CS 760: Machine Learning
Reinforcement Learning I

Josiah Hanna

University of Wisconsin-Madison

November 28, 2023

Announcements

- Homework 7 due December 7 at 9:30 am.
- Final exam: December 18 from 2:45 - 4:45 pm in the Social Sciences building.
- Course evaluation due 12/13.
- Looking ahead: this week and next on RL; then societal impacts.

Lecture Goals

At the end of today's lecture, you will be able to:

1. Formulate a sequential decision-making application as a reinforcement learning (RL) problem.

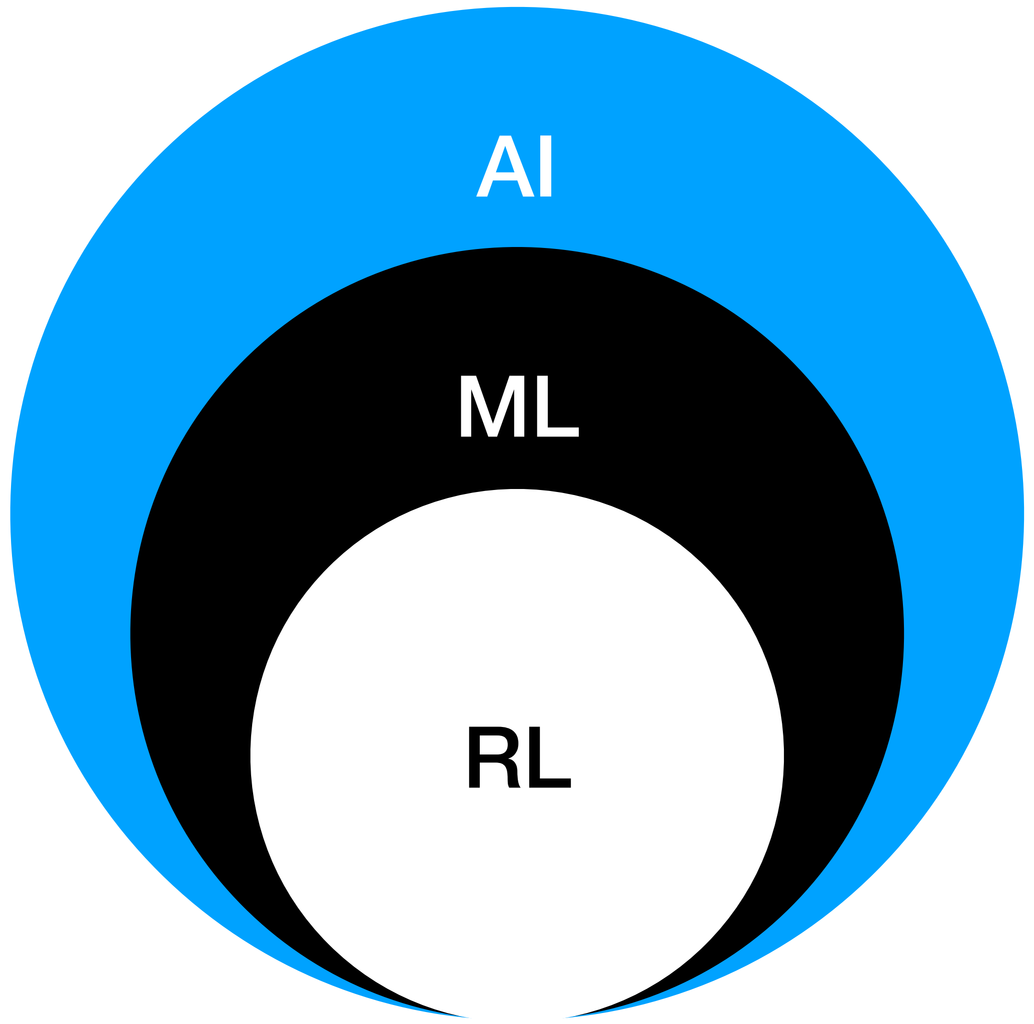
2. Be able to define key RL terminology such as policies and value functions.

What is Reinforcement Learning?

- Machine learning paradigm that focuses on learning from rewards and trial and error interaction.
- The learning agent takes actions, receives rewards, and over time learns to take actions that lead to the most reward.
- Think: training a dog to do tricks.

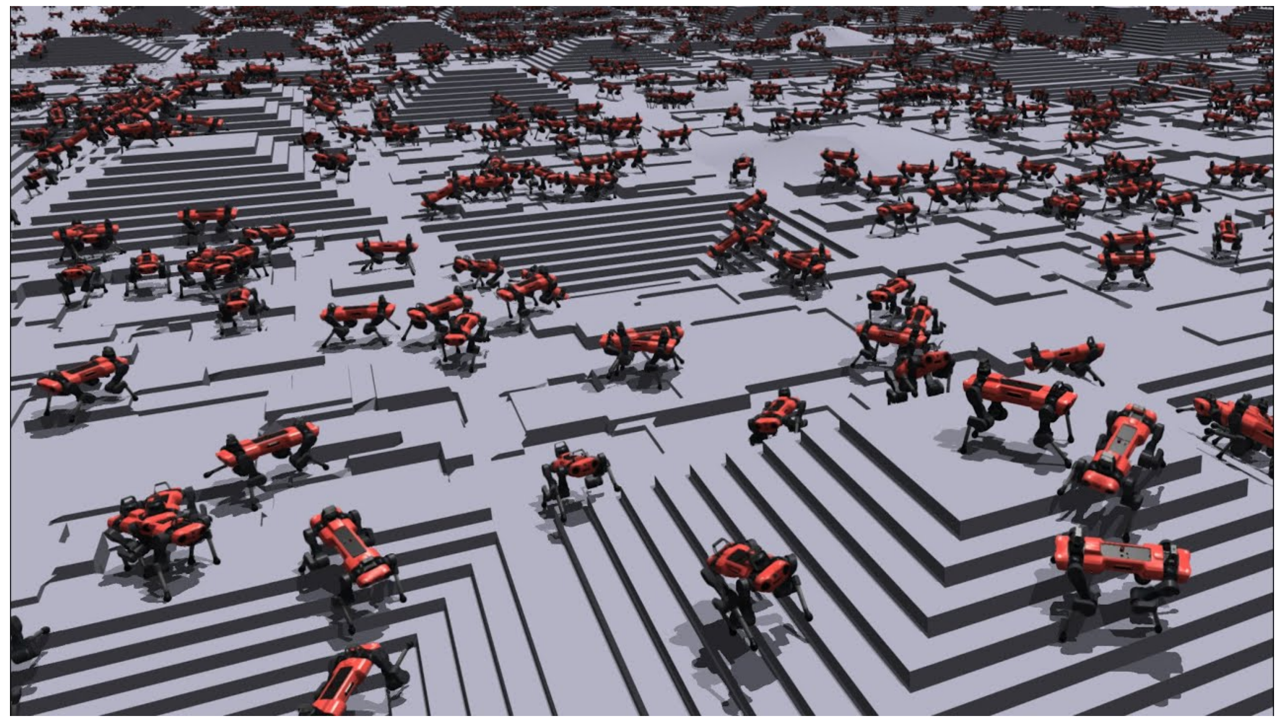
RL within Artificial Intelligence

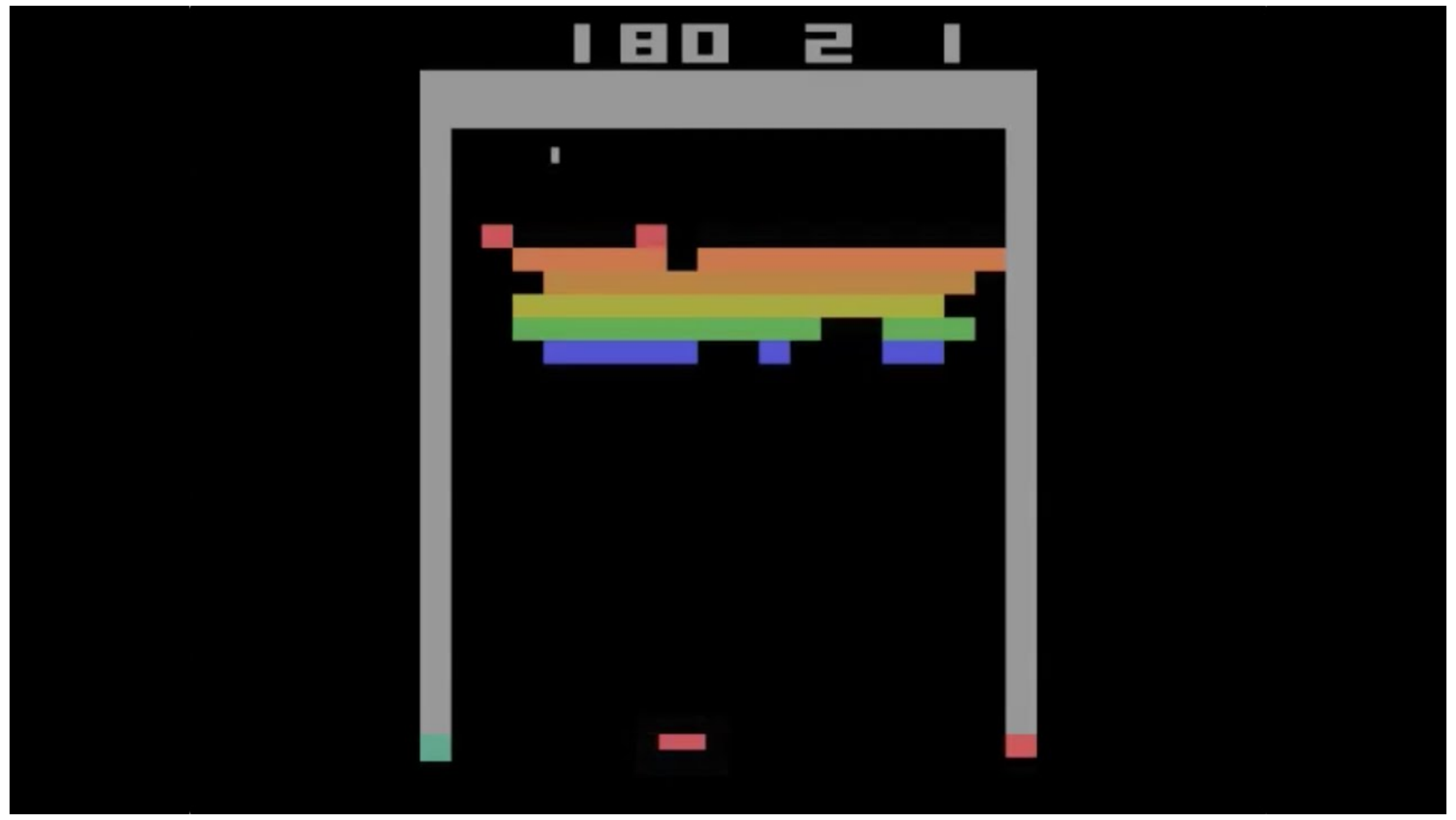
- Supervised learning: learn from labelled examples.
  - Given a data set of $\{(x_i, y_i)\}_{i=1}^{m}${(X,Y)}, learn to map new instances of x to appropriate y.
  - Ex: image classification, object detection, spam filtering.
- Unsupervised learning: discover structure in unlabelled data.
  - Ex: clustering, generating images, language modeling
- Reinforcement learning: learn from rewarded interaction.
- Reinforcement learning also relates to non-learning AI planning methods.



What Can RL Do?

- Play video games
- Play board games
- Control robots
- Recommend ads and web content
- Trade stocks
- Recommend medical treatments
- Control home thermostat systems
- Cooling of data centers
- Networking
- Databases
- Program Synthesis





Be an RL Agent*

- You (as a class) are the learning agent.
- Three actions: stand, clap, or wave
- Observations: colors
- Rewards: depends on color you see and action you take.
- Goal: find the optimal policy.
    - Policy: mapping from colors to actions.
    - Optimal policy: policy that gives you the most reward.

*** Activity credit to Peter Stone.**

Be an RL Agent

- How did you learn?
- What structure does the world have?

Challenges of Reinforcement Learning

- Credit Assignment:
  - May take many actions before reward is received. Which ones were most important?
  - Example: you study 15 minutes a day all semester. The morning of the final exam, you eat a bowl of yogurt. You receive an A on the final. Was it the studying or the yogurt that led to the A?
- Exploration vs. Exploitation
  - Should you keep trying actions that led to reward in the past or try new actions that might lead to even more reward?

Markov Decision Processes

RL problems are formalized as Markov decision processes, $\langle \mathcal{S}, \mathcal{A}, r, p \rangle$:

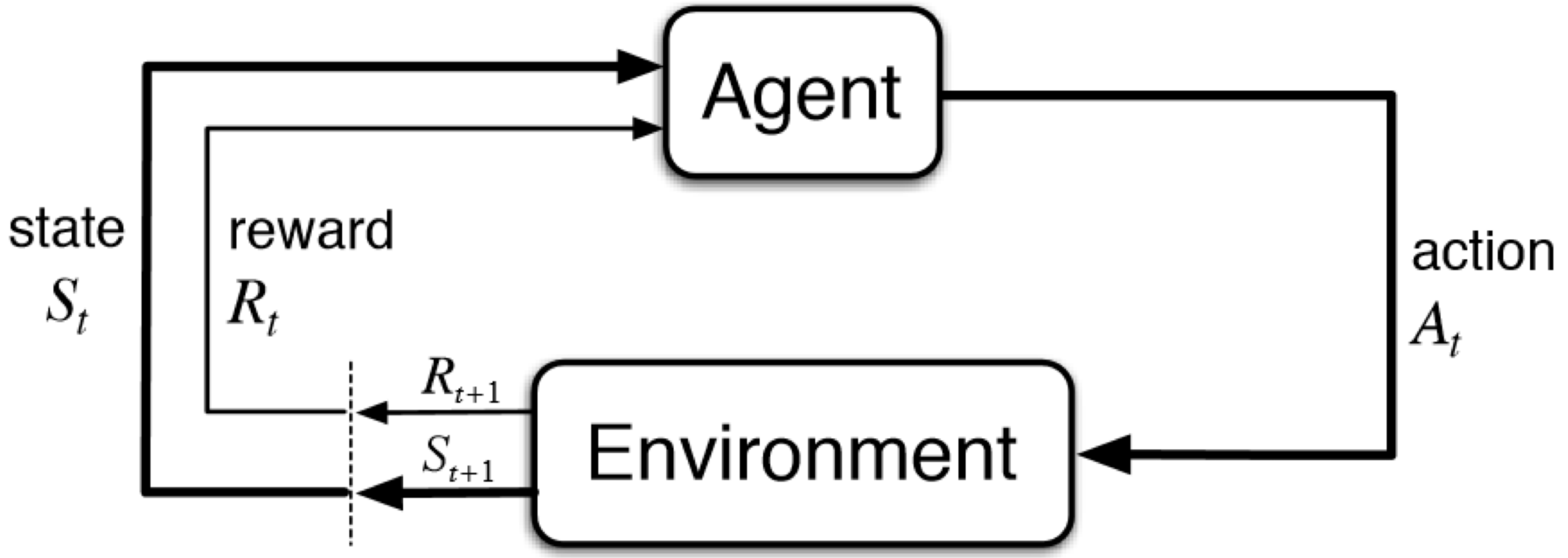
- States: $s \in \mathcal{S}$
- Actions: $a \in \mathcal{A}$
- Rewards: $R \sim r(s, a)$
- State transitions: $S \sim p(\cdot \mid s, a)$
    - **Markov property:** next state only depends on current state and action taken.
- Goal: Find a policy, $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes cumulative reward.

We do not know r and p. This is the learning challenge!

For brevity will use $p(s', r \mid s, a)$ to denote joint probability of next state and reward.

Data in Reinforcement Learning

Agent learns from the sequence of data seen while acting in task Markov decision process:

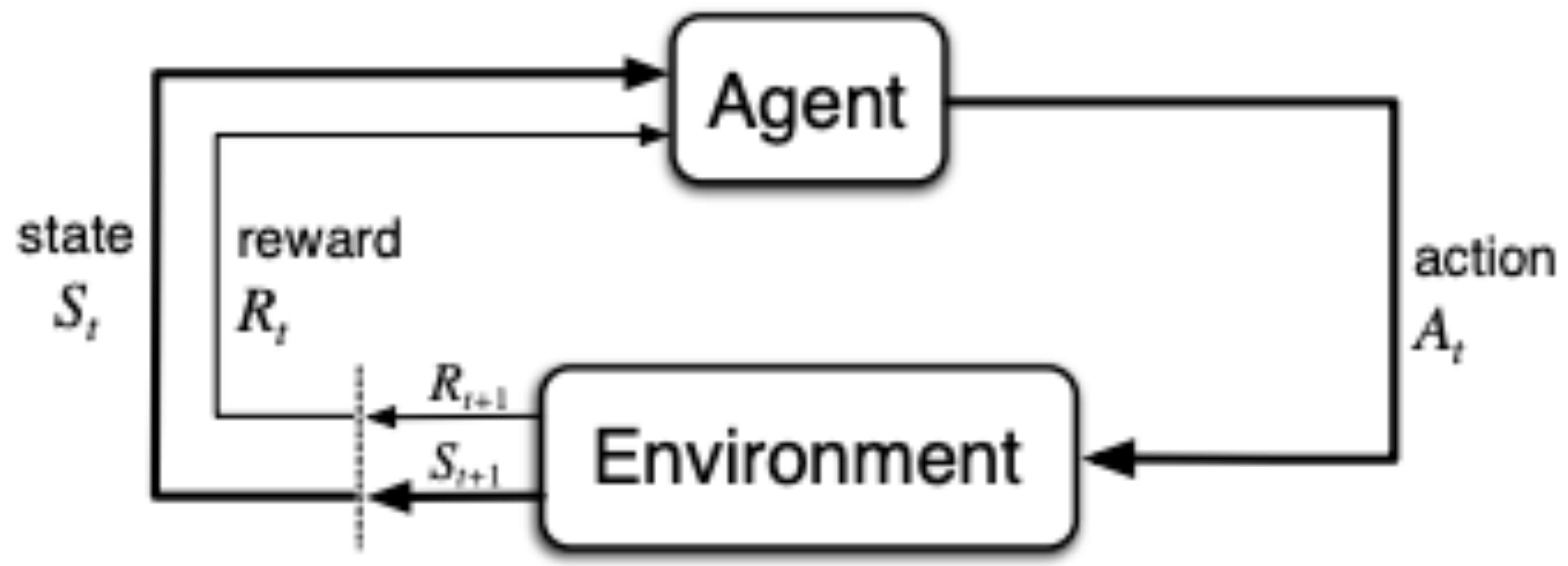


$$\ldots S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, \ldots$$

$$S_{t+1}, R_{t+1} \sim p(\cdot \,|\, S_t, A_t)$$

$$A_{t+1} \leftarrow \pi(S_{t+1})$$

Reinforcement Learning



Agent's objective is to find policy, π, so as to maximize the expected cumulative discounted reward from each state:

$$v_\pi(s) = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R_t \,|\, S_0 = s, A_t \leftarrow \pi(S_t), S_{t+1} \sim p(\cdot \,|\, S_t, A_t)]$$

$$= \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \,|\, S_0 = s, A_t \leftarrow \pi(S_t), S_{t+1} \sim p(\cdot \,|\, S_t, A_t)]$$

For brevity, $\mathbf{E}_\pi$ will be used for $\mathbf{E}[\ldots \,|\, A_t \leftarrow \pi(S_t), S_{t+1}, R_{t+1} \sim p(\cdot \,|\, S_t, A_t)]$

Example RL Problems

- What are the states? Actions? Rewards?
- Atari Breakout
- Home thermostat
- Stock trading

Defining State

- Informally, state is the information available to the agent to base its decision on.
- Formally, an element of the state space, i.e., $s \in \mathcal{S}$.
- Must include information about all aspects of the past that affect the future.
- **Markov property:** future is conditionally independent of the past given current state.

$$\Pr(S_{t+1} = s, R_{t+1} = r \,|\, s_t, a_t) = \Pr(S_{t+1} = s, R_{t+1} = r \,|\, s_t, a_t, s_{t-1}, a_{t-1}, \ldots)$$

Thinking about State

- States as elements of a finite set.
    - Simpler model to analyze.
- State as a collection of variables that describe the world at that moment in time.
    - For example, an autonomous vehicle's state includes the vehicle's location, where other vehicles are, road conditions, etc.
    - I.e., states are feature vectors in $\mathbb{R}^d$.

State Examples

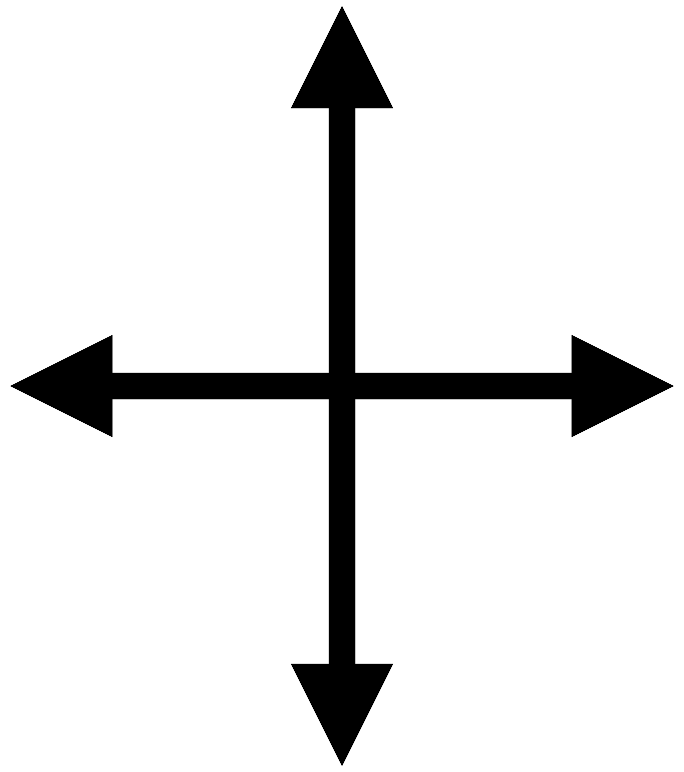
- Recommendation agent for a social media timeline.
- A robot with a camera and a laser range finder.
- Home thermostat system.
- Recommending medical treatment.

Defining Reward

- The agent's objective is to maximize its cumulative reward.
- Expected reward, $r(s, a)$, gives immediate benefit or cost of taking action a in state s.
- Ideally, communicates what to achieve not how to achieve it.
- In practice, reward often used to guide learning agent ("shaping" reward).

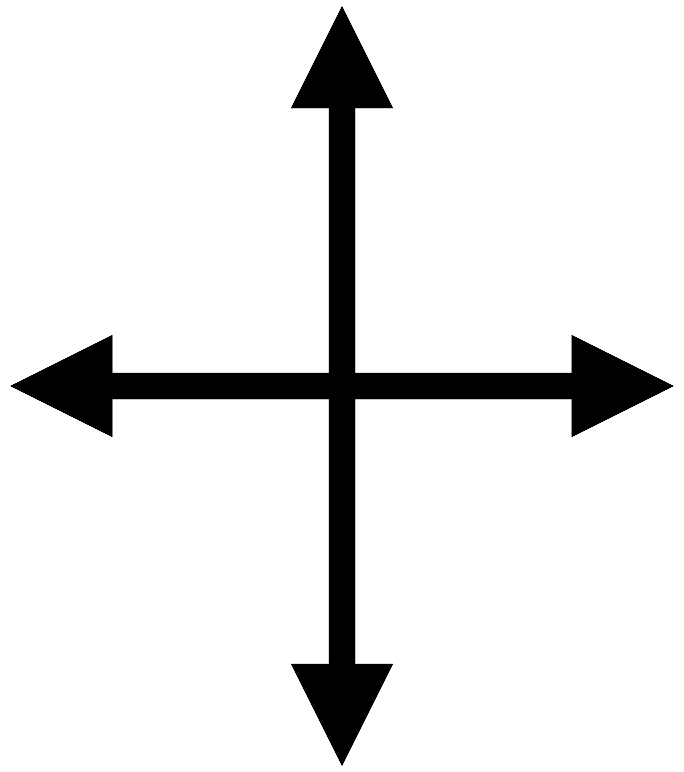
Reward Examples

0	0	0	Start
0	0	0	0
0	0	0	0
0			
0	0	0	+1



Reward Examples

0	0	0	Start
0	0	0	0.1
0.5	0.4	0.3	0.2
0.6			
0.7	0.8	0.9	+1



Reward Examples

- Recommendation agent for a social media timeline.
- An autonomous vehicle learning to drive.
- Home thermostat system.
- Recommending medical treatment.

Policies

- The agent's decision making rule.
- Formally, a function outputting the conditional probability of selecting an action in a particular state: $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$.
- A deterministic policy is a function mapping states to actions: $\pi : \mathcal{S} \rightarrow \mathcal{A}$.

Returns and Episodes

- Episodes are subsequences of interaction that begin in some initial state and end in a special terminal state.

- **The initial state of one episode is independent of interaction in the preceding episode.**

- The return from step t is: $G_t := R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots$

- Recursive definition: $G_t = R_{t+1} + \gamma G_{t+1}$.

Value functions

- Many RL algorithms use **value functions** to aid in long-term credit assignment.
- Two types of value function: state-value and action-value functions.

$$v_\pi(s) = \mathbb{E}_\pi[G_t \,|\, S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|\, S_t = s]$$

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \,|\, S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|\, S_t = s, A_t = a]$$

Recursive Relationship of State Values

$$v_\pi(s) := \mathbb{E}_\pi[G_t \mid S_t = s]$$

Recursive definition of return

$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

Definition of expectation

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)[r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']]$$

Definition of state-value

$$= \sum_a \pi(a \mid s) \sum_{s'} \sum_r p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

Final equation is called the Bellman equation for state values.

Page 59 of “Reinforcement Learning: An Introduction”

Action Values

Write action-values in terms of environment dynamics and state-values:

$$q_\pi(s, a) := \mathbb{E}_\pi[G_t \,|\, S_t = s, A_t = a]$$

Definition of return

$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \,|\, S_t = s, A_t = a]$$

Definition of expectation

$$= \sum_{s'} \sum_{r} p(s', r \,|\, s, a)[r + \gamma \mathbb{E}_\pi[G_{t+1} \,|\, S_{t+1} = s']]$$

Definition of state-value

$$= \sum_{s'} \sum_{r} p(s', r \,|\, s, a)[r + \gamma v_\pi(s')]$$

Exercise 3.13, page 58 of "Reinforcement Learning: An Introduction"

Action Values

Write state-values in terms of action-values:

From previous slide

$$q_\pi(s,a) = \sum_{s'} \sum_r p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

From two slides back.

$$v_\pi(s) = \sum_a \pi(a \mid s) \underbrace{\sum_{s'} \sum_r p(s', r \mid s, a)[r + \gamma v_\pi(s')]}_{q_\pi(s,a)}$$

$$v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$$

Exercise 3.12, page 58.

Golf Example

- State is ball location. Actions are putt (short distance, accurate) or drive ball (long distance, less accurate).
- Reward is -1 until the ball goes in the hole.
- What is value of policy that always putts?

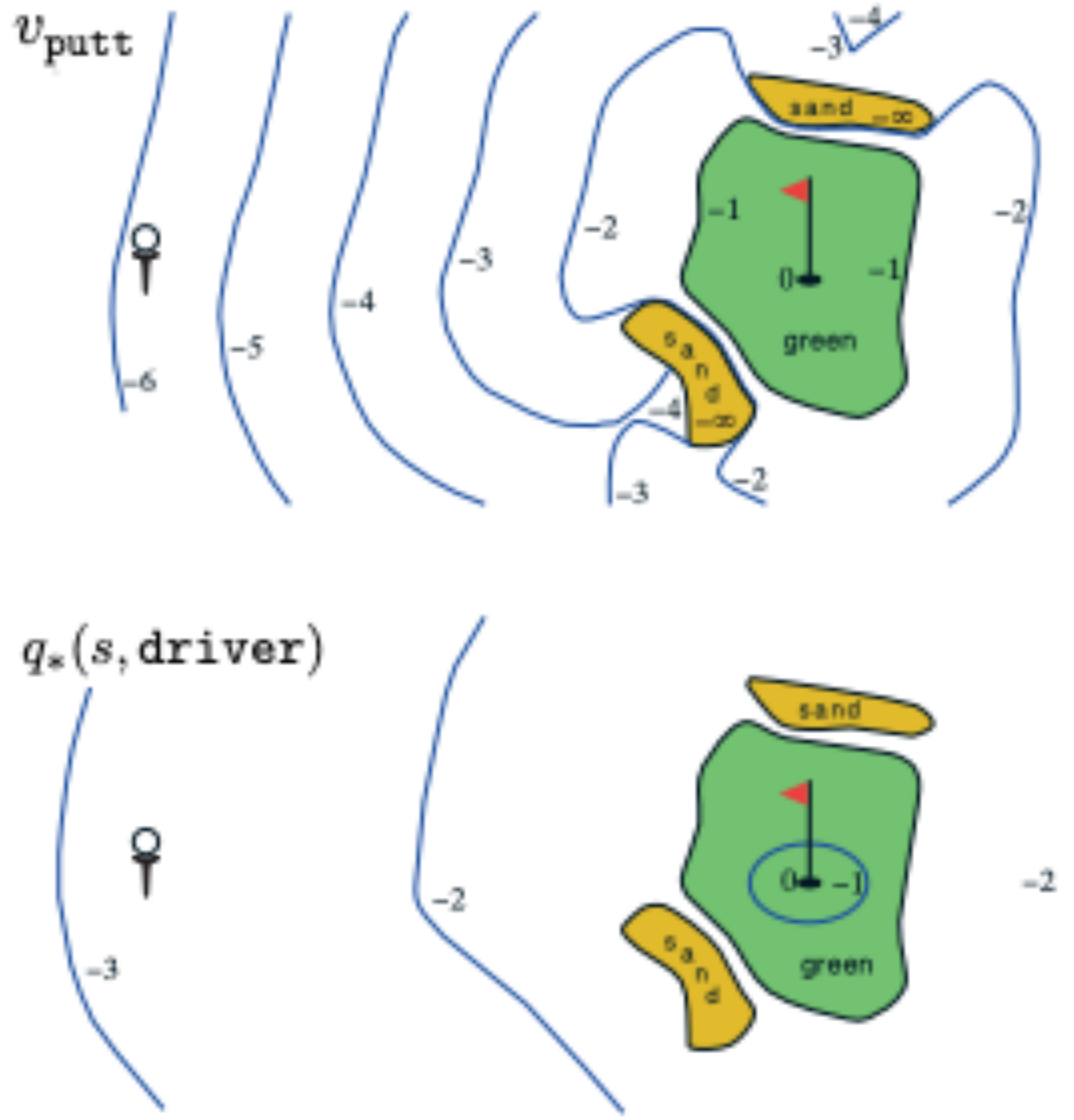


Figure 3.3: A golf example: the state-value function for putting (upper) and the optimal action-value function for using the driver (lower). ■

Optimality

- Agent's objective: find policy that maximizes $v_\pi(s)$ for all s.

- The optimal policy — policy that has maximal value in all states. $\pi^\star \geq \pi$ if $v_{\pi^\star}(s) \geq v_\pi(s)$ for all states and possible policies.

- Possibly multiple but always at least one deterministic optimal policy in a finite MDP.

- $\pi^\star(s) = \arg\max_a q_\star(s, a)$ $\qquad$ $q_\star(s, a) = \mathbf{E}_\pi[R_{t+1} + \gamma v_\star(S_{t+1}) \mid S_t = s, A_t = a]$

Value of taking action a and then acting optimally for all future time-steps.

Golf Example

- State is ball location. Actions are putt (short distance, accurate) or drive ball (long distance, less accurate).
- Reward is -1 until the ball goes in the hole.
- What is action-value of using driver and then following the optimal policy?

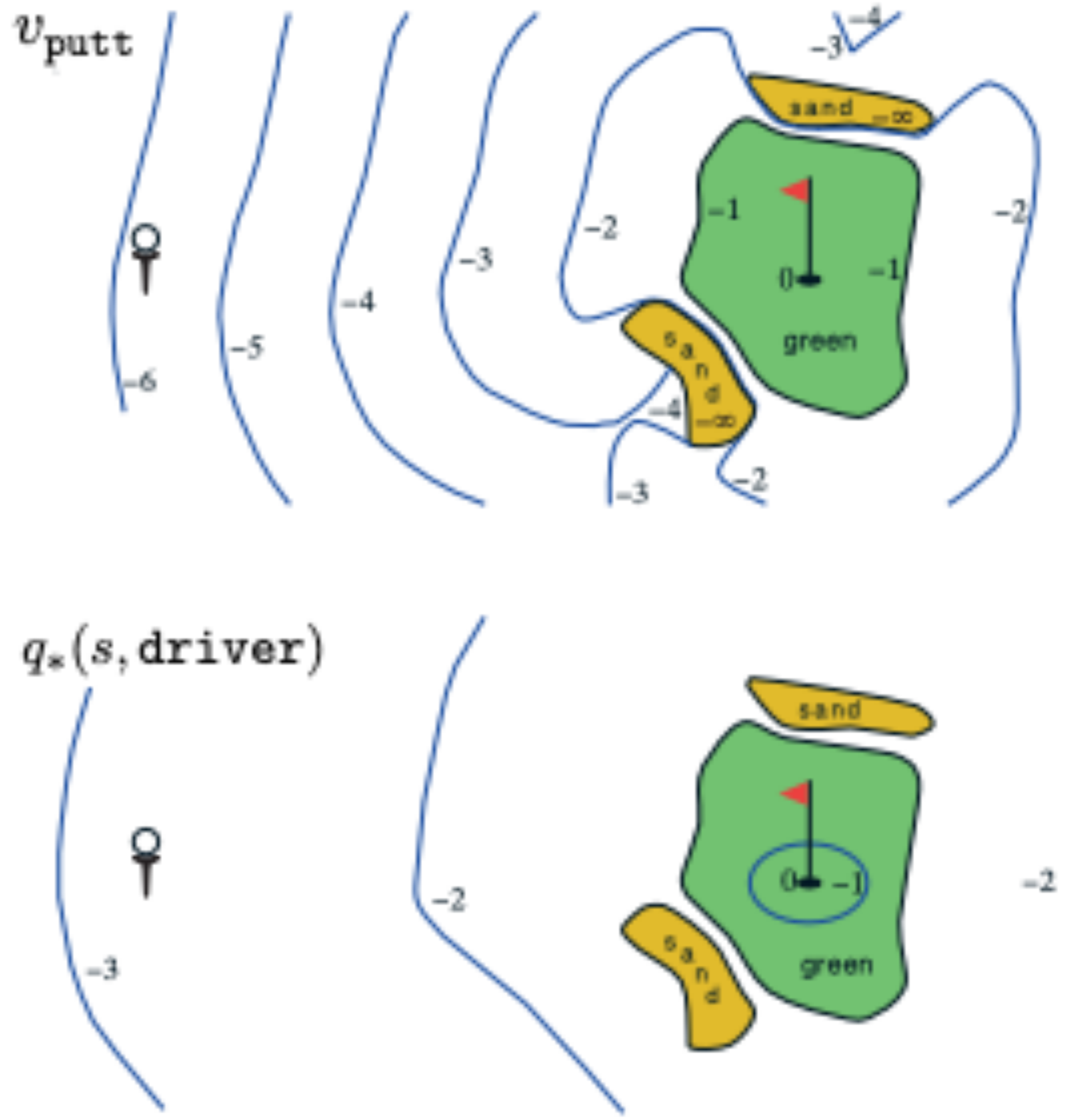


Figure 3.3: A golf example: the state-value function for putting (upper) and the optimal action-value function for using the driver (lower). ■

Quiz

Consider an MDP with 2 states, {A,B}, and 2 actions, {“stay”, “move”}. Let r be the reward function such that r(A) = 1 and r(B) = 0. Let γ be the discount factor and let $\pi(A) = \pi(B) = \texttt{move}$. What is the value function $v_\pi(A)$?

1. 0
2. $\dfrac{1}{1-\gamma}$
3. $\dfrac{1}{1-\gamma^2}$
4. 1

Quiz

Consider an MDP with 2 states, {A,B}, and 2 actions, {“stay”, “move”}. Let r be the reward function such that r(A) = 1 and r(B) = 0. Let γ be the discount factor and let $\pi(A) = \pi(B) = \texttt{move}$. What is the value function $v_\pi(A)$?

1. 0
2. $\frac{1}{1-\gamma}$
3. $\frac{1}{1-\gamma^2} = 1 + \gamma(0) + \gamma^2(1) + \gamma^3(0) + \gamma^4(1) + \ldots = \sum_{k=0}^{\infty} 1(\gamma^2)^k$
4. 1

Or can solve using Bellman equations:

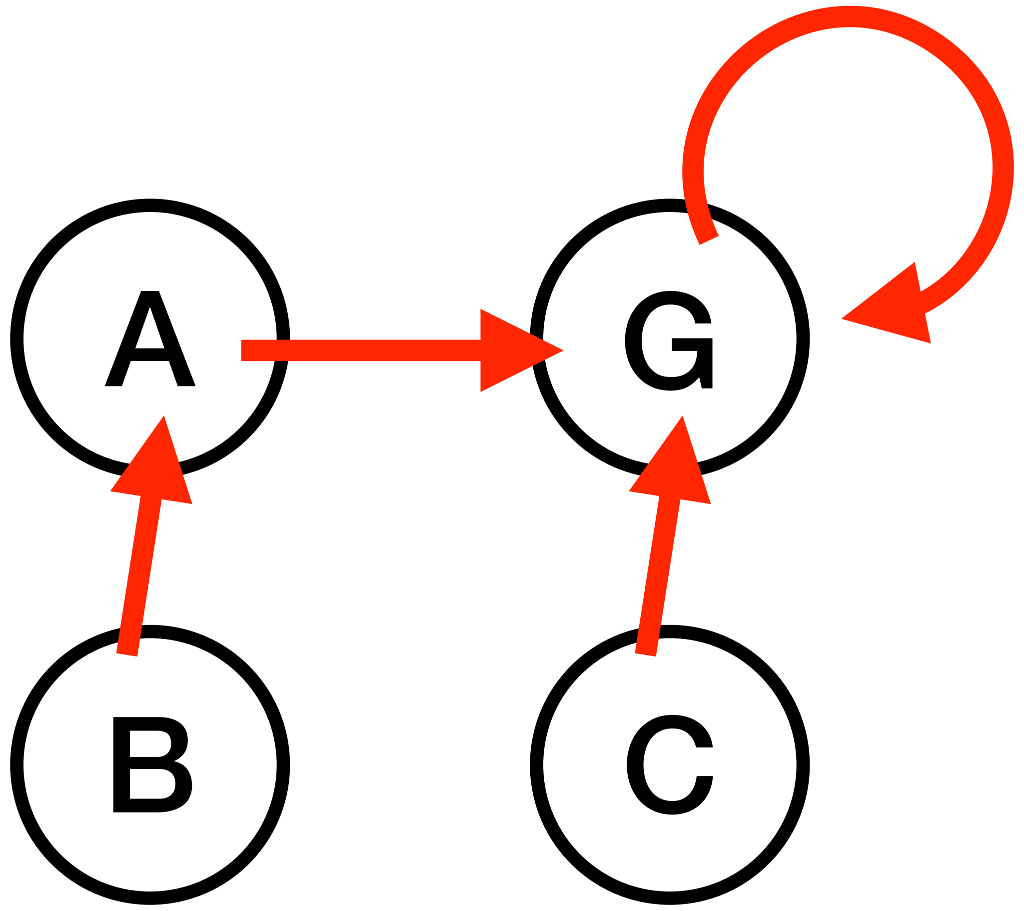
$$v_\pi(A) = 1 + \gamma v_\pi(B)$$
$$v_\pi(B) = 0 + \gamma v_\pi(A)$$

Thus, $v_\pi(A) = 1 + \gamma^2 v_\pi(A)$

Then solve for $v_\pi(A)$

Quiz

Consider the following MDP which has deterministic transitions and $\gamma = 0.8$. The policy's action is shown with a red arrow. What is $v_\pi(B)$ in this MDP?



Two approaches:

1. Compute reward total for entire (infinite) sequence).
2. Compute $v_\pi(G)$ then $v_\pi(A)$ and then $v_\pi(B)$.

r(B) = 20; r(A) = 10; r(C) = 20; r(G) = 100

Summary

- Formalized RL problems (Markov decision processes) and the learning objective.
- Agent's state must include all information from past that is needed to predict the future — Markov property.
- Terms to know: Policy, return, value function.
- The value of a policy in a given state is the expected return from that state.
- The optimal policy maximizes the value function in all states.

Thanks Everyone!

Slides adapted from Advanced Topics in RL and based on Chapter 3 of Reinforcement Learning: An Introduction.