Markov Decision Processes & Dynamic Programming

Lucas Janson

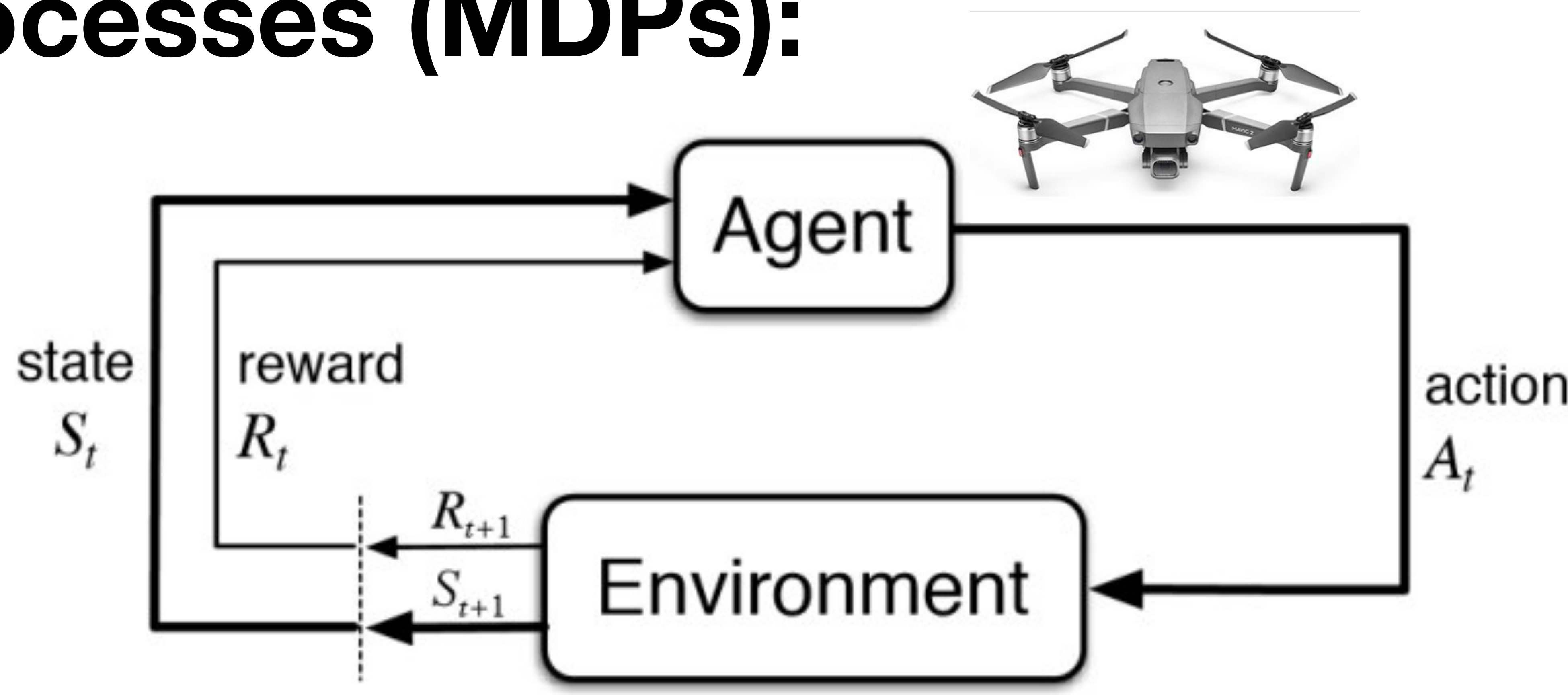
CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Today

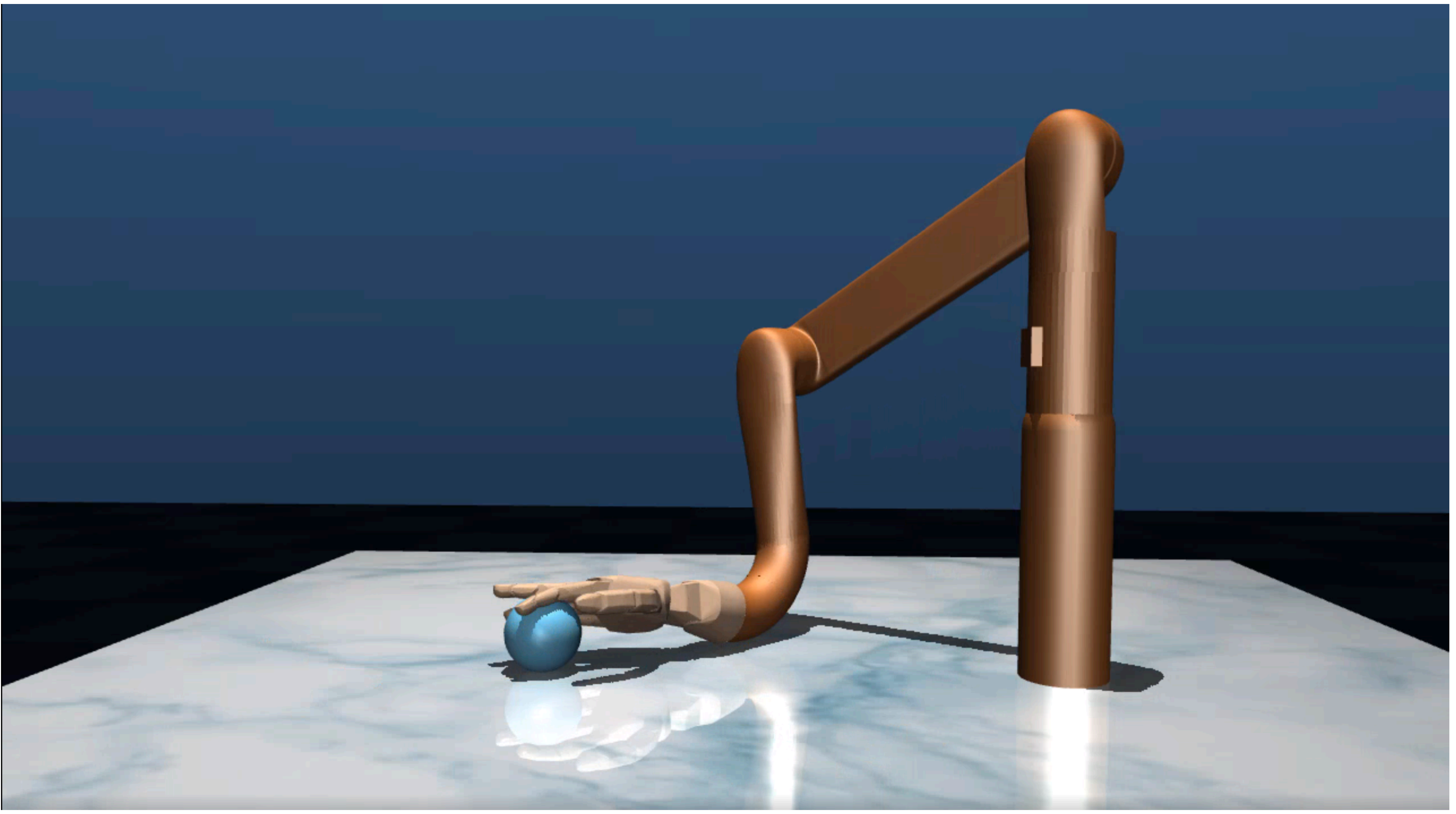
- Recap
- Problem Statement
- Bellman Consistency & Policy Evaluation
- Optimality
- The Bellman Equations & Dynamic Programming

Finite Horizon Markov Decision Processes (MDPs):



- An MDP: $\mathscr{M} = \{\mu, S, A, P, r, H\}$
  - μ is a distribution over initial states (sometimes we assume we start a given state s_0)
  - S a set of states
  - A a set of actions
  - $P : S \times A \mapsto \Delta(S)$ specifies the dynamics model, i.e. $P(s' \mid s, a)$ is the probability of transitioning to s' from state s via action a
  - $r : S \times A \to [0,1]$
    - For now, let's assume this is a deterministic function
    - (sometimes we use a cost $c : S \times A \to [0,1]$)
  - A time horizon $H \in \mathbb{N}$

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

Policy $\pi(s)$: a function mapping from robot state to action (i.e., torque)

Reward/Cost:

$r(s, a)$: immediate reward at state (s, a), or

$c(s, a)$: torque magnitude + dist to goal

Horizon: timescale H

$$\pi^\star = \arg\min_\pi \mathbb{E}\left[c(s_0, a_0) + c(s_1, a_1) + c(s_2, a_2) + \ldots c(s_{H-1}, a_{H-1}) \,\middle|\, s_0, \pi\right]$$

Today

- Recap
- Problem Statement
- Bellman Consistency & Policy Evaluation
- Optimality
- The Bellman Equations & Dynamic Programming

The Episodic Setting and Trajectories

- Policy $\pi := \{\pi_0, \pi_1, \ldots, \pi_{H-1}\}$
  - deterministic policies: $\pi_t : S \mapsto A$; stochastic policies: $\pi_t : S \mapsto \Delta(A)$
  - we also consider time-dependent policies (but not a function of the history)
- Sampling a trajectory τ on an episode: for a given policy π
  - Sample an initial state $s_0 \sim \mu$:
  - For $t = 0,1,2,\ldots H-1$
    - Take action $a_t \sim \pi_t(\cdot \mid s_t)$
    - Observe reward $r_t = r(s_t, a_t)$
    - Transition to (and observe) s_{t+1} where $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
  - The sampled trajectory is $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{H-1}, a_{H-1}, r_{H-1}\}$

The Probability of a Trajectory & The Objective

- Probability of trajectory: let $\rho_{\pi,\mu}(\tau)$ denote the probability of observing trajectory $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{H-1}, a_{H-1}, r_{H-1}\}$ when acting under π with $s_0 \sim \mu$.
  - Shorthand: we sometimes write ρ or ρ_π when π and/or μ are clear from context.
  - The rewards in this trajectory must be $r_t = r(s_t, a_t)$ (else $\rho_\pi(\tau) = 0$).
  - For π stochastic:
    $\rho_\pi(\tau) = \mu(s_0)\pi(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\ldots\pi(a_{H-2} \mid s_{H-2})P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi(a_{H-1} \mid s_{H-1})$
  - For π deterministic:
    $\rho_\pi(\tau) = \mu(s_0)\mathbf{1}\left(a_0 = \pi(s_0)\right)P(s_1 \mid s_0, a_0)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\mathbf{1}\left(a_{H-1} = \pi(s_{H-1})\right)$

- Objective: find policy π that maximizes our expected cumulative episodic reward:

$$\max_{\pi} \mathbb{E}_{\tau\sim\rho_\pi}\left[r(s_0, a_0) + r(s_1, a_1) + \ldots + r(s_{H-1}, a_{H-1})\right]$$

Today

- Recap
- Problem Statement
- Bellman Consistency & Policy Evaluation
- Optimality
- The Bellman Equations & Dynamic Programming

Policy Evaluation = Computing Value function and/or Q function

We evaluate policies via quantities that allow us to reason about the policy's long-term effect:

- Value function $V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \,\middle|\, s_h = s\right]$

- Q function $Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \,\middle|\, (s_h, a_h) = (s, a)\right]$

- At the last stage, what are:

$Q_{H-1}^\pi(s, a) =$ $\qquad\qquad$ $V_{H-1}^\pi(s) =$

Policy Evaluation = Computing Value function and/or Q function

We evaluate policies via quantities that allow us to reason about the policy's long-term effect:

- Value function $V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| s_h = s\right]$

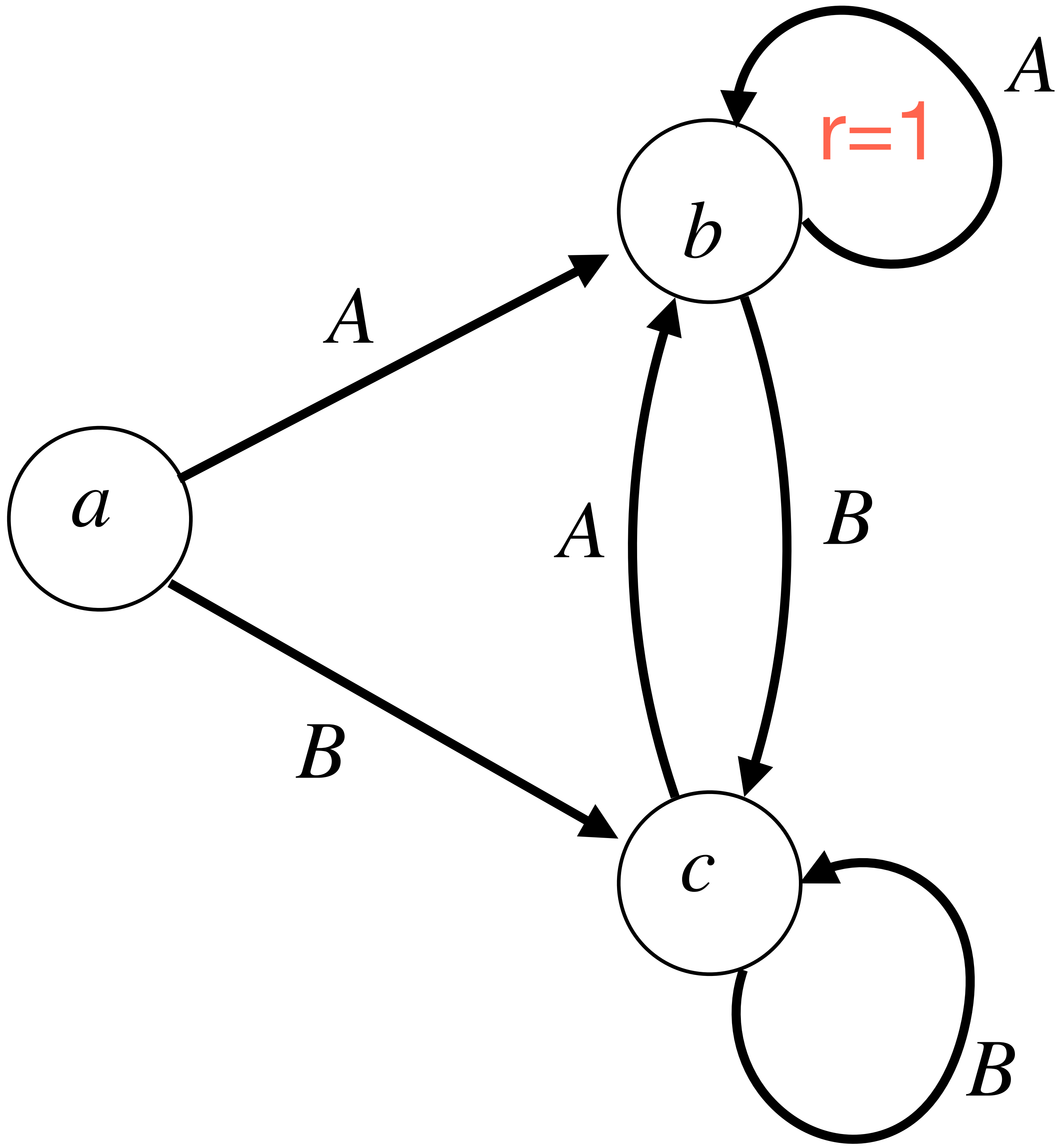
- Q function $Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| (s_h, a_h) = (s, a)\right]$

- At the last stage, for a stochastic policy:

$$Q_{H-1}^\pi(s, a) = r(s, a) \qquad V_{H-1}^\pi(s) = \sum_a \pi_{H-1}(a \mid s) r(s, a)$$

Example of Policy Evaluation (i.e. computing V^π and Q^π)

Consider the following **deterministic** MDP w/ 3 states & 2 actions, with $H = 3$



Reward: $r(b, A) = 1$, & 0 everywhere else

- Consider the deterministic policy
  $\pi_0(s) = A,\ \pi_1(s) = A,\ \pi_2(s) = B,\ \forall s$

- What is V^π?
  $V_2^\pi(a) = 0,\ V_2^\pi(b) = 0,\ V_2^\pi(c) = 0$

  $V_1^\pi(a) = 0,\ V_1^\pi(b) = 1,\ V_1^\pi(c) = 0$

  $V_0^\pi(a) = 1,\ V_0^\pi(b) = 2,\ V_0^\pi(c) = 1$

Notation

- $x \sim D$ means sampling from D
- $a \sim \pi(\cdot \mid s)$ means sampling from the distribution $\pi(\cdot \mid s)$, i.e. choosing action a with probability $\pi(a \mid s)$
- For a distribution D over a finite set $\mathcal{X}$,
$$E_{x \sim D}[f(x)] = \sum_{x \in \mathcal{X}} D(x) f(x)$$
- We use the notation:
$$E_{s' \sim P(\cdot \mid s,a)}[f(s')] = \sum_{s' \in S} P(s' \mid s, a) f(s')$$

Bellman Consistency

- For a **deterministic** policy, $\pi := \left\{\pi_0, \pi_1, \ldots, \pi_{H-1}\right\}, \pi_h : S \mapsto A, \forall h,$
- By definition, $V_h^\pi(s) = Q_h^\pi(s, \pi_h(s))$
- At $H-1$, $Q_{H-1}^\pi(s,a) = r(s,a), \;\; V_{H-1}^\pi(s) = r(s, \pi_{H-1}(s))$
- Bellman consistency conditions: for a given policy π,
  - $V_h^\pi(s) = r(s, \pi_h(s)) + \mathbb{E}_{s' \sim P(\cdot|s, \pi_h(s))}\left[V_{h+1}^\pi(s')\right]$
  - $Q_h^\pi(s,a) = r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[V_{h+1}^\pi(s')\right]$

Proof: Bellman Consistency for V-function:

Let $r_h = r(s_h, \pi(s_h))$ (note it's random via s_h)

By definition and by the law of total expectation:

$$V_h^\pi(s) = \mathbb{E}\left[r_h + r_{h+1} + \ldots + r_{H-1} \middle| s_h = s \right]$$

$$= \mathbb{E}\left[r_h + \mathbb{E}\left[r_{h+1} + \ldots + r_{H-1} \middle| s_h = s, s_{h+1} \right] \middle| s_h = s \right]$$

By the Markov property:

$$= \mathbb{E}\left[r_h + \mathbb{E}\left[r_{h+1} + \ldots + r_{H-1} \middle| s_{h+1} \right] \middle| s_h = s \right]$$

$$= \mathbb{E}\left[r_h + V_{h+1}^\pi(s_{h+1}) \middle| s_h = s \right]$$

$$= r(s, \pi_h(s)) + \mathbb{E}_{s' \sim P(\cdot | s, \pi_h(s))}\left[V_{h+1}^\pi(s') \right]$$

Computation of V^π via Backward Induction

- For a deterministic policy, $\pi := \{\pi_0, \pi_1, \ldots, \pi_{H-1}\}$, $\pi_h : S \mapsto A, \forall h$,
  Bellman consistency $\implies$ we can compute V_h^π, assuming we know the MDP.

- Initialize: $V_H^\pi(s) = 0, \ \forall s \in S$
- For $h = H - 1, \ldots 0$, set:
  $V_h^\pi(s) = r(s, \pi_h(s)) + \mathbb{E}_{s' \sim P(\cdot|s, \pi_h(s))}\left[V_{h+1}^\pi(s')\right], \ \forall s \in S$

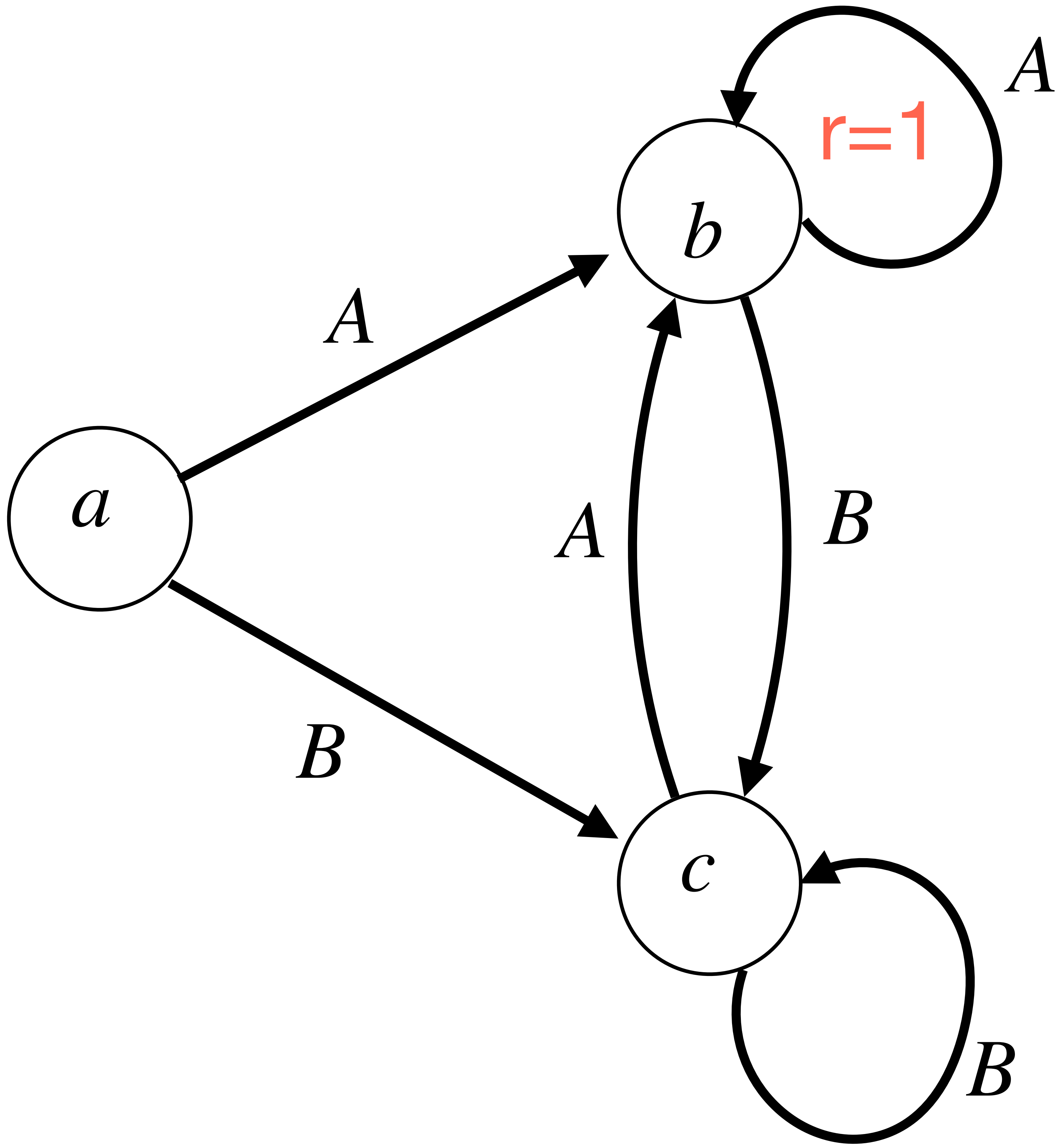
- What is the per iteration computational complexity of DP?
  (assume scalar $+, -, \times, \div$ are $O(1)$ operations)
- What is the total computational complexity of DP?

Today

- Recap
- Problem Statement
- Bellman Consistency & Policy Evaluation
- Optimality
- The Bellman Equations & Dynamic Programming

Example of Optimal Policy $\pi^\star$

Consider the following **deterministic** MDP w/ 3 states & 2 actions, with $H = 3$



Reward: $r(b, A) = 1$, & 0 everywhere else

- What's the optimal policy?

  $\pi_h^\star(s) = A, \ \ \forall s, h$

- What is optimal value function, $V^{\pi^\star} = V^\star$?

  $V_2^\star(a) = 0, \ \ V_2^\star(b) = 1, \ \ V_2^\star(c) = 0$

  $V_1^\star(a) = 1, \ \ V_1^\star(b) = 2, \ \ V_1^\star(c) = 1$

  $V_0^\star(a) = 2, \ \ V_0^\star(b) = 3, \ \ V_0^\star(c) = 2$

How do we compute $\pi^\star$ and $V^\star$?

- Naively, we could compute the value of all policies and take the best one.
- Suppose $|S|$ states, $|A|$ actions, and horizon H.
  How many different polices there are?

- Can we do better?

Properties of an Optimal Policy $\pi^\star$

- Let Π be the set of all time dependent, history dependent, stochastic policies.

- **Theorem:** Every finite horizon MDP has a deterministic, history-independent optimal policy, that dominates all other policies, everywhere.
  - i.e. there exists a deterministic policy $\pi^\star := \left\{\pi_0^\star, \pi_1^\star, \ldots, \pi_{H-1}^\star\right\}, \pi_h^\star : S \mapsto A$ such that

$$V_h^{\pi^\star}(s) \geq V_h^{\pi}(s) \quad \forall s, h, \forall \pi \in \Pi$$

- $\implies$ we can write: $V_h^\star = V_h^{\pi^\star}$ and $Q_h^\star = Q_h^{\pi^\star}$.
- $\implies$ $\pi^\star$ doesn't depend on the initial state distribution μ.

What's the Proof Intuition?

- **Theorem:** Every finite horizon MDP has a deterministic, history-independent optimal policy, that dominates all other policies, everywhere.

- What's the Proof Intuition?
  - “Only the state matters”: how got here doesn’t matter to where we go next, conditioned on the action.
    - This explains both determinism and history-independence
- Caveat: some legitimate reward functions are not additive/linear (so, naively, not an MDP). (But, RL is general: think about redefining the state so you can do these.)

Today

- Recap
- Problem Statement
- Bellman Consistency & Policy Evaluation
- Optimality
- The Bellman Equations & Dynamic Programming

The Bellman Equations

- A function $V = \{V_0, \ldots V_{H-1}\},\ V_h : S \to R$ satisfies the Bellman equations if

$$V_h(s) = \max_a \left\{ r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{h+1}(s') \right] \right\}, \forall s$$

(assume $V_H = 0$).

- **Theorem:**
  - V satisfies the Bellman equations if and only if $V = V^\star$.
  - The optimal policy is: $\pi_h^\star(s) = \arg\max_a \left\{ r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{h+1}^\star(s') \right] \right\}$.

Computation of $V^\star$ with Dynamic Programming

- Theorem: the following Dynamic Programming algorithm computes $\pi^\star$ and $V^\star$
  Prf: the Bellman equations directly lead to this backwards induction.

- Initialize: $V_H^\pi(s) = 0 \ \forall s \in S$
  For t= $H-1,\ldots 0$, set:
  - $V_h^\star(s) = \max_a \left[r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{h+1}^\star(s') \right] \right], \forall s \in S$
  - $\pi_h^\star(s) = \arg\max_a \left[r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{h+1}^\star(s') \right] \right], \forall s \in S$

- What is the per iteration computational complexity of DP?
  (assume scalar $+, -, \times, \div$ are $O(1)$ operations)
- What is the total computational complexity of DP?

Summary:

- **Dynamic Programming lets us efficiently compute optimal policies.**
  - We remember the results on “sub-problems”
  - Optimal policies are history independent.

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy