Reinforcement Learning

Emma Brunskill
Stanford University
Spring 2017

With thanks to Dan Klein, Ariel Procaccia and other colleagues for slide inspiration

Welcome! Today’s Plan

- Overview about reinforcement learning
- Course logistics
- Introduction/review of sequential decision making under uncertainty

Reinforcement Learning

Learn to make good sequences of decisions

Repeated Interactions with World

Learn to make good sequences of decisions

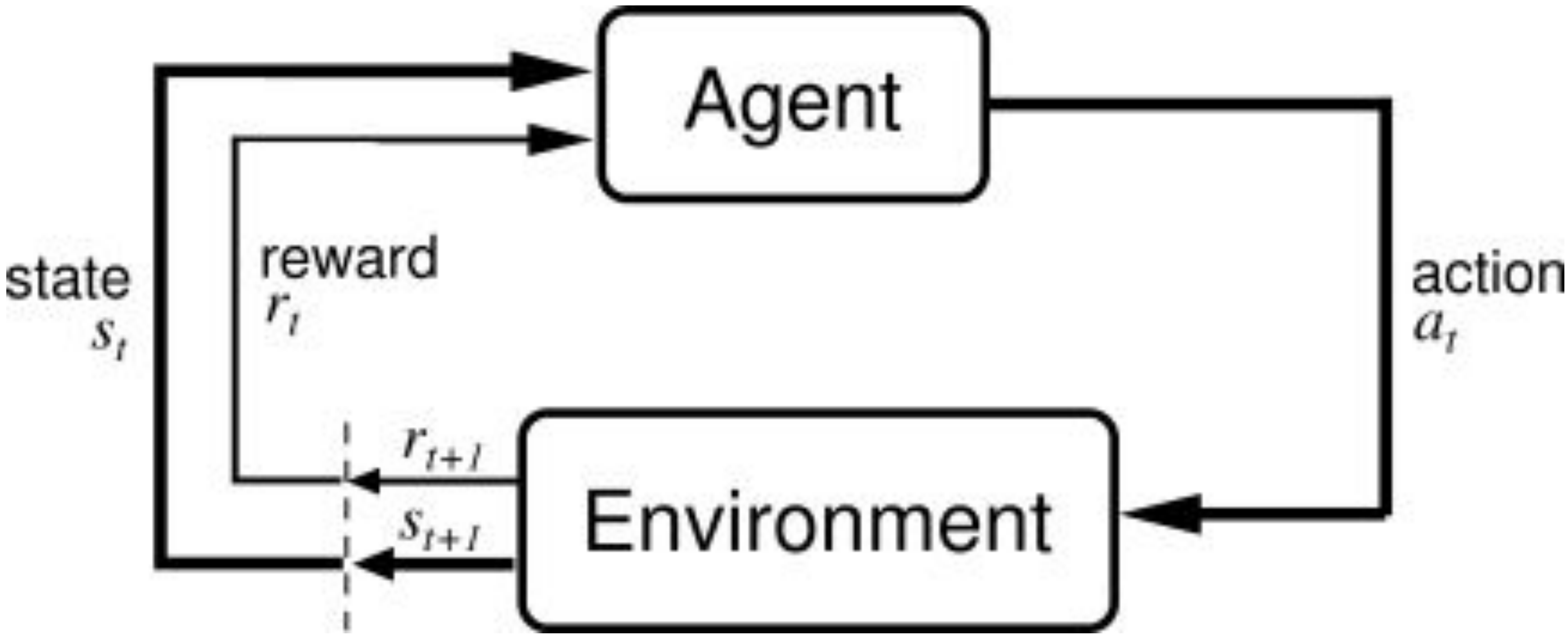
Reward for Sequence of Decisions

Learn to make good sequences of decisions

Don’t Know in Advance How World Works

Learn to make good sequences of decisions

Reinforcement Learning



Policy: mapping from history of past actions, states, rewards to next action

Figure from Sutton & Barto, RL textbook

Critical Component of Intelligence

- Understanding and advancing how an artificial agent can learn to make good decisions to do new tasks is fundamental challenge in artificial intelligence and machine learning

RL, Behavior & Intelligence

Childhood: primitive brain & eye, swims around, attaches to a rock

Adulthood: digests brain. Sits

Suggests brain is helping guide decisions (no more decisions, no need for brain?)

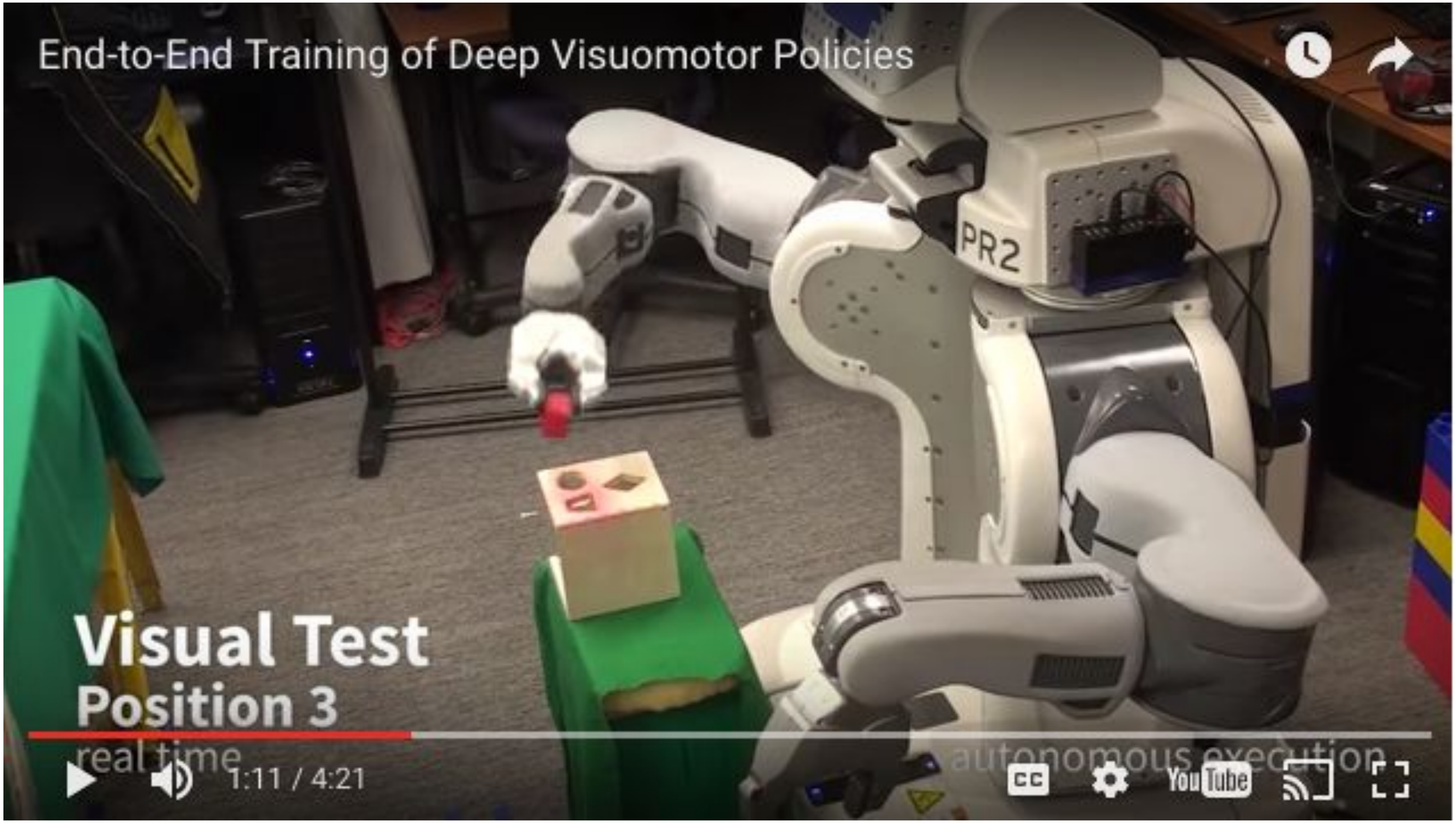
Example from Yael Niv

- Reinforcement Learning and Decision Making: multidisciplinary conference, this year chaired by me and Nathaniel Daw

Atari

DeepMind Nature 2015

Robotics

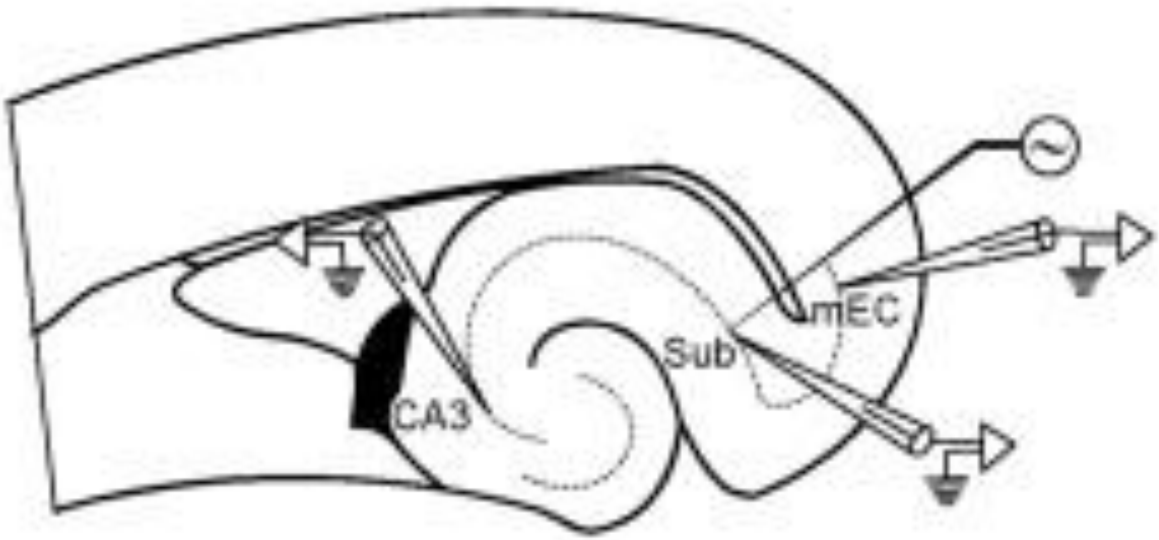


https://youtu.be/CE6fBDHPbP8?t=71 Finn, Leveine, Darrell, Abbeel JMLR 2017

Educational Games

RL used to optimize Refraction 1, Mandel, Liu, Brunskill, Popovic AAMAS 2014

Healthcare



Adaptive control of epileptiform excitability in an in vitro model of limbic seizures.

Panuccio,Guez, Vincent, Avoli Pineau

NLP, Vision, ...

Yeung, Russakovsky, Mori, Li *2016*

Reinforcement Learning Involves

- Optimization
- Generalization
- Exploration
- Delayed consequences

Delayed Consequences

- Decisions now can impact things much later…
  - Saving for retirement
  - Finding a key in Montezuma's revenge
- Introduces two challenges
  1) When planning: decisions involve reasoning about not just immediate benefit of a decision but how its longer term ramifications
  2) When learning: temporal credit assignment is hard (what caused later high or low rewards?)

Exploration

- Learning about the world by making decisions
  - Agent as scientist
  - Learn to ride a bike by trying (and falling)
  - Finding a key in Montezuma's revenge
- Censored data
  - Only get a reward (label) for decision made
  - Don't know what would have happened if had taken red pill instead of blue pill (Matrix movie reference)
- Decisions impact what learn about
  - If choose going to Stanford instead of going to MIT, will have different later experiences…

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

Generalization

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

→ Go Up

Input: Image

How many images are there? $(256^{100*200})^3$

Optimization

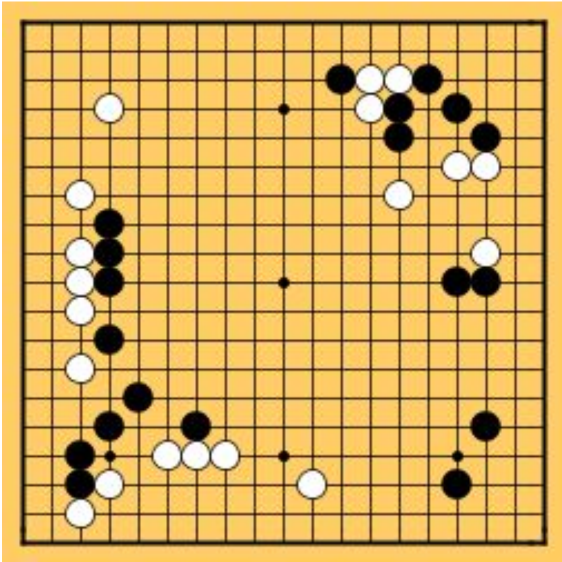
- Goal is to find an optimal policy
  - Yields highest expected rewards
- Or at least very good

Reinforcement Learning Involves

- Optimization
- Generalization
- Exploration
- Delayed consequences

AI Planning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences



- Computes good sequence of decisions
- But given model of how decisions impact world

Supervised Machine Learning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences

- Learns from experience
- But provided correct labels

Unsupervised Machine Learning (vs RL)

- Optimization
- Generalization
- Exploration
- Delayed consequences

- Learns from experience
- But no labels from world

Imitation Learning

- Optimization
- Generalization
- Exploration
- Delayed consequences

- Learns from experience… of others
- Assumes input demos of good policies

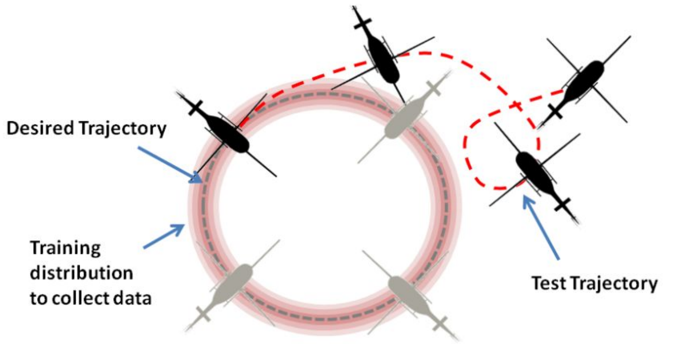
Imitation Learning

Abbeel, Coates and Ng helicopter team, Stanford

Imitation Learning

- Reduces RL to supervised learning
- Benefits
  - Great tools for supervised learning
  - Avoids exploration problem
  - With big data lots of data about outcomes of decisions
- Limitations
  - Can be expensive to capture
  - Limited by data collected

- Imitation learning + RL promisi



Ross & Bagnell 2013

How Do We Proceed?

- Explore the world
- Use experience to guide future decisions

Other issues

- Where do rewards come from?
  - And what happens if we get it wrong?
- Robustness / Risk sensitivity
- We are not alone…
  - Multi agent RL

Today's Plan

- Overview about reinforcement learning
- Course logistics
- Introduction/review of sequential decision making under uncertainty

Basic Logistics

- Instructor: Emma Brunskill
- TAs: Anthony Kim, Saied Mehdian, Barak Oshri, Shuhui Qu, Connie Zeng
- Time: MW 1:30-2:50pm
- Location: McMurtry Room 360
- Additional information
  - Course webpage: http://cs234.stanford.edu
  - Schedule, Piazza link, lecture slides, assignments…

Prerequisites

- Python proficiency
- Basic probability and statistics
- Multivariate calculus and linear algebra
- Machine learning (e.g. CS229 or CS221)
- The terms loss function, derivative, and gradient descent should be familiar
- Have heard of Markov decision processes and RL before in an AI or ML class
  - We will cover the basics, but quickly

Our Goal is that by the End of the Class You Will Be Able to:

- Define the key features of RL vs AI & other ML
- Define MDP, POMDP, bandit, batch offline RL, online RL
- Describe the exploration vs exploitation challenge and compare and contrast 2 or more approaches
- Given an application problem (e.g. from computer vision, robotics, etc) decide if it should be formulated as a RL problem, if yes how to formulate, what algorithm (from class) is best suited to addressing, and justify answer
- Implement several RL algorithms incl. a deep RL approach
- Describe multiple criteria for analyzing RL algorithms and evaluate algorithms on these metrics: e.g. regret, sample complexity, computational complexity, convergence, etc.
- List at least two open challenges or hot topics in RL

Grading

- 3 assignments
  - Basics of decision making and RL 10%
  - Generalization in RL and Deep RL 17%
  - Sample efficient RL 16%
- Midterm: 25%
- Final course project: 32%
  - 2-3 people (1 possible)
  - Includes milestone, presentation, writeup
  - Interacting with project mentor
- Final required poster session

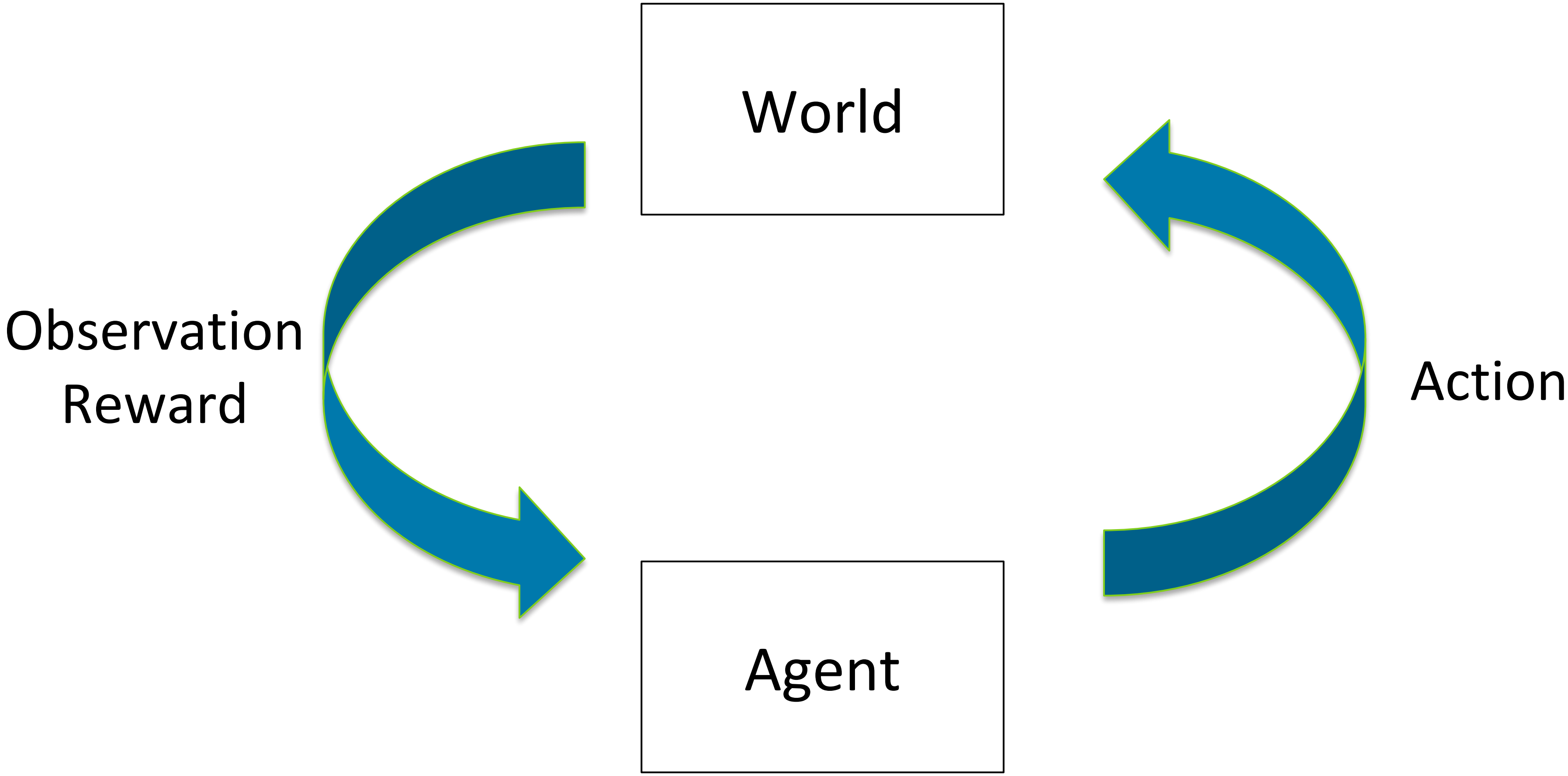
Grading

- Late policy
  - 6 free late days
  - See webpage for details on how many per assignment/project and penalty if use more
- Collaboration: see webpage and be sure clear on what is considered allowed collaboration

A Quick Poll: You:

1. Have heard of RL. That’s why you’re taking this class– to learn more!
2. Think a prior class mentioned something about Q-learning.
3. Do research in RL or deep RL. Make sure to cite my latest paper!

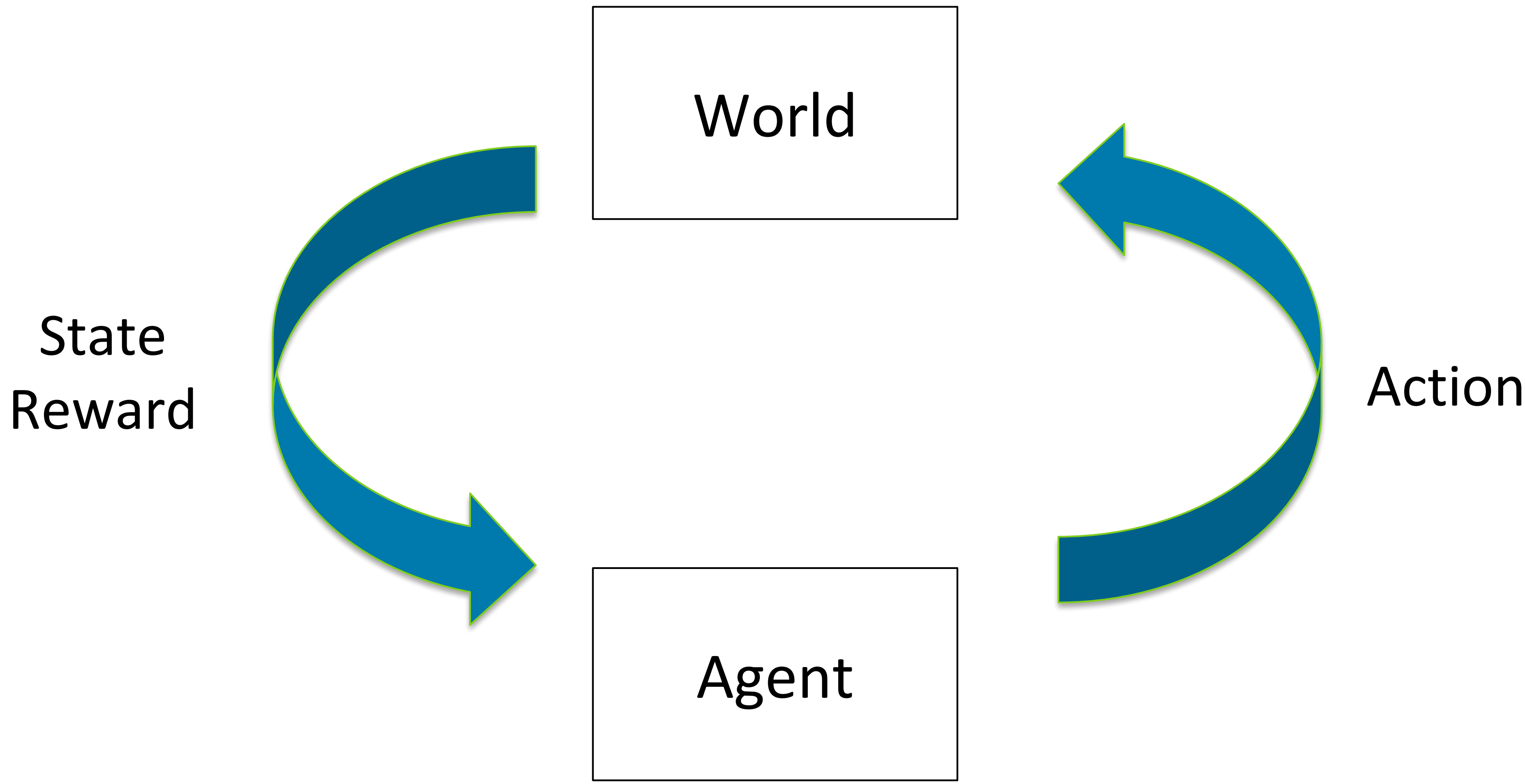
Decision Making Under Uncertainty



Markov Decision Process:

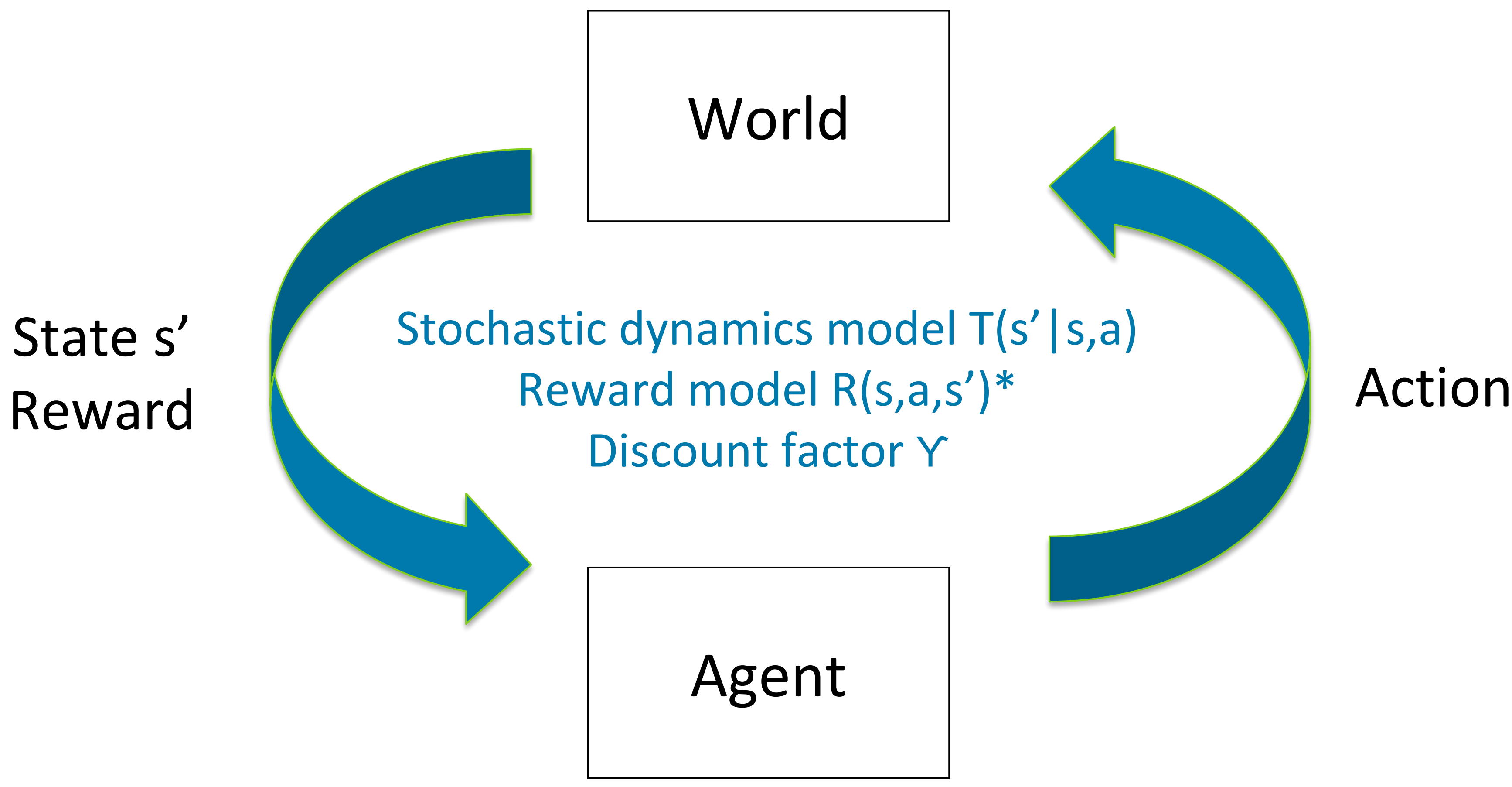
State = Observation

Sufficient statistic that captures how world behaves



Policy mapping from state $\longrightarrow$ action

Markov Decision Process: $<S,A,R,T,\gamma>$



Policy mapping from state $\rightarrow$ action

Markov Decision Process: <S,A,R,T,γ>

- S: set of states
- A: set of actions
- R: reward model R(s) / R(s,a) / R(s,a,s')
- T: dynamics model $p(s_{t+1} | s_t, a_t)$
- γ: discount factor

Markov Property

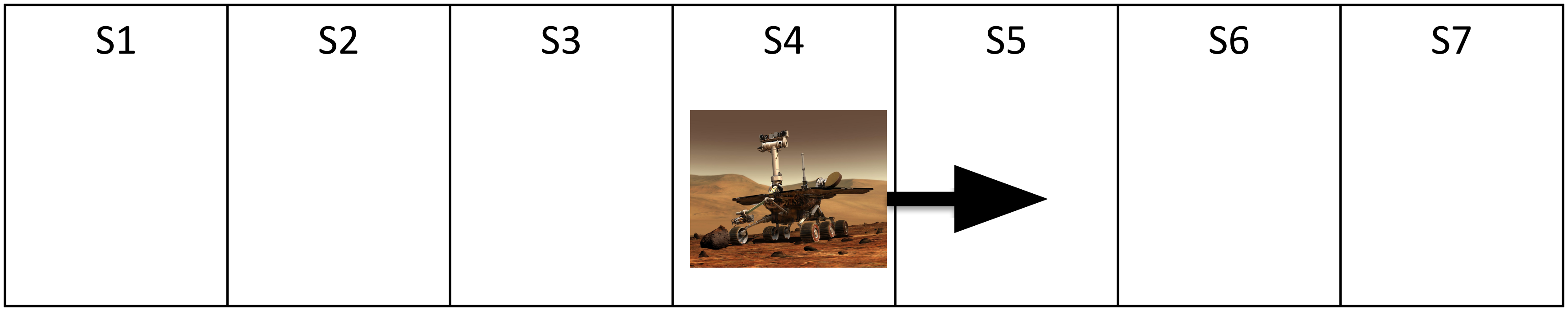
- Called **Markov** decision process because the outcome of an action depends only on the current state (vs entire history)
- $p(s_{t+1} | s_1, a_1, s_2, a_2, \ldots s_t, a_t) = p(s_{t+1} | s_t, a_t)$

- Why is this not too restrictive of an assumption?

MDP Policies

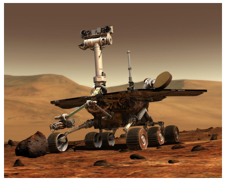
- Policy $\pi^*: S \rightarrow A$
  - Specifies what action to take in each state

Example: Simple Mars Rover



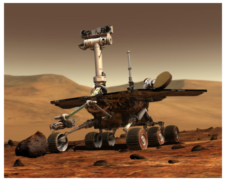
- 7 discrete states (location of rover)
- 2 actions: TryLeft or TryRight

Example: Simple Mars Rover

S1	S2	S3	S4	S5	S6	S7
Okay Field Site +1						Fantastic Field Site +10

- 7 discrete states (location of rover)
- 2 actions: TryLeft or TryRight
- Reward
  - +1 in state S1
  - +10 in state S7
  - 0 otherwise

How Many Deterministic Policies?

S1	S2	S3	S4	S5	S6	S7
Okay Field Site +1						Fantastic Field Site +10

- 7 discrete states (location of rover)
- 2 actions: TryLeft or TryRight
- Reward
  - +1 in state S1
  - +10 in state S7
  - 0 otherwise

How Good is a Policy?

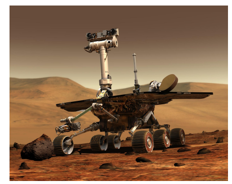
- For a given state s
- Value of policy $V^{\pi}(s)$: Expected discounted sum of rewards obtain if follow policy π starting in state s

$$V^{\pi}(s) = E_T\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \pi(s_i)) | s_0 = s\right]$$

- Optimal policy: $\mathrm{argmax}_{\pi}\ V^{\pi}(s)$

Discounting

$$V^{\pi}(s) = E_T\left[\sum_{i=0}^{\infty}\gamma^i r(s_i, \pi(s_i))|s_0 = s\right]$$

S1	S2	S3	S4	S5	S6	S7
Okay Field Site +1						Fantastic Field Site +10

- 7 discrete states (location of rover)
- 2 actions: TryLeft or TryRight
  - Deterministic: Succeeds unless hit edge, then stay
- Reward: For all actions:
  - +1 in state S1, +10 in state S7, 0 otherwise

1. If γ=1 what is optimal policy in each state?
2. If γ=0.1 what is optimal policy in each state?
3. Find γ which makes TryLeft or TryRight of equal value in s4

MDP Policy Value

- Value of policy $V^{\pi}(s)$: Expected discounted sum of rewards obtain if follow policy π starting in state s

$$V^{\pi}(s) = E_T\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \pi(s_i)) | s_0 = s\right]$$

- Due to Markov property can decompose

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|\pi(s), s) V^{\pi}(s')$$

Immediate reward

Discounted sum of future rewards

Q: State-Action Value

- Expected immediate reward for taking action a
- And expected future reward get after taking that action from that state and following π

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V^{\pi}(s')$$

Optimal Value, Q & Policy

- Optimal V: highest possible value for each s (under any possible policy)
- Satisfies the Bellman Equation

$$V^*(s) = \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V^*(s') \right]$$

- Optimal Q function:

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V^*(s')$$

- Optimal policy

$$\pi^*(s,a) = \arg\max_a Q^*(s,a)$$

MDP Planning

- How to compute π^*?
- Know full MDP
  - Given the dynamics and reward model
  - Computational challenge, not learning

Value Iteration

- Bellman equation inspires an update rule
- First compute value for each state as if only get to take 1 action
- Then bootstrap for what to do if take 2 actions…

Value Iteration (VI)

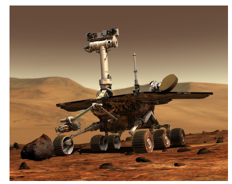
1. Initialize $V_0(s_i)$=0 for all states $s_{i,}$
2. Set k=1
3. Loop until [finite horizon, convergence]
   - For each state s,

$$V_{k+1}(s) = \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V_k(s') \right]$$

4. Extract Policy

Bellman backup

$$V_{k+1}(s) = \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V_k(s') \right]$$

S1	S2	S3	S4	S5	S6	S7
Okay Field Site +1						Fantastic Field Site +10

- 7 discrete states (location of rover)
- 2 actions: TryLeft or TryRight
  - Deterministic: Succeeds unless hit edge, then stay
- Reward: For all actions:
  - +1 in state S1, +10 in state S7, 0 otherwise

1. If Υ=1 what is value of each state?
2. If Υ=0.1 what is the value of each state?

Computational Complexity: Value Iteration (VI)

1. Initialize $V_0(s_i)=0$ for all states s_i,
2. Set k=1
3. Loop until [finite horizon, convergence]
   - For each state s,

$$V_{k+1}(s) = \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V_k(s') \right]$$

4. Extract Policy

Will Value Iteration Converge?

Contraction Operator

- Let O be an operator
- If |OV – OV'| <= |V-V|'
- Then O is a contraction operator

- Let B be the Bellman backup operator

$$\begin{aligned} V_{k+1}(s) &= \max_a \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|a,s) V_k(s') \right] \\ &= BV_k \end{aligned}$$

Will Value Iteration Converge?

- Yes, if discount factor $\gamma < 1$ or end up in a terminal state with probability 1
- Bellman backup is a contraction if discount factor, $\gamma < 1$
- If apply it to two different value functions, distance between value functions shrinks after apply Bellman equation to each

Bellman Backup is a Contraction on V (γ<1)

|| V-V'|| = Infinity norm (find max difference over all states, e.g. max(s) |V(s) – V'(s) |

$$
\begin{aligned}
\left\|BV-BV'\right\| &= \left\| \max_{a}\left[R(s,a)+\gamma\sum_{s_j\in S} p(s_j \mid s_i,a)V(s_j)\right] - \max_{a'}\left[R(s,a')-\gamma\sum_{s_j\in S} p(s_j \mid s_i,a')V'(s_j)\right]\right\| \\
&\le \max_{a}\left\|\left[R(s,a)+\gamma\sum_{s_j\in S} p(s_j \mid s_i,a)V(s_j) - R(s,a)+\gamma\sum_{s_j\in S} p(s_j \mid s_i,a)V'(s_j)\right]\right\| \\
&\le \gamma\max_{a}\left\|\left[\sum_{s_j\in S} p(s_j \mid s_i,a)V(s_j) - \sum_{s_j\in S} p(s_j \mid s_i,a)V'(s_j)\right]\right\| \\
&= \gamma\max_{a}\left\|\left[\sum_{s_j\in S} p(s_j \mid s_i,a)(V(s_j) - V'(s_j))\right]\right\| \\
&\le \gamma\max_{a,s_i}\sum_{s_j\in S} p(s_j \mid s_i,a)\left|V(s_j)-V'(s_j)\right| \\
&\le \gamma\max_{a,s_i}\sum_{s_j\in S} p(s_j \mid s_i,a)\left\|V-V'\right\| \\
&= \gamma\left\|V-V'\right\|
\end{aligned}
$$

Properties of Bellman Operator (γ<1)

- Only has 1 fixed point (the point reach if apply a contraction operator many times)
  - If had two, then would not get closer when operator, violating bound derived on prior slide
- When apply contraction function to any argument, value must get closer to fixed point
  - Fixed point doesn't move
  - Repeated operator applications yield fixed point

Check Understanding

- Prove value iteration converges to a unique solution for discrete state and action space and $\gamma<1$
- Does the initialization of values in value iteration impact anything?

Summary

- Overview about reinforcement learning
- Course logistics
- Introduction/review of sequential decision making under uncertainty