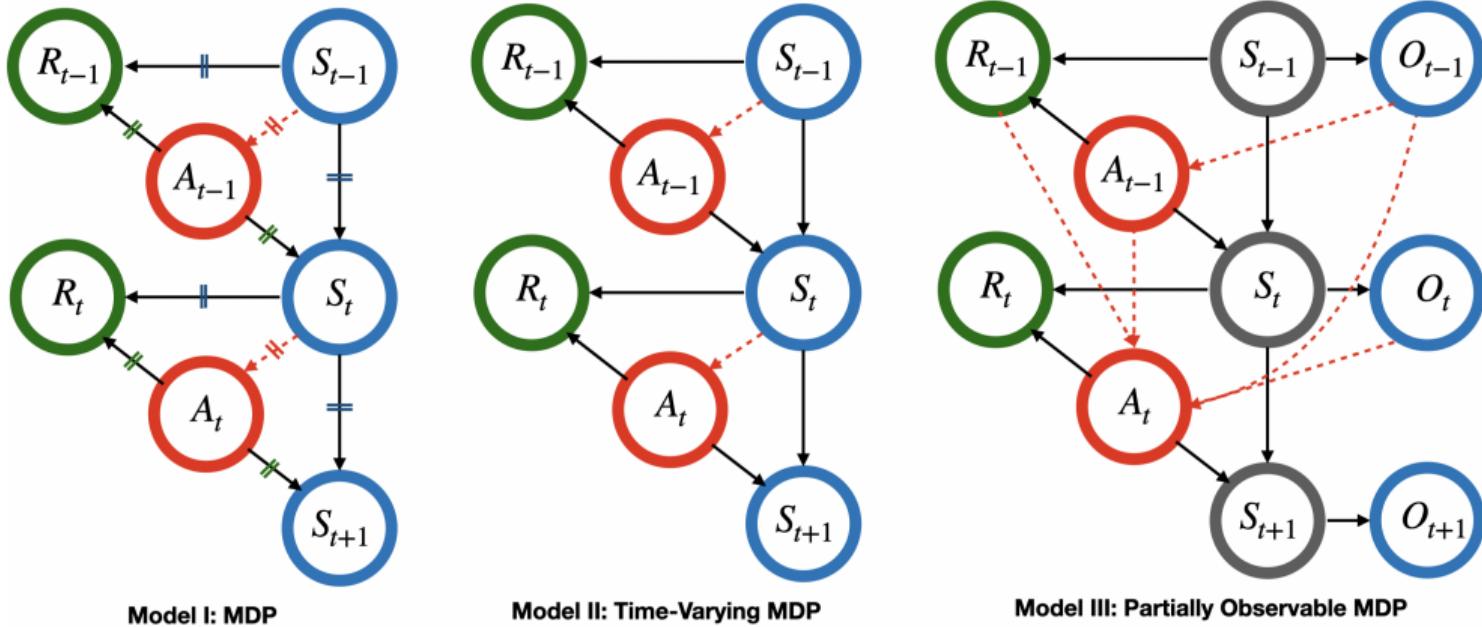


Figure: Causal diagrams for MDPs, TMDPs and POMDPs. Solid lines represent the causal relationships. Dashed lines indicate the information needed to implement the optimal policy. The parallel sign \parallel indicates that the conditional probability function given parent nodes is equal.

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

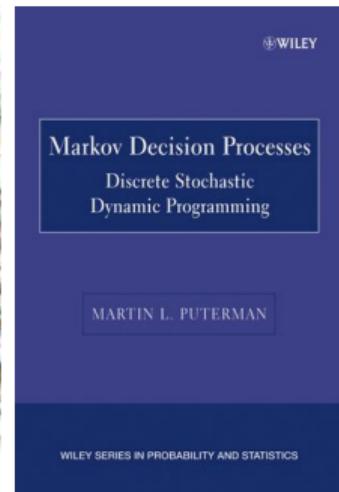
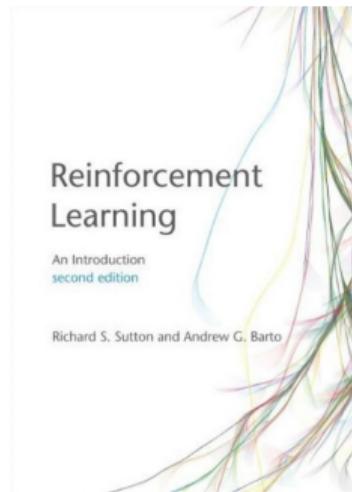
2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Textbooks

- **Reinforcement Learning: An Introduction**
(Second Edition) by Sutton and Barto (2018)
 - Ebook free online ([link](#))
 - 50K citations so far
- **Markov decision processes: discrete stochastic dynamic programming** by Puterman (2014)

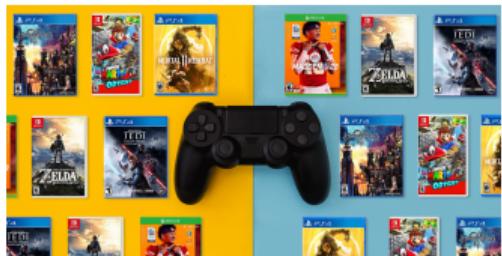


Useful Resources

- Deepmind & UCL reinforcement learning (RL) course by David Silver
 - Course webpage [link](#)
 - Videos available on YouTube
 - Slides available on webpage
- UC Berkeley PhD-level deep RL course by Sergey Levine
 - Course webpage [link](#)
 - Some more resources [link](#)
- Working draft on “**Reinforcement Learning: Theory and Algorithms**” by Alekh, Nan, Sham and Wen [link](#)



Applications



(a) Games



(b) Health Care



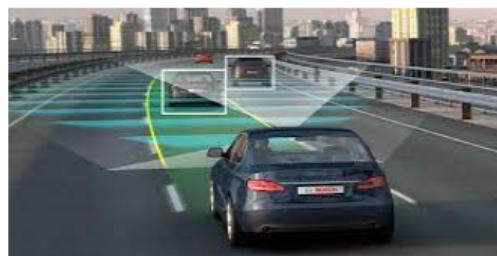
(c) Ridesharing



(d) Robotics



(e) Finance



(f) Automated Driving

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

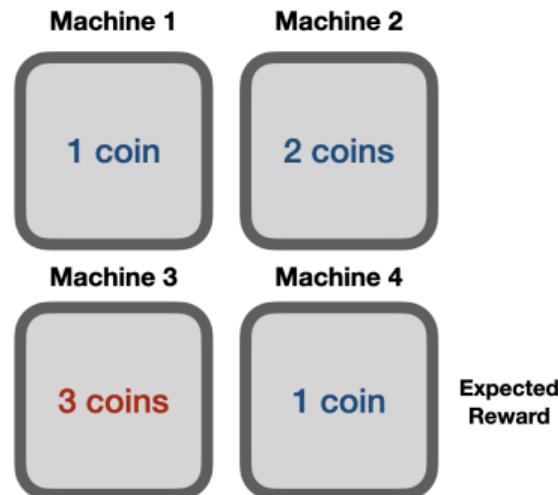
Multi-Armed Bandit (MAB) Problem



- The **simplest** RL problem
- A casino with **multiple** slot machines
- Playing each machine yields an independent **reward**.
- Limited knowledge (unknown reward distribution for each machine) and resources (**time**)
- **Objective:** determine which machine to pick at each time to maximize the expected **cumulative rewards**

Multi-Armed Bandit Problem (Cont'd)

- k -armed bandit problem (k machines)
- $A_t \in \{1, \dots, k\}$: arm (machine) pulled (experimented) at time t
- $R_t \in \mathbb{R}$: reward at time t
- $Q(a) = \mathbb{E}(R_t | A_t = a)$ expected reward for each arm a (**unknown**)
- **Objective**: maximize $\sum_{t=1}^T \mathbb{E}R_t$.



Greedy Action Selection

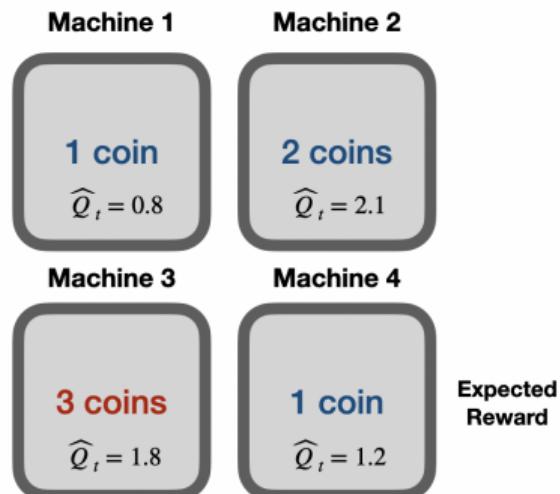
- **Action-value methods:** estimate the expected reward (i.e., value) of actions and use these estimates to select actions
- Estimated reward at time t :

$$\hat{Q}_t(a) = \frac{\sum_{i=1}^t R_i \mathbb{I}(A_i = a)}{\sum_{i=1}^t \mathbb{I}(A_i = a)}$$

- **Greedy policy:**

$$A_t = \arg \max_a \hat{Q}_{t-1}(a).$$

- Might be **suboptimal** in the long run.



Exploration-Exploitation Dilemma

- **Exploitation:** To maximize reward, the agent prefers the greedy policy that selects actions that maximizes the estimated expected reward.
- **Exploration:** To discover which actions yield a higher reward, the agent must try actions that it has less selected to improve the estimation accuracy.
- **Trade-off** between exploration and exploitation:
 - Neither exploration nor exploitation can be used exclusively.
 - The agent must try various actions and progressively favour high-reward actions.
- Practical algorithms: **ϵ -greedy, upper confidence bound (UCB), Thompson sampling.**

ϵ -Greedy

- **Input:** Choose a small value parameter $\epsilon \in (0, 1)$.
- At each step **perform**:
 - With probability $1 - \epsilon$: adopt the **greedy policy**;
 - With probability ϵ : choose a **randomly selected arm** from the set of all arms.
- Combines exploration and exploitation:
 - At each time, each arm is selected with probability at least $k^{-1}\epsilon$.
 - Greedy action is selected with probability $1 - \epsilon + k^{-1}\epsilon$.

Incremental Implementation

- Average reward received from arm \mathbf{a} by time t :

$$\hat{Q}_t(\mathbf{a}) = \mathbb{N}_t^{-1}(\mathbf{a}) \sum_{i=1}^t \mathbb{I}(\mathbf{A}_i = \mathbf{a}) \mathbf{R}_i,$$

where $\mathbb{N}_t(\mathbf{a}) = \sum_{i=1}^t \mathbb{I}(\mathbf{A}_i = \mathbf{a})$.

- If arm \mathbf{a} is selected at time $t + 1$, then

$$\begin{aligned}\hat{Q}_{t+1}(\mathbf{a}) &= \{\mathbb{N}_t(\mathbf{a}) + 1\}^{-1} \left\{ \sum_{i=1}^t \mathbb{I}(\mathbf{A}_i = \mathbf{a}) \mathbf{R}_i + \mathbf{R}_{t+1} \right\} \\ &= \frac{\mathbb{N}_t(\mathbf{a})}{\mathbb{N}_t(\mathbf{a}) + 1} \left\{ \mathbb{N}_t^{-1}(\mathbf{a}) \sum_{i=1}^t \mathbb{I}(\mathbf{A}_i = \mathbf{a}) \mathbf{R}_i \right\} + \frac{\mathbf{R}_{t+1}}{\mathbb{N}_t(\mathbf{a}) + 1} \\ &= \frac{\mathbb{N}_t(\mathbf{a})}{\mathbb{N}_t(\mathbf{a}) + 1} \hat{Q}_t(\mathbf{a}) + \frac{\mathbf{R}_{t+1}}{\mathbb{N}_t(\mathbf{a}) + 1}.\end{aligned}$$

Algorithm

- **Input:** $0 < \varepsilon < 1$, termination time T .
- **Initialization:** $t = 0$, $\hat{Q}(\mathbf{a}) = \mathbf{0}$, $\mathbb{N}(\mathbf{a}) = \mathbf{0}$, for $\mathbf{a} = 1, 2, \dots, k$.
- **While** $t < T$:
 - **Update** t : $t \leftarrow t + 1$.
 - ε -greedy action selection:

$$\mathbf{a}^* \leftarrow \begin{cases} \arg \max_{\mathbf{a}} \hat{Q}(\mathbf{a}), & \text{with probability } 1 - \varepsilon, \\ \text{random arm,} & \text{with probability } \varepsilon. \end{cases}$$

- **Receive reward** R from arm \mathbf{a}^* .
- **Update** $\mathbb{N}(\mathbf{a}^*)$: $\mathbb{N}(\mathbf{a}^*) \leftarrow \mathbb{N}(\mathbf{a}^*) + 1$.
- **Update** $\hat{Q}(\mathbf{a}^*)$:

$$\hat{Q}(\mathbf{a}^*) \leftarrow \frac{\mathbb{N}(\mathbf{a}^*) - 1}{\mathbb{N}(\mathbf{a}^*)} \hat{Q}(\mathbf{a}^*) + \frac{1}{\mathbb{N}(\mathbf{a}^*)} R.$$

Example: Four Bernoulli Arms



Reward
distributions

Bernoulli(0.1)

Bernoulli(**0.4**)

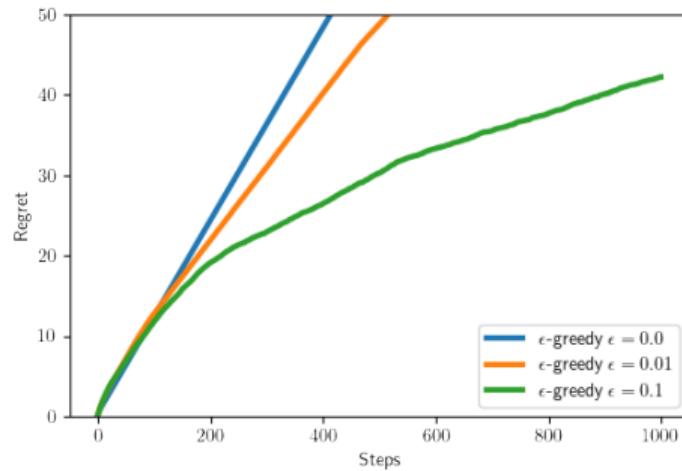
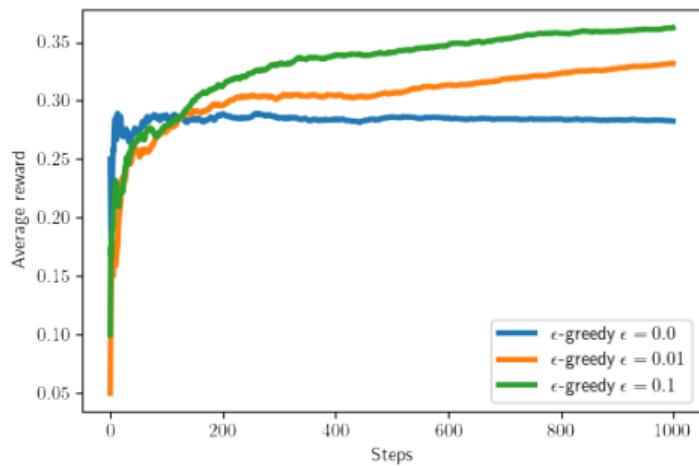
Bernoulli(0.1)

Bernoulli(0.1)



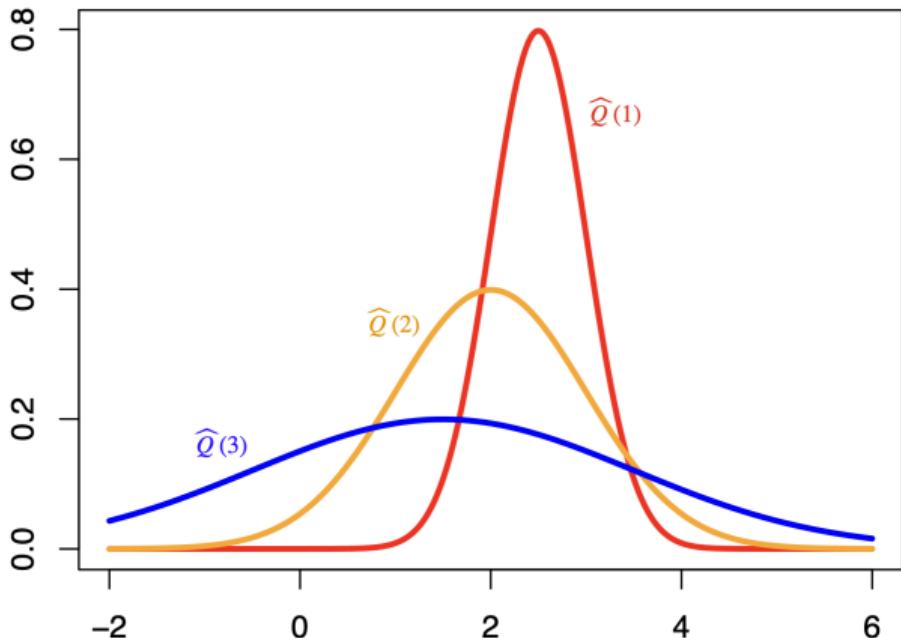
Best arm

Example: Four Bernoulli Arms (Cont'd)



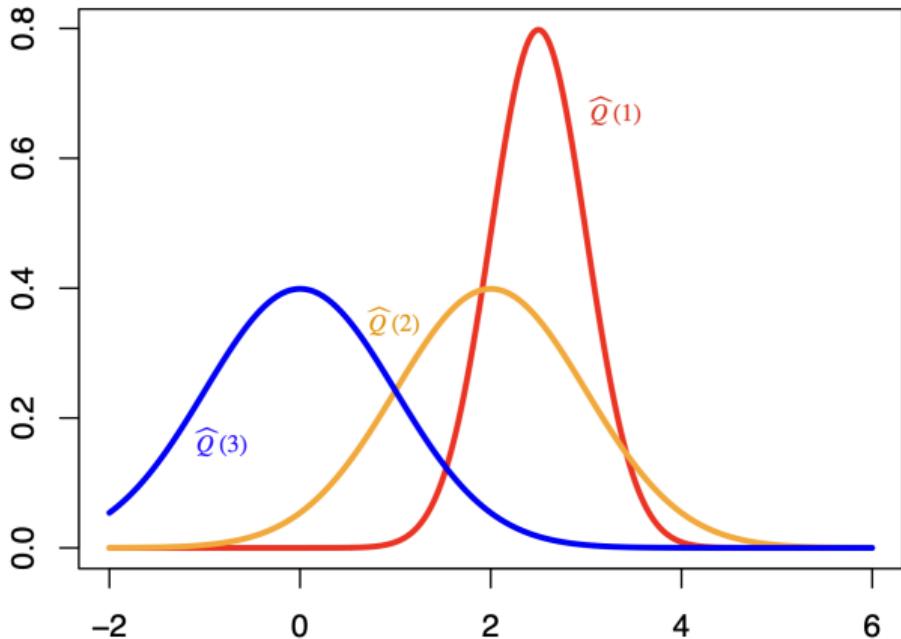
Optimism in the Face of Uncertainty

- The **optimistic principle**:
- The more **uncertain** we are about an action-value;
- The more **important** it is to explore that action;
- It could be the **best** action.
- Likely to pick blue action.
- **Different** from ϵ -greedy which selects arms uniformly random.



Optimism in the Face of Uncertainty (Cont'd)

- After picking blue action;
- Become less **uncertain** about the value;
- More likely to pick other actions;
- Until we home in on best action.



Upper Confidence Bound

- Estimate an **upper confidence** $U_t(a)$ for each action value such that

$$Q(a) \leq \hat{Q}_t(a) + U_t(a),$$

with high probability.

- $U_t(a)$ quantifies the **uncertainty** and depends on $N_t(a)$ (number of times arm a has been selected up to time t)
 - Large $N_t(a) \rightarrow$ small $U_t(a)$;
 - Small $N_t(a) \rightarrow$ large $U_t(a)$.
- Select actions maximizing upper confidence bound

$$a^* = \arg \max_a [\hat{Q}_t(a) + U_t(a)].$$

- Combines **exploration** ($U_t(a)$) and **exploitation** ($\hat{Q}_t(a)$).

Upper Confidence Bound (Cont'd)

- Set $U_t(a) = \sqrt{c \log(t)/N_t(a)}$ for some positive constant c .
- According to **Hoeffding's inequality** ([link](#)), when rewards are bounded between **0** and **1**, the event

$$Q(a) \leq \hat{Q}_t(a) + U_t(a),$$

holds with probability at least $1 - t^{-2c}$ (converges to 1 as $t \rightarrow \infty$).

Algorithm

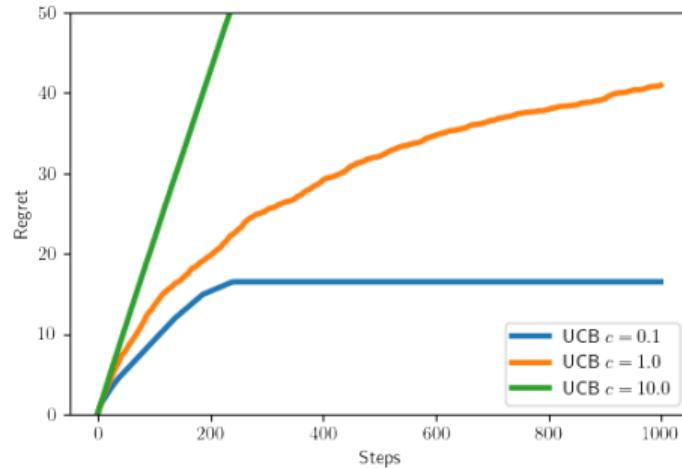
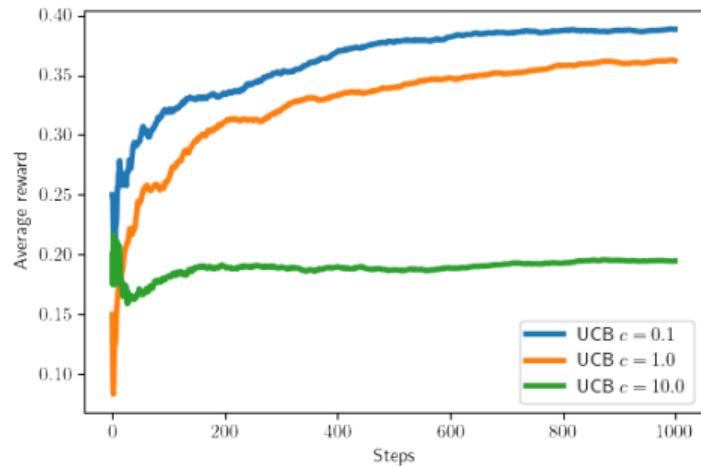
- **Input:** some positive constant c , termination time T .
- **Initialization:** $t = 0$, $\hat{Q}(\mathbf{a}) = \mathbf{0}$, $\mathbb{N}(\mathbf{a}) = \mathbf{0}$, for $a = 1, 2, \dots, k$.
- **While** $t < T$:
 - **Update** t : $t \leftarrow t + 1$.
 - **UCB action selection:**

$$\mathbf{a}^* \leftarrow \arg \max_{\mathbf{a}} [\hat{Q}(\mathbf{a}) + \sqrt{c \log(t) / \mathbb{N}_t(\mathbf{a})}].$$

- **Receive reward** R from arm \mathbf{a}^* .
- **Update** $\mathbb{N}(\mathbf{a}^*)$: $\mathbb{N}(\mathbf{a}^*) \leftarrow \mathbb{N}(\mathbf{a}^*) + 1$.
- **Update** $\hat{Q}(\mathbf{a}^*)$:

$$\hat{Q}(\mathbf{a}^*) \leftarrow \frac{\mathbb{N}(\mathbf{a}^*) - 1}{\mathbb{N}(\mathbf{a}^*)} \hat{Q}(\mathbf{a}^*) + \frac{1}{\mathbb{N}(\mathbf{a}^*)} R.$$

Example: Four Bernoulli Arms (Revisited)



Thompson Sampling

- A **highly-competitive** algorithm to address exploration-exploitation trade-off.
- Impose **statistical models** for the reward distribution with parameter θ .
- Impose **prior distributions** for θ .
- At time t ,
 - Use **Bayes rule** to update the **posterior distribution** of θ .
 - Sample a model parameter θ_t from the posterior distribution.
 - Compute action-value given θ_t , i.e., $\mathbb{E}(R|A = a, \theta_t)$.
 - Select action maximizing action-value

$$a^* = \arg \max_a \mathbb{E}(R|A = a, \theta_t).$$

- Posterior distribution quantifies the **uncertainty** of the estimated model parameter (**exploration**).
- $\mathbb{E}(R|A = a, \theta_t)$ estimates the oracle action value (**exploitation**).

Thompson Sampling (Bernoulli Bandit Example)

- **Statistical models:**
 - Reward of the a th arm follows a Bernoulli distribution with mean $\theta(a)$.
 - $\theta(a)$ follows a Beta(α, β) distribution (**prior**).
 - **Conjugate** distribution of binomial, i.e. posterior distribution is Beta as well
 - α and β measures the beliefs for **success** and **failure**
- **Bayesian inference:**
 - $\theta(a)$ follows a Beta($S_a + \alpha, F_a + \beta$) distribution (**posterior**) where (S_a, F_a) corresponds to the success and failure counters under arm a .
- **Compute action value:**

$$\mathbb{E}(R|A=a, \theta_t) = \theta_t(a).$$

Algorithm (Bernoulli Bandit Example¹)

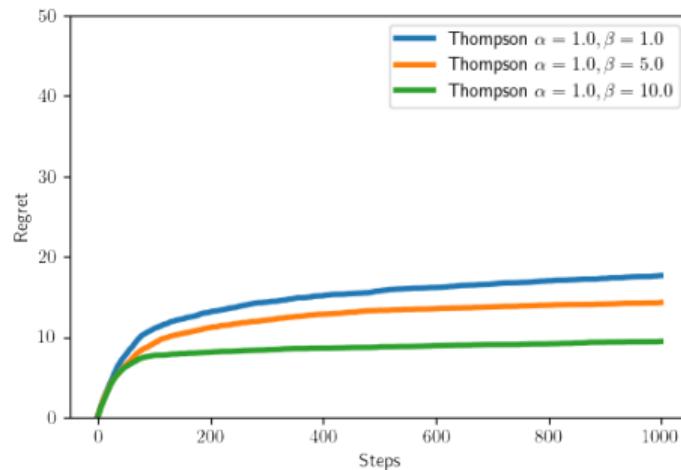
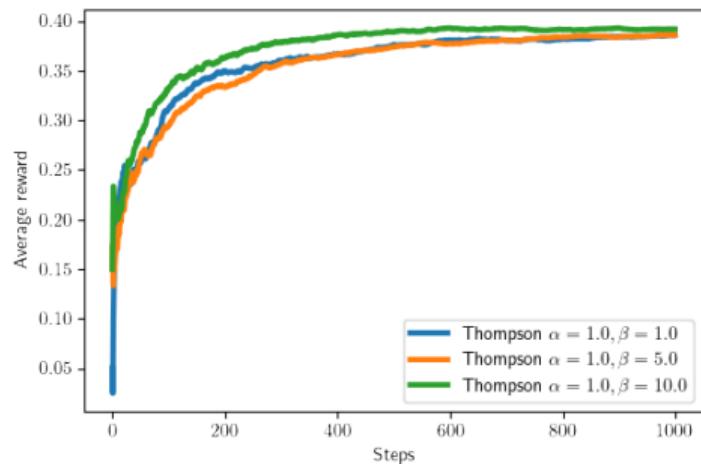
- **Input:** hyper-parameters $\alpha, \beta > 0$, termination time T .
- **Initialization:** $t = 0$, $S_a = F_a = 0$, for $a = 1, 2, \dots, k$.
- **While** $t < T$:
 - **Update** t : $t \leftarrow t + 1$.
 - **Posterior sampling:** For $a = 1, 2, \dots, k$, sample

$$\theta_a \sim \text{Beta}(S_a + \alpha, F_a + \beta)$$

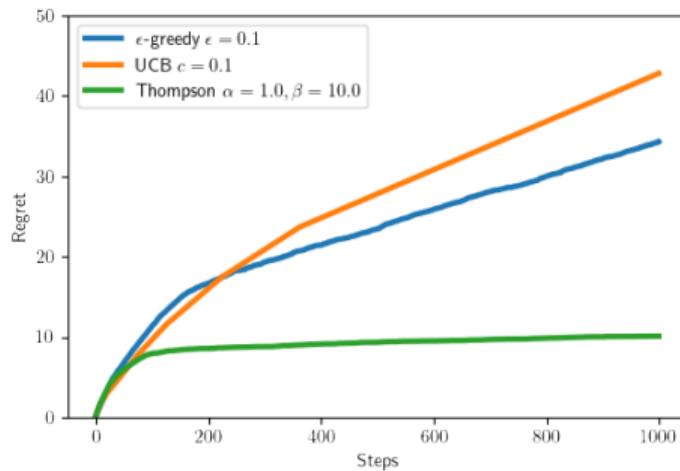
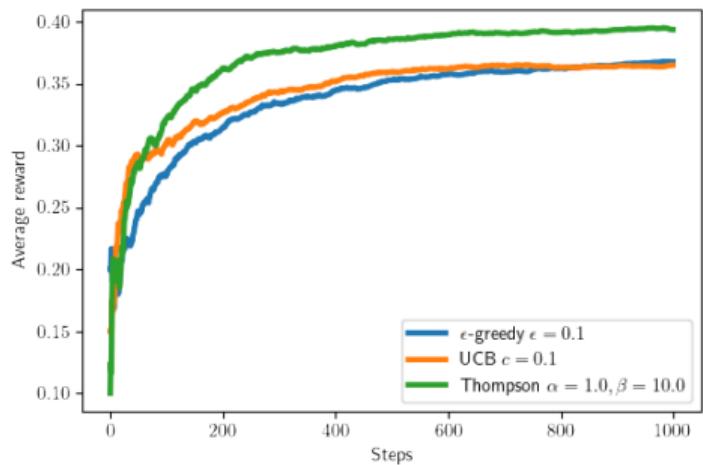
- **Action selection:** $a^* \leftarrow \arg \max_a \theta_a$.
- **Receive reward** R from arm a^* .
- **Update** S_a and F_a :
 - If $R = 1$, $S_a \leftarrow S_a + 1$;
 - If $R = 0$, $F_a \leftarrow F_a + 1$.

¹The general algorithm can be found in Chapelle and Li [2011]

Example: Four Bernoulli Arms (Revisited)



Example: Four Bernoulli Arms (Cont'd)



Theory

Define the **regret** $\mathcal{R}(\mathbf{T})$ as the difference between the cumulative reward under the **best action** and that under the **selected actions**, up to time \mathbf{T} .

Theorem (UCB, Auer et al. [2002])

The expected regret of the UCB algorithm $\mathbb{E}\mathcal{R}(\mathbf{T})$ is upper bounded by $C_1 \log(\mathbf{T})$ for some constant $C_1 > 0$.

Theorem (TS, Agrawal and Goyal [2012])

The expected regret of the Thompson sampling algorithm $\mathbb{E}\mathcal{R}(\mathbf{T})$ is upper bounded by $C_2 \log(\mathbf{T})$ for some constant $C_2 > 0$.

- Both algorithms achieve logarithmic expected regret.
- Their performances are nearly the same as the oracle method that works as if the best action were known.

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

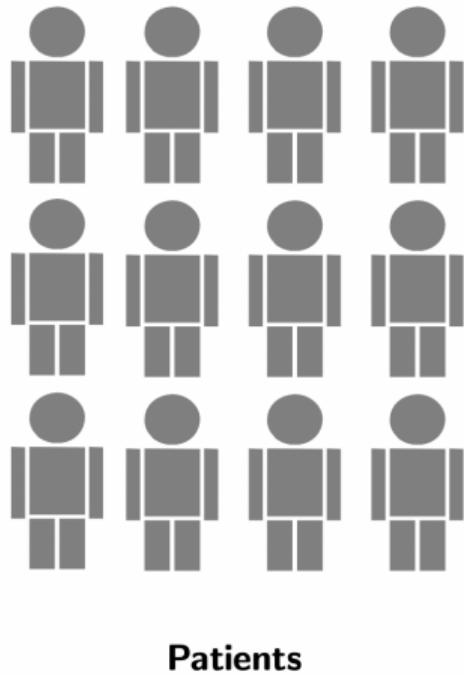
- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Contextual Bandits

- Extension of MAB with **contextual** information.
- A **widely-used** model in medicine and technological industries.
- At time t , the agent
 - Observe a context S_t ;
 - Select an action A_t ;
 - Receives a reward R_t (depends on both S_t and A_t).
- **Objective**: maximize cumulative reward.
- **ϵ -greedy, UCB and Thompson sampling** can be similarly adopted [see e.g., Chu et al., 2011, Agrawal and Goyal, 2013, Zhou et al., 2020, Zhang et al., 2020].

Application I: Precision Medicine



Treatment A

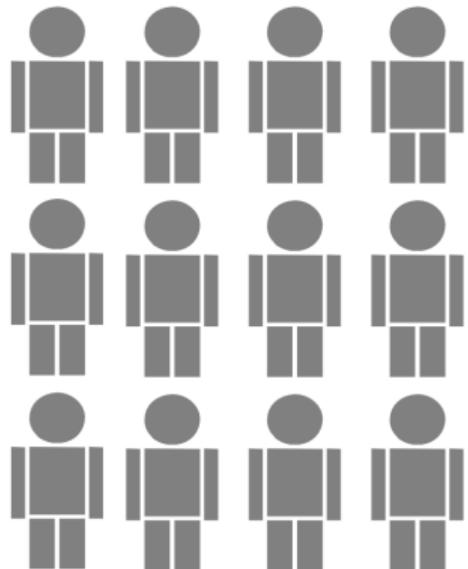


Treatment B



Treatment C

One-Size-Fits-All

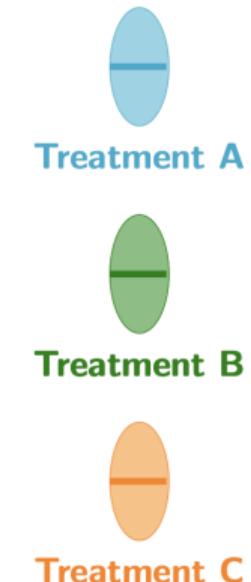
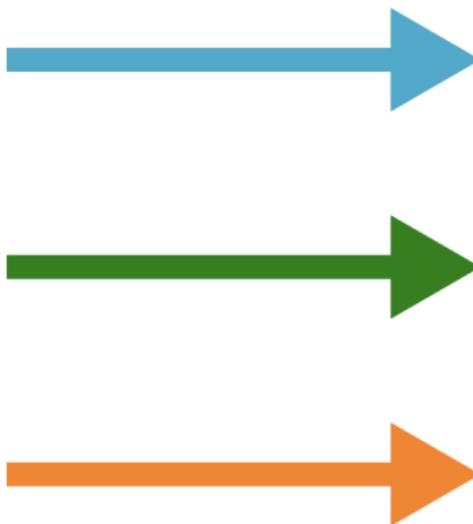


Patients

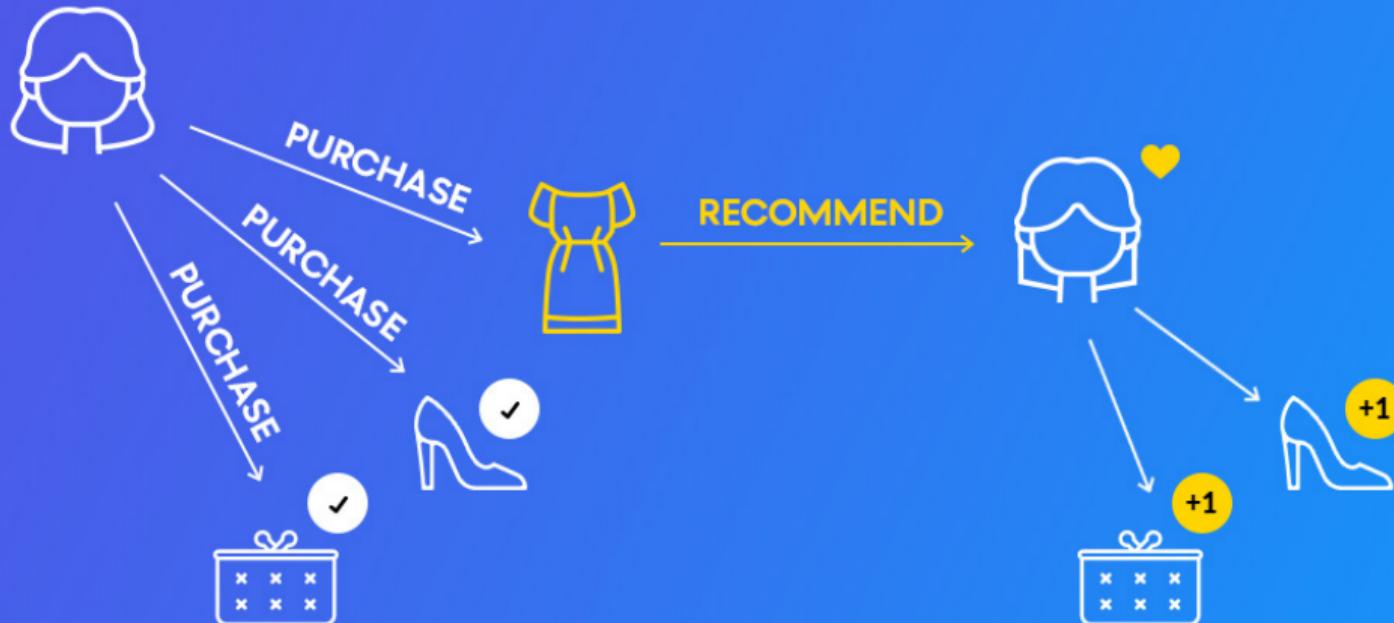


Treatment B

Individualized Treatment Regime



Application II: Personalized Recommendation



Lecture Outline

1. Introduction to Reinforcement Learning (RL)

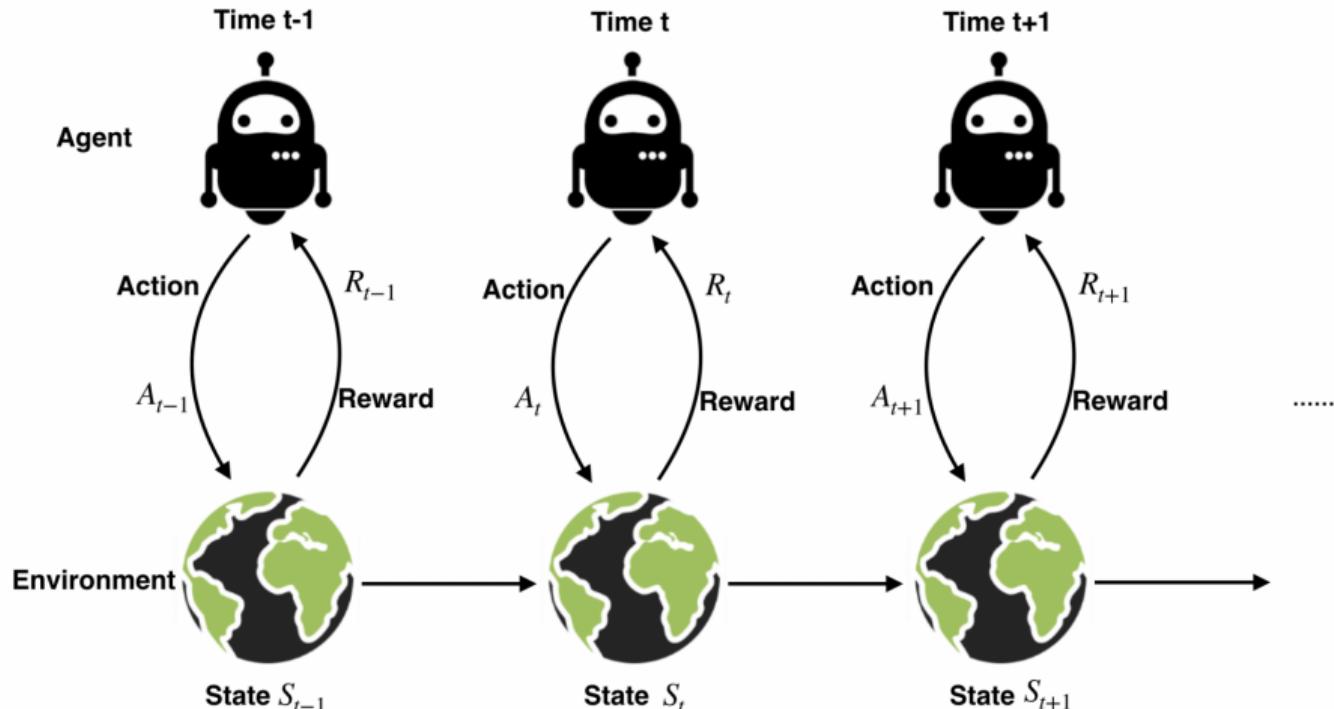
- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Sequential Decision Making



Objective: find an optimal policy that maximizes the cumulative reward

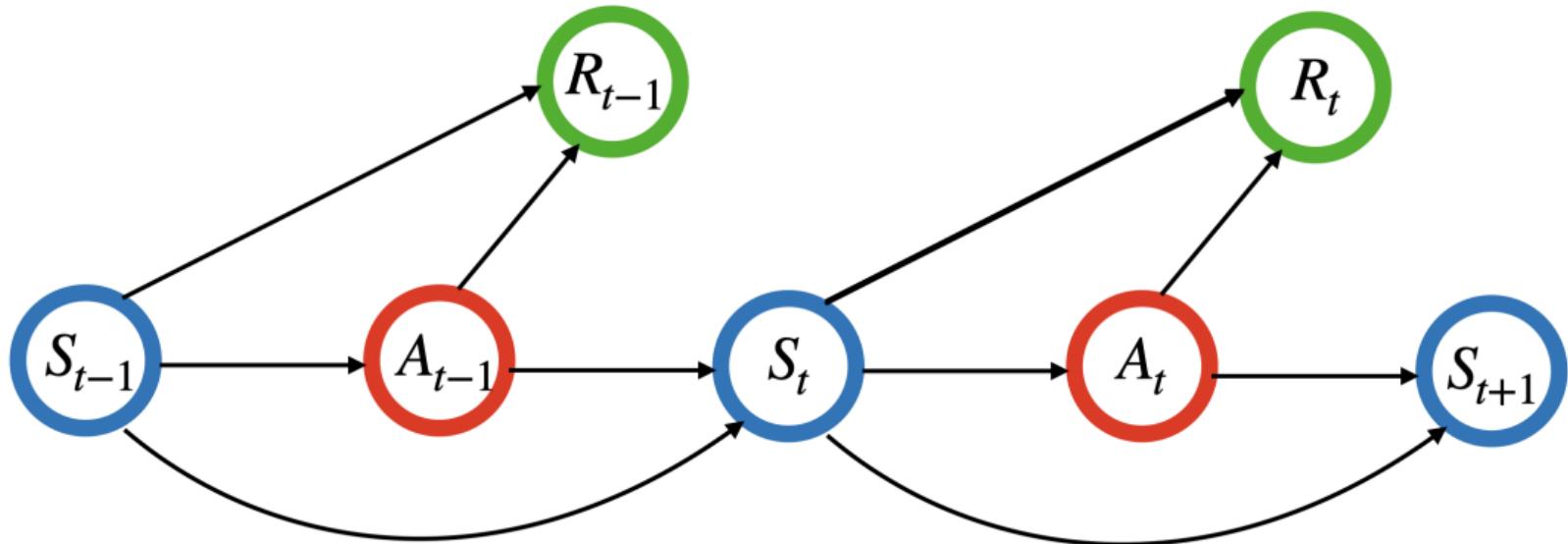
Markov Decision Processes

Definition

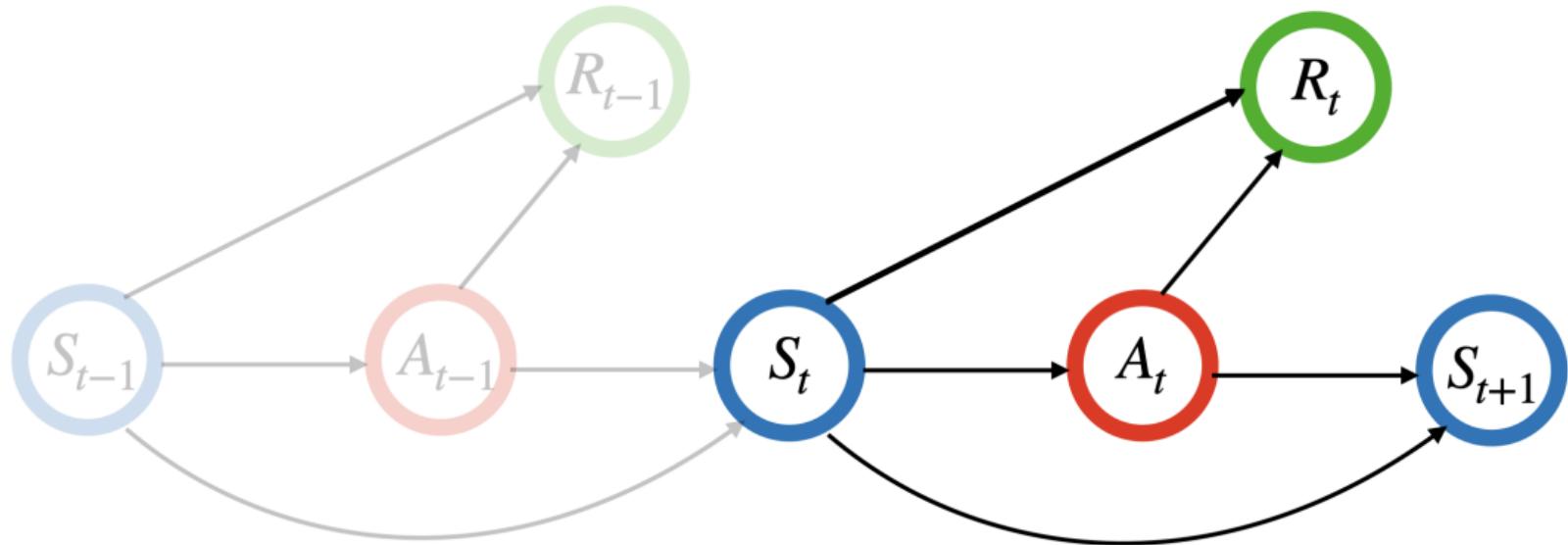
$\{S_t, A_t, R_t\}_t$ forms a Markov decision process if and only if

- $\Pr(S_{t+1}, R_t | A_t, S_t) = \Pr(S_{t+1}, R_t | A_t, S_t, R_{t-1}, A_{t-1}, S_{t-1}, \dots)$ (Markovianity)
- $\Pr(S_{t+1}, R_t | A_t = a, S_t = s) = \Pr(S_t, R_{t-1} | A_{t-1} = a, S_{t-1} = s)$
(time-homogeneity)
- The current **state-action** pair captures all relevant information from the history
- When A_t depends the history only through S_t , $\{S_t, A_t, R_t\}_t$ forms a Markov chain.

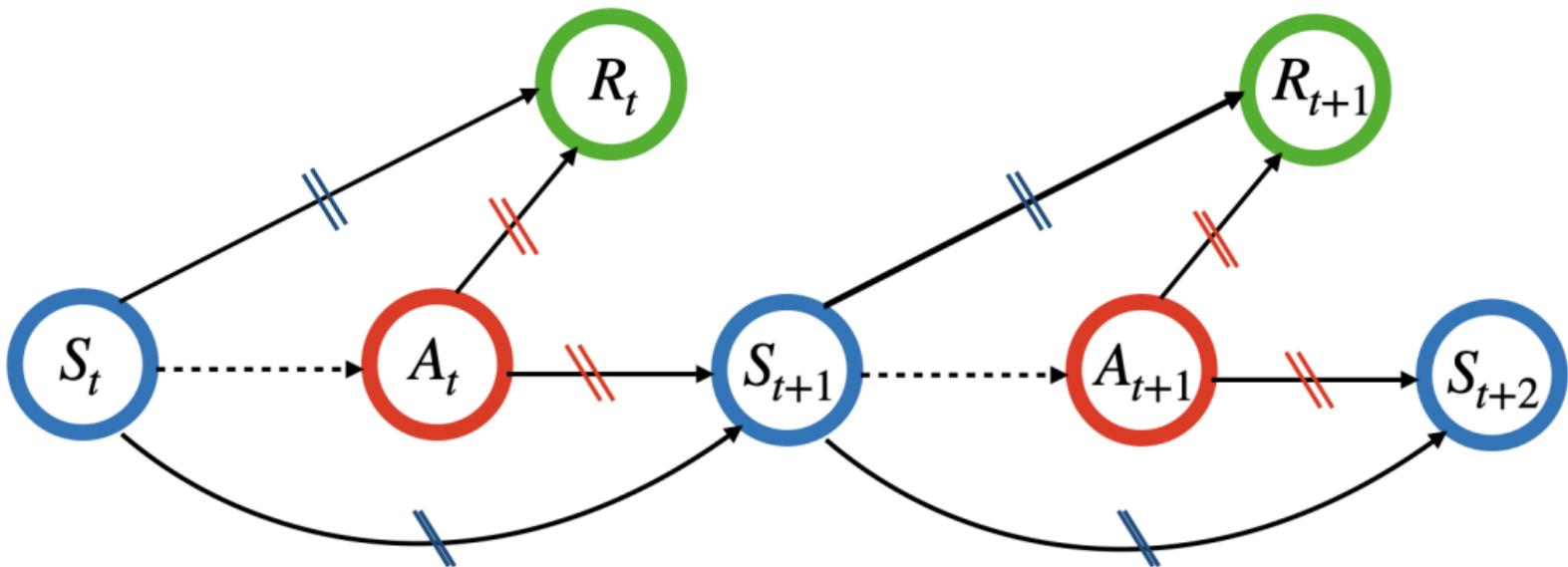
Markov Assumption



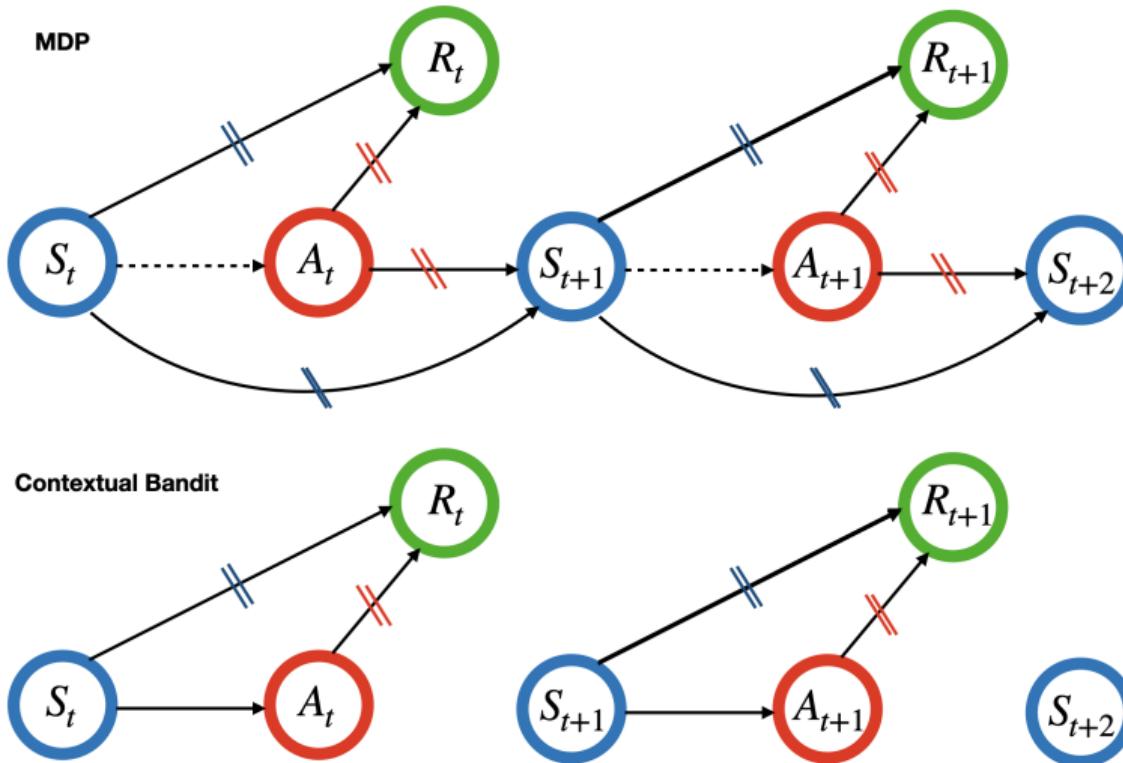
Markov Assumption



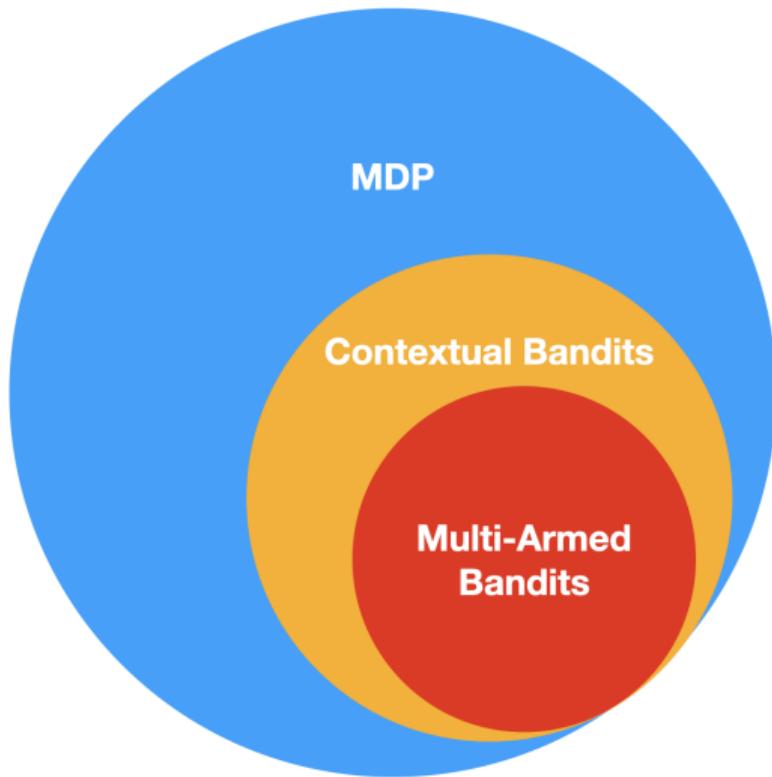
Stationarity Assumption



MDP vs Contextual Bandits



MDP v.s. Contextual Bandits (Cont'd)



Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Time-Varying MDPs

- The **time-homogeneity** assumption is likely to be violated in real applications (e.g., mobile health, ridesharing)
- **Nonstationarity** is the case most commonly encountered in reinforcement learning [Sutton and Barto, 2018]

Definition

$\{S_t, A_t, R_t\}_t$ forms a time-varying Markov decision process iff

$$\Pr(S_{t+1}, R_t | A_t, S_t) = \Pr(S_{t+1}, R_t | A_t, S_t, R_{t-1}, A_{t-1}, S_{t-1}, \dots) \quad (\text{Markovianity})$$

Causal Diagram: TMDP

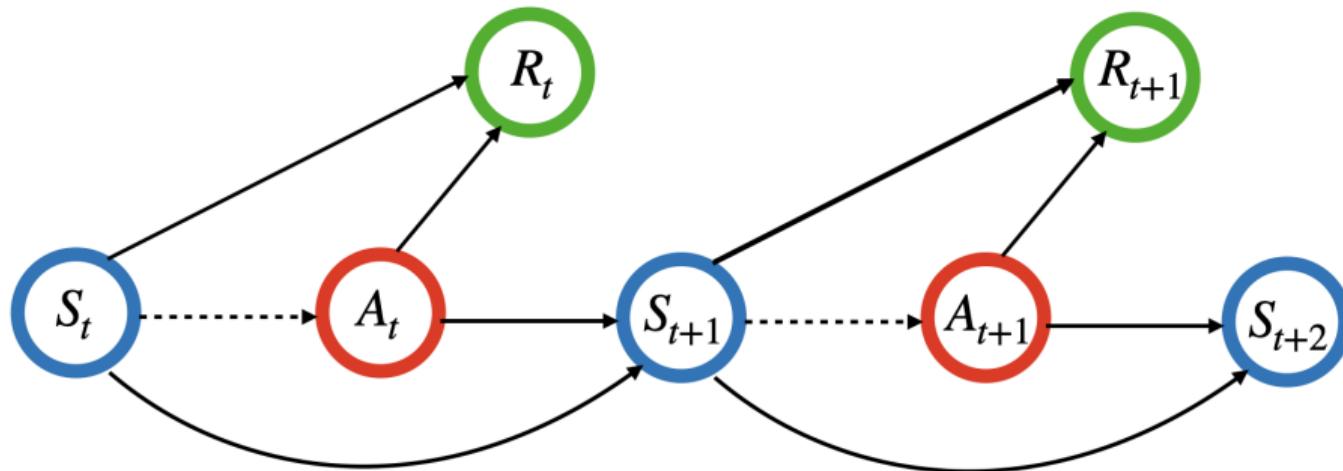


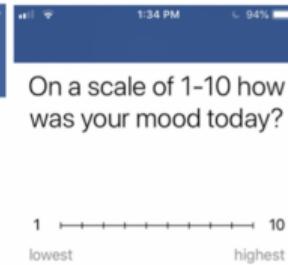
Figure: Causal diagrams for MDPs. Solid lines represent causal relationships. The parent nodes for the action is **not** specified in the model. A_t could either depend on S_t or the history.

Mobile Health Example: Intern Health Study

- Mental health management
- Subject: First-year medical interns
- S_t : Interns' mood scores, sleep hours and step counts
- A_t : Send text notifications or not
- R_t : Mood scores or step counts

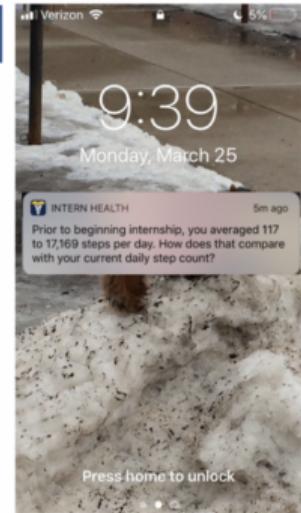


(i) App Dashboard



Done
Cancel

(ii) Mood EMA



Press home to unlock
(iii) Notifications

- The study lasts for half an year
- Treatment effects are usually **time-inhomogeneous** (decays over time)
- Leading to TMDPs

Ridesharing Example: Order-Dispatching



- S_t : Supply (available drivers) and demand (call orders)
- A_t : Order-dispatching: match a driver with an order
- R_t : Passengers' answer rate/Drivers' income
- Weekday-weekend differences, peak and off-peak differences lead to time-inhomogeneity

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

Partially Observable MDPs

- Difference between MDPs and POMDPs: states **fully-observable** or **partially-observable**
- The fully-observability assumption might be violated in practice
- In healthcare, patients' characteristics might not be fully recorded

Causal Diagram: POMDP

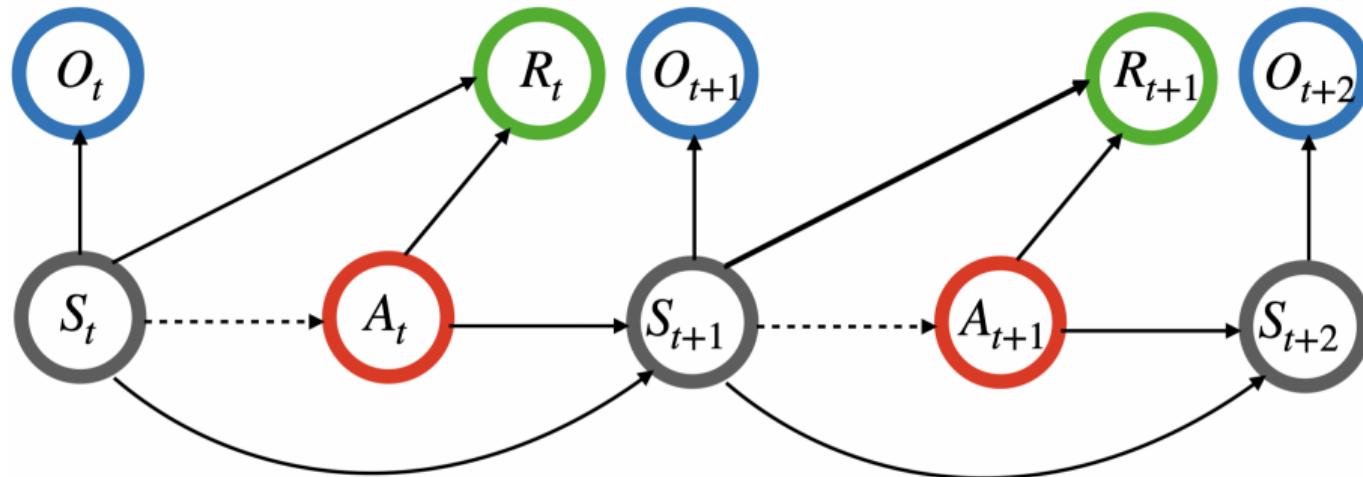
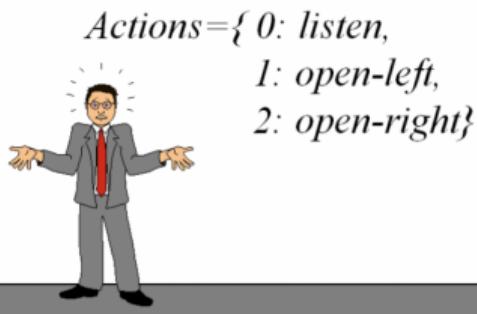
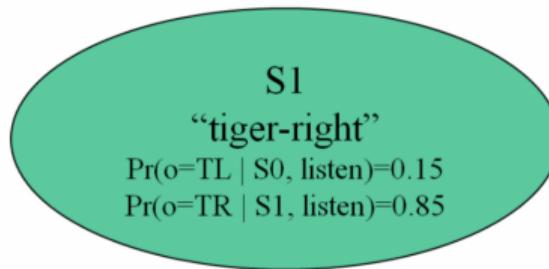
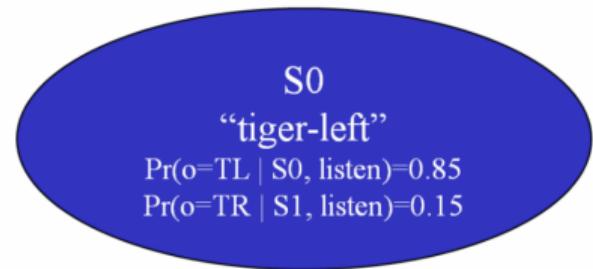


Figure: Causal diagrams for POMDPs. Solid lines represent causal relationships. $\{S_t\}_t$ denotes latent states. The parent nodes for the action is **not** specified in the model. A_t could either depend on O_t or the history.

Example: the Tiger Problem



Reward Function

- Penalty for wrong opening: -100
- Reward for correct opening: +10
- Cost for listening action: -1

Observations

- to hear the tiger on the left (TL)
- to hear the tiger on the right (TR)

Example: the Tiger Problem (Cont'd)

Suppose we choose to listen at each time

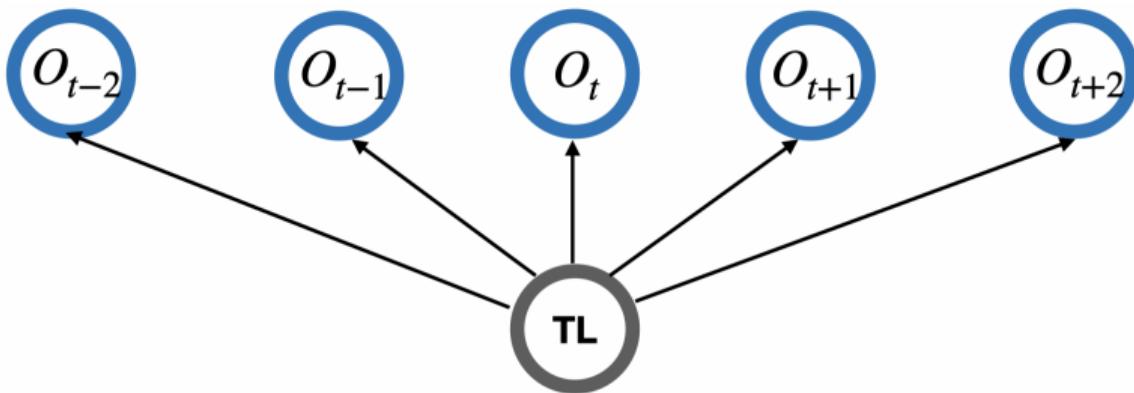


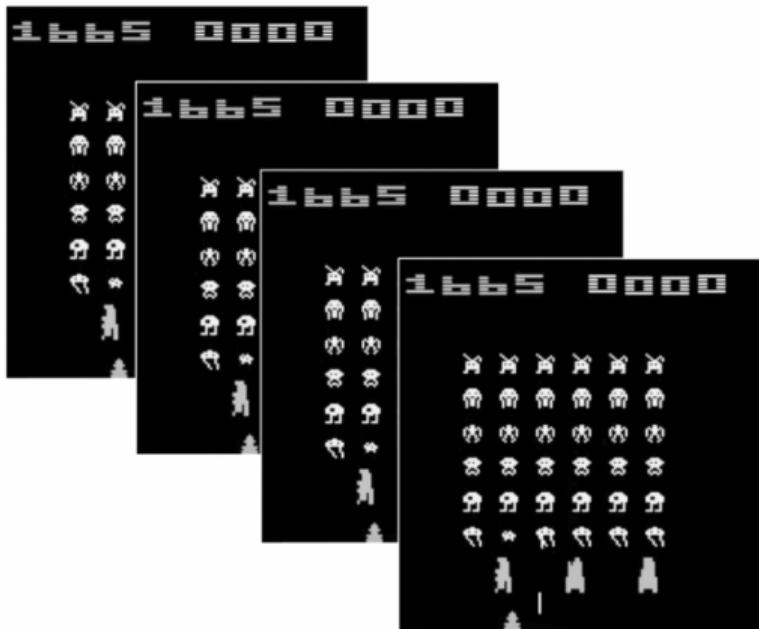
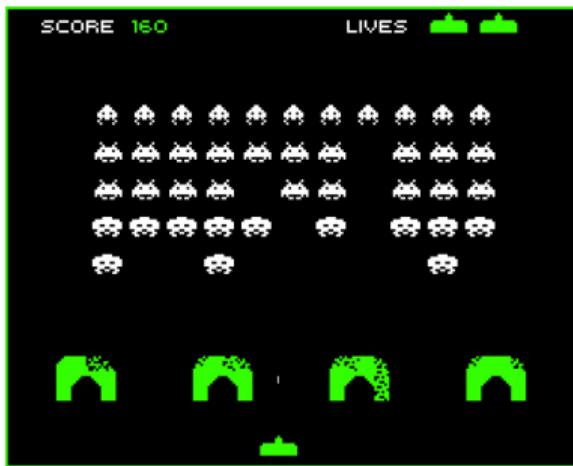
Figure: Causal diagram for the tiger problem. TL denotes the tiger location. O_t denotes the inferred location of the tiger at time t .

Converting non-MDPs into MDPs

- MDP assumptions: Markovianity & time-homogeneity
- To ensure **time-homogeneity**: include time variables in the state
- In ridesharing, include dummy variables weekdays/weekends & peak/off-peak hours
- In mobile health, use more recent observations
- To ensure **Markovianity**: concatenate measurements over multiple time steps

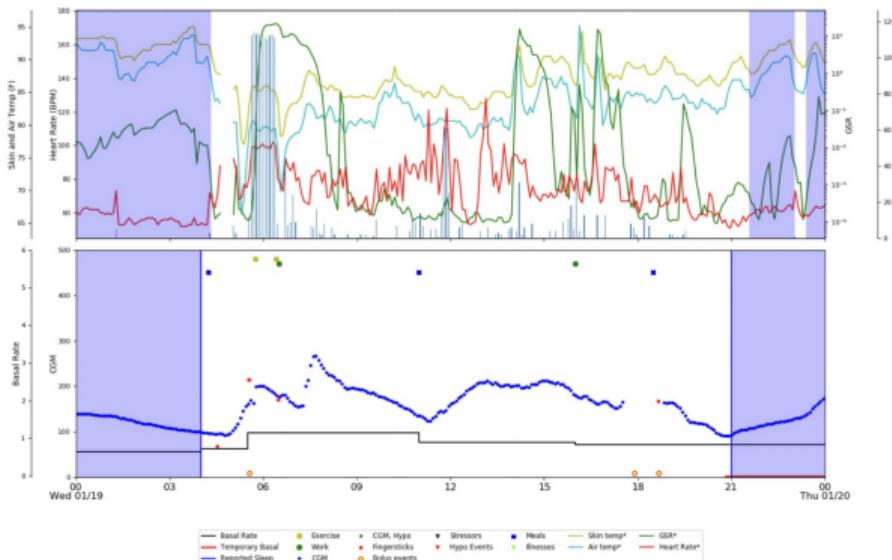
Stacking Frames in Atari Games

Input is a stack of 4 most recent frames [Mnih et al., 2015]



Concatenating Observations in Diabetes Study

- Management of **Type-I diabetes**
- **Subject:** Patients with diabetes.
- S_t : Patient's **glucose levels, food intake, exercise intensity**
- A_t : **Insulin doses injected**
- R_t : **Index of Glycemic Control**
(function of patient's glucose level)



- Markovianity holds when concatenating 4 most recent observations [Shi et al., 2020]
- Concatenating observations also yield better policies

Lecture Outline

1. Introduction to Reinforcement Learning (RL)

- 1.1 Multi-Armed Bandits
- 1.2 Contextual Bandits

2. Markov Decision Processes (MDPs)

- 2.1 Time-Varying MDPs (TMDPs)
- 2.2 Partially Observable MDPs (POMDPs)

3. The Existence of the Optimal Stationary Policy

The Agent's Policy

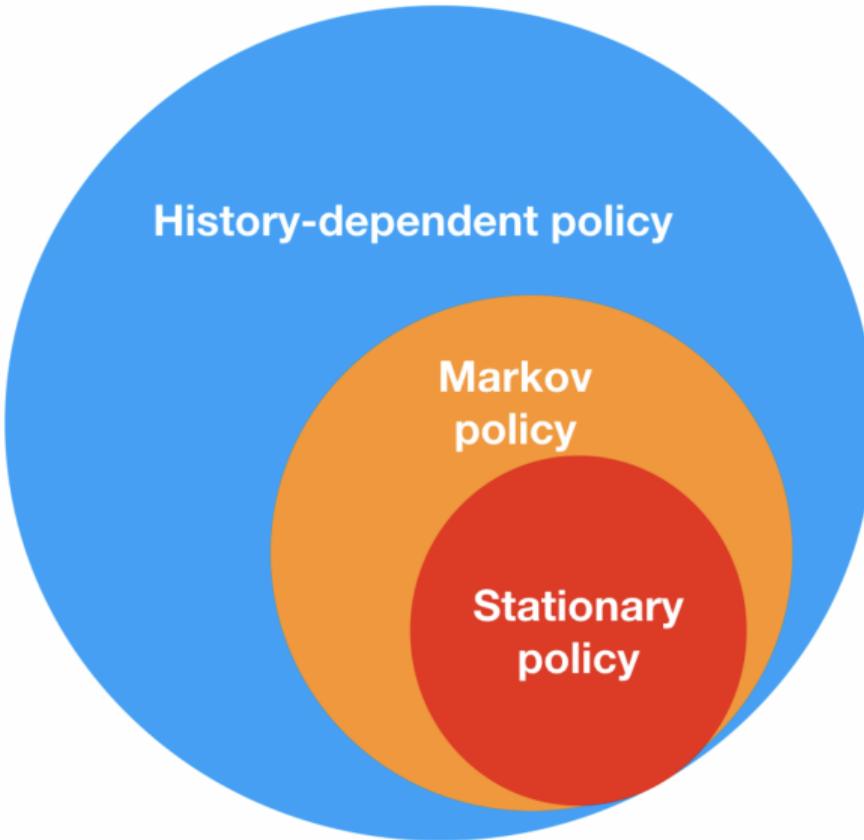
- The agent implements a **mapping** π_t from the observed data to a probability distribution over actions at each time step
- The collection of these mappings $\pi = \{\pi_t\}_t$ is called **the agent's policy**:

$$\pi_t(a|\bar{s}) = \Pr(A_t = a | \bar{S}_t = \bar{s}),$$

where $\bar{S}_t = (\mathcal{S}_t, \mathcal{R}_{t-1}, \mathcal{A}_{t-1}, \mathcal{S}_{t-1}, \dots, \mathcal{R}_0, \mathcal{A}_0, \mathcal{S}_0)$ is the set of **observed data history** up to time t .

- **History-Dependent Policy:** π_t depends on \bar{S}_t .
- **Markov Policy:** π_t depends on \bar{S}_t only through S_t .
- **Stationary Policy:** π is Markov & π_t is **homogeneous** in t , i.e., $\pi_0 = \pi_1 = \dots$.

The Agent's Policy (Cont'd)



The Agent's Policy (Cont'd)

- The collection of these mappings $\pi = \{\pi_t\}_t$ is called **the agent's policy**:

$$\pi_t(a|\bar{s}) = \Pr(A_t = a | \bar{S}_t = \bar{s}),$$

where $\bar{S}_t = (\mathcal{S}_t, \mathcal{R}_{t-1}, \mathcal{A}_{t-1}, \mathcal{S}_{t-1}, \dots, \mathcal{R}_0, \mathcal{A}_0, \mathcal{S}_0)$.

- **Random Policy:** $\pi_t(\bullet|\bar{s})$ is a probability distribution over the action space
- **Deterministic Policy:** each probability distribution is degenerate
 - i.e., for any t and \bar{s} , $\pi_t(a|\bar{s}) = 1$ for some a and 0 for other actions
 - use $\pi_t(\bar{s})$ to denote the action that the agent selects

Goals, Objectives and the Return

The agent's goal: find a policy that maximizes the **expected return** received in long run

Definition (Return, Average Reward Setting)

The **return** G_t is the average reward from time-step t .

$$G_t = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i=t}^{t+T-1} R_i.$$

Definition (Return, Discounted Reward Setting)

The **return** G_t is the cumulative discounted reward from time-step t .

$$G_t = \sum_{i=0}^{+\infty} \gamma^i R_{i+t}.$$

Discounted Reward Setting (Our Focus)

Definition (Return)

The **return** G_t is the cumulative discounted reward from time-step t .

$$G_t = \sum_{i=0}^{+\infty} \gamma^i R_{i+t}$$

- The **discount factor** $0 \leq \gamma < 1$ represents the **trade-off** between **immediate** and **future** rewards.
- The value of receiving reward R after k time steps is $\gamma^k R$.
- $\gamma = 0$ leads to “**myopic**” evaluation
- γ close to 1 leads to “**far-sighted**” evaluation (close to the average reward)

Why Discount?

- **Mathematically convenient:** avoids infinite returns.
- **Computationally convenient:** easier to develop practical algorithms.
- In finance, immediate rewards earn more **interests** than delayed rewards
- Animal/human behaviour shows **preference** for immediate reward
 - Go to bed late and you'll be tired tomorrow
 - Eat heartily in winter and you'll need to trim fat in summer
- Possible to set $\gamma = 1$ in **finite horizon** settings (number of decision steps is finite; e.g., precision medicine applications where patients receive only a finite number of treatments)

(State) Value Function

Definition

The (state) value function $V^\pi(s)$ is expected return starting from s under π ,

$$V^\pi(s) = \mathbb{E}^\pi(G_t | S_t = s) = \mathbb{E}^\pi\left(\sum_{i=0}^{+\infty} \gamma^i R_{i+t} | S_t = s\right).$$

- V^π is **independent** of the time t in its definition, under **time-homogeneity**
- \mathbb{E}^π denotes the expectation assuming the system follows π

Bellman Equation

Definition

The Bellman equation for the state value function is given by

$$V^\pi(s) = \mathbb{E}^\pi\{R_t + \gamma V^\pi(S_{t+1}) | S_t = s\}.$$

- The value function can be **decomposed** into two parts:
 - Immediate reward R
 - discounted value of success state $\gamma V^\pi(S_{t+1})$
- Forms the basis for **value evaluation** (more in later lectures)

Bellman Equation (Proof)

$$\begin{aligned}V^\pi(s) &= \mathbb{E}^\pi(G_t | S_t = s) \\&= \mathbb{E}^\pi(R_t + \gamma(R_{t+1} + \gamma R_{t+2} + \dots) | S_t = s) \\&= \mathbb{E}^\pi(R_t | S_t = s) + \gamma \mathbb{E}^\pi(G_{t+1} | S_t = s) \\&= \mathbb{E}^\pi(R_t | S_t = s) + \gamma \mathbb{E}^\pi\{\mathbb{E}^\pi(G_{t+1} | S_{t+1}, S_t) | S_t = s\} \\&= \mathbb{E}^\pi(R_t | S_t = s) + \gamma \mathbb{E}^\pi\{\mathbb{E}^\pi(G_{t+1} | S_{t+1}) | S_t = s\} \\&= \mathbb{E}^\pi(R_t | S_t = s) + \gamma \mathbb{E}^\pi\{V^\pi(S_{t+1}) | S_t = s\},\end{aligned}$$

The second last equation holds due to the **Markov assumption**.

Bellman Optimality Equation

Definition

The Bellman optimality equation for the state-value function is given by

$$V^{\pi^{\text{opt}}}(s) = \max_a \mathbb{E}\{R_t + \gamma V^{\pi^{\text{opt}}}(S_{t+1}) | A_t = a, S_t = s\}.$$

- According to the Bellman equation,

$$V^{\pi^{\text{opt}}}(s) = \mathbb{E}^{\pi^{\text{opt}}}\{R_t + \gamma V^{\pi^{\text{opt}}}(S_{t+1}) | S_t = s\}.$$

- The optimal policy selects the action that maximizes the value: $\mathbb{E}^{\pi^{\text{opt}}} = \max_a \mathbb{E}$

Existence of Optimal Stationary Policy in MDPs

Theorem (See also Puterman [2014], Theorem 6.2.10)

Assume the state-action space is **discrete** and the rewards are **bounded**. Then there exists an **optimal stationary policy** $\pi^{opt} = \{\pi_t^{opt}\}_t$ such that

- $\pi_1^{opt} = \pi_2^{opt} = \dots = \pi_t^{opt} = \dots$
- $\mathbb{E}^{\pi^{opt}} G_0 \geq \mathbb{E}^\pi G_0$ for any **history-dependent policy** π

- When the system dynamics satisfies the **Markov** and **time-homogeneity** assumption, so does the **optimal policy**.
- Lay the **foundation** for most existing RL algorithms
- Simplify the calculation since it suffices to focus on stationary policies

Sketch of the Proof [Shi et al., 2020]

Goal HR, MR, SR denote classes of history-dependent, Markov and stationary policies. To show $\sup_{\pi \in \text{SR}} V^\pi(s) = \sup_{\pi \in \text{HR}} V^\pi(s)$ for any s .

Step 1 Show $\sup_{\pi \in \text{MR}} V^\pi(s) = \sup_{\pi \in \text{HR}} V^\pi(s)$ for any s under Markovianity.

Step 2 Show for any function ν that satisfies the **Bellman optimality equation**,

$$\nu(s) = \max_a [\mathbb{E}\{\mathcal{R}_t + \gamma \nu(\mathcal{S}_{t+1}) | \mathcal{A}_t = a, \mathcal{S}_t = s\}]$$

$$\nu(s) = \sup_{\pi \in \text{MR}} V^\pi(s) \text{ for any } s.$$

Step 3 Show the existence of $\pi^* \in \text{SR}$ such that V^{π^*} satisfies the Bellman optimality equation. This together with Step 2 yields

$$\sup_{\pi \in \text{SR}} V^\pi(s) = \sup_{\pi \in \text{MR}} V^\pi(s).$$

Sketch of the Proof (Step 1)

The key to Step 1 is to show for any $\pi \in \mathbf{HR}$ and any s , there exists a Markov policy $\dot{\pi} = \{\dot{\pi}_t\}_{t \geq 0}$ where $\dot{\pi}_t$ depends on S_t only such that

$$\Pr^\pi(A_t = a, S_t = s' | S_0 = s) = \Pr^{\dot{\pi}}(A_t = a, S_t = s' | S_0 = s), \quad (1)$$

for any $t \geq 0, a, s'$ where the probabilities \Pr^π and $\Pr^{\dot{\pi}}$ are taken by assuming the system dynamics follow π and $\dot{\pi}$, respectively.

Under the **Markov assumption**, we have

$$\mathbb{E}^\pi(R_t | S_0 = s) = \mathbb{E}^\pi[r(S_t, A_t) | S_0], \quad \forall t \geq 0.$$

This together with (1) yields that

$$\mathbb{E}^\pi(R_t | S_0 = s) = \mathbb{E}^{\dot{\pi}}(R_t | S_0 = s), \quad \forall t \geq 0,$$

and hence $V^\pi(s) = V^{\dot{\pi}}(s)$.

Sketch of the Proof (Step 2)

First, by iteratively apply the inequality

$$\nu(\mathbf{s}) \geq \max_{\mathbf{a}} \mathbb{E}[\mathbf{R}_t + \gamma \nu(\mathbf{S}_{t+1}) | \mathbf{A}_t = \mathbf{a}, \mathbf{S}_t = \mathbf{s}]$$

we can show that $\nu(\mathbf{s}) \geq \sup_{\pi \in \text{MR}} V^\pi(\mathbf{s})$ for any \mathbf{s}

Second, define the operator

$$\mathcal{L}\nu(\mathbf{s}) = \max_{\mathbf{a}} \mathbb{E}[\nu(\mathbf{S}_{t+1}) | \mathbf{A}_t = \mathbf{a}, \mathbf{S}_t = \mathbf{s}]$$

The operator $\mathcal{I} - \gamma \mathcal{L}$ is bounded and linear, and is thus invertible and its inverse equals $\sum_{k \geq 0} \gamma^k \mathcal{L}^k$. This together with

$$\nu(\mathbf{s}) \leq \max_{\mathbf{a}} \mathbb{E}[\mathbf{R}_t + \gamma \nu(\mathbf{S}_{t+1}) | \mathbf{A}_t = \mathbf{a}, \mathbf{S}_t = \mathbf{s}]$$

yields that $\nu(\mathbf{s}) \leq \sup_{\pi \in \text{MR}} V^\pi(\mathbf{s})$ for any \mathbf{s}

Sketch of the Proof (Step 3)

For any function ν , define the norm $\|\nu\|_\infty = \sup_s |\nu(s)|$. We have for any ν_1 and ν_2 that

$$\begin{aligned} & \sup_s \left| \max_a \mathbb{E}[R_t + \gamma \nu_1(S_{t+1}) | A_t = a, S_t = s] \right. \\ & \quad \left. - \max_a \mathbb{E}[R_t + \gamma \nu_2(S_{t+1}) | A_t = a, S_t = s] \right| \\ & \leq \gamma \max_a \sup_s |\mathbb{E}[\nu_1(S_{t+1}) - \nu_2(S_{t+1}) | A_t = a, S_t = s]| \\ & \qquad \qquad \qquad \leq \gamma \|\nu_1 - \nu_2\|_\infty \end{aligned}$$

By **Banach's fix point theorem**, there exists a unique value function ν_0 that satisfies the optimal Bellman equation. This together with the first two steps completes the proof.

Existence of Optimal Markov Policy in TMDPs

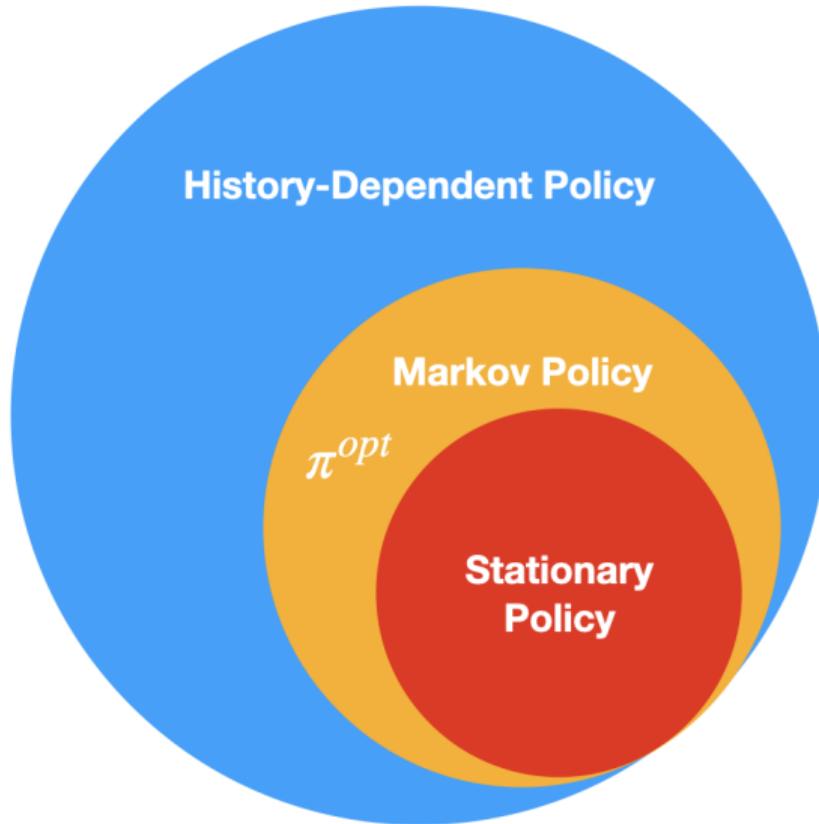
Theorem (See also Puterman [2014], Theorem 5.5.1)

Assume the state-action space is **discrete**. Then there exists an **optimal Markov policy** $\pi^{opt} = \{\pi_t^{opt}\}_t$ such that

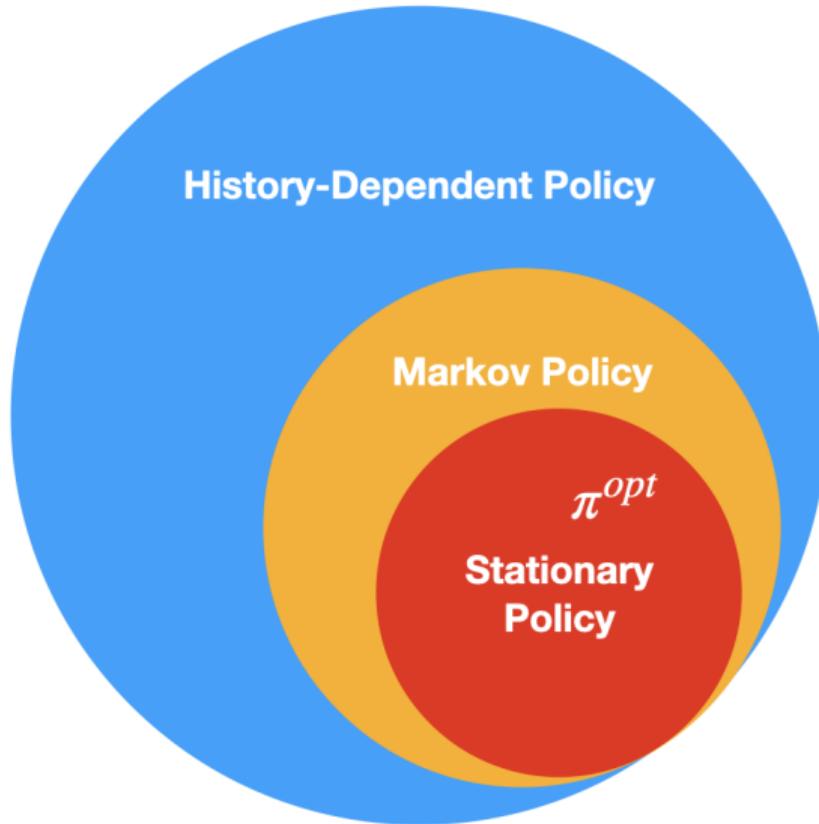
- each π_t^{opt} depends on the data history only through S_t
- $\mathbb{E}^{\pi^{opt}} G_0 \geq \mathbb{E}^\pi G_0$ for any **history-dependent policy** π

When the system dynamics satisfies the **Markov** assumption, so does the **optimal policy**.

In TMDPs



In MDPs



Summary

- Exploration-exploitation tradeoff
- ϵ -greedy
- Upper confidence bound
- Thompson sampling
- Multi-armed bandits
- Contextual bandits
- Markov decision processes

Summary (Cont'd)

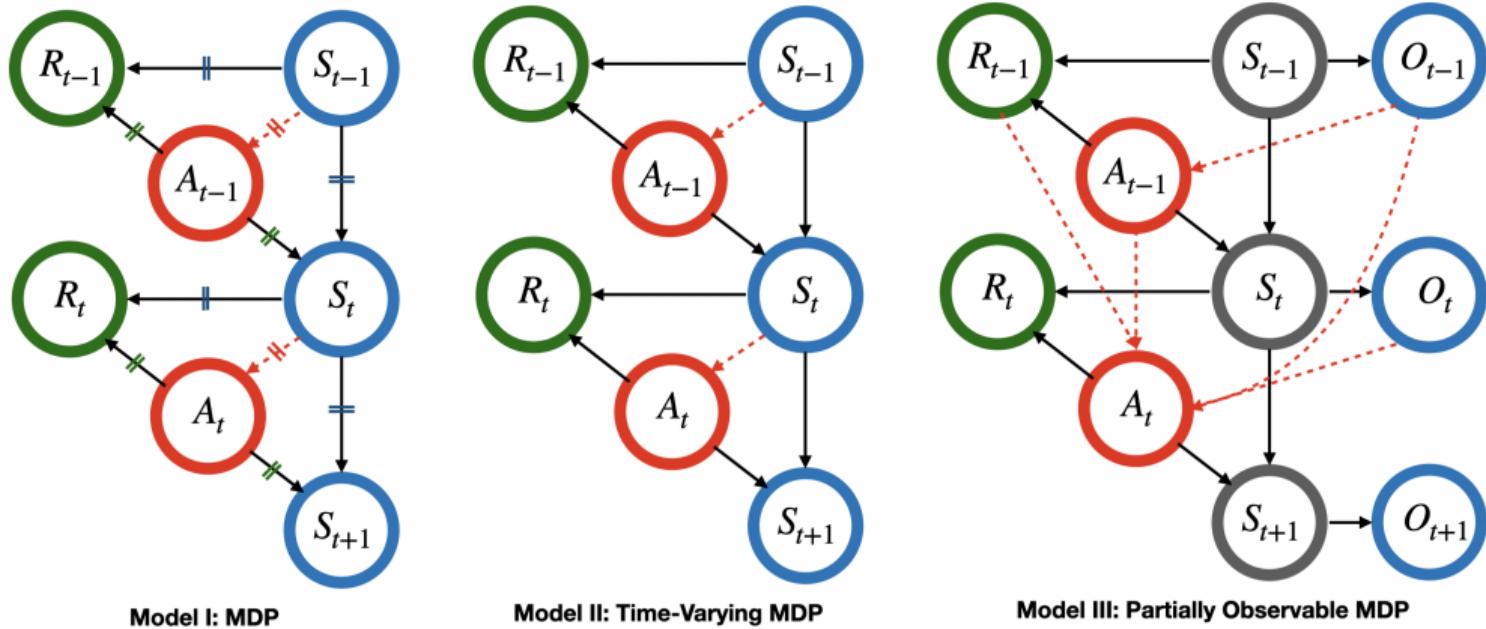


Figure: Causal diagrams for MDPs, TMDPs and POMDPs. Solid lines represent the causal relationships. Dashed lines indicate the information needed to implement the optimal policy. The parallel sign \parallel indicates that the conditional probability function given parent nodes is equal.

References |

- Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.
- Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135. PMLR, 2013.
- Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.
- Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.
- Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

References II

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- Chengchun Shi, Runzhe Wan, Rui Song, Wenbin Lu, and Ling Leng. Does the markov decision process fit the data: Testing for the markov property in sequential decision making. *arXiv preprint arXiv:2002.01751*, 2020.
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

References III

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

Questions