Lecture 1: Introduction to RL

Professor Emma Brunskill

CS234 RL

Spring 2024

- Today the 3rd part of the lecture includes some slides from David Silver's introduction to RL slides or modifications of those slides

Today's Plan

- Overview of reinforcement learning
- Course logistics
- Introduction to sequential decision making under uncertainty

Reinforcement Learning

Learning through experience/data to make good decisions under uncertainty

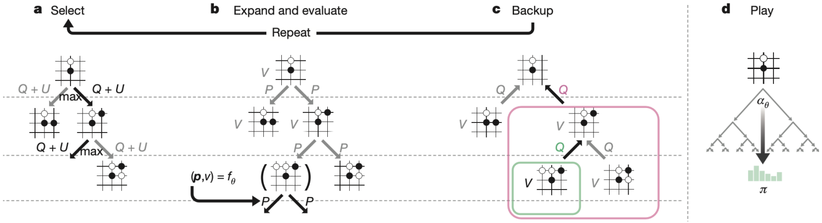
Reinforcement Learning

- Learning through experience/data to make good decisions under uncertainty
- Essential part of intelligence
- Builds strongly from theory and ideas starting in the 1950s with Richard Bellman

Reinforcement Learning

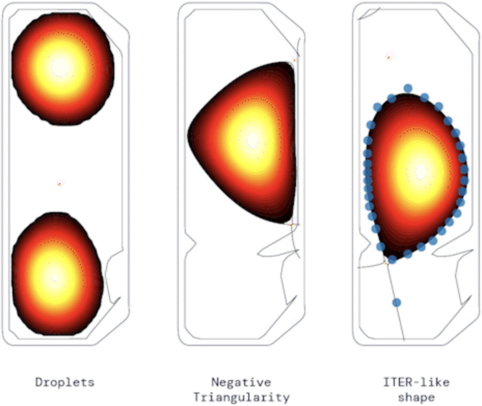
- Learning through experience/data to make good decisions under uncertainty
- Essential part of intelligence
- Builds strongly from theory and ideas starting in the 1950s with Richard Bellman
- A number of impressive successes in the last decade

Beyond Human Performance on the Board Game Go[1]



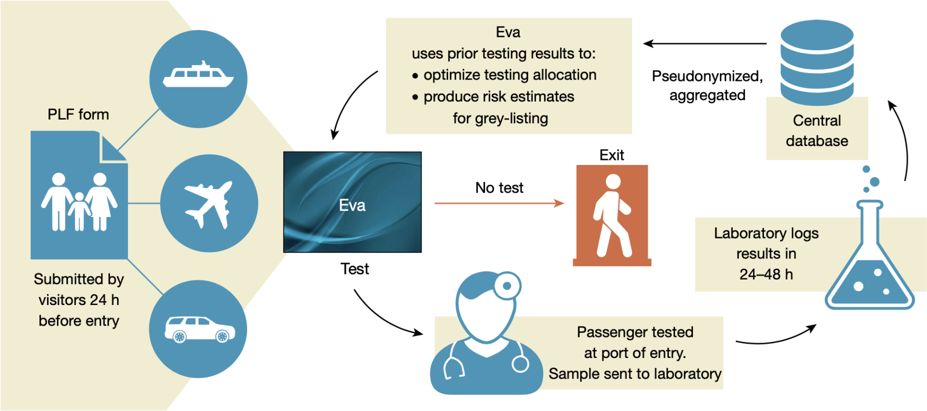
[1] Image credits: Silver et al. Nature 2017
https://www.nature.com/articles/nature24270

Learning Plasma Control for Fusion Science[2]



[2]Image credits: left Alain Herzog / EPFL, right DeepMind & SPC/EPFL. Degrave et al. Nature 2022 https://www.nature.com/articles/s41586-021-04301-9

Efficient and targeted COVID-19 border testing via RL [3]



[3] Bastani et al. Nature 2021
https://www.nature.com/articles/s41586-021-04014-z

ChatGPT (https://openai.com/blog/chatgpt/)

behavior cloning
imitation learning

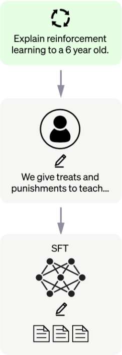
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



model of a reward
model based RL

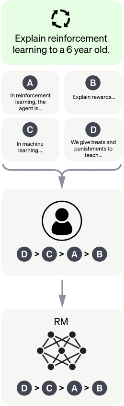
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



reinforcement learning
RLHF

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

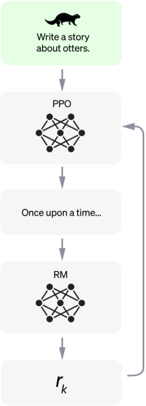
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

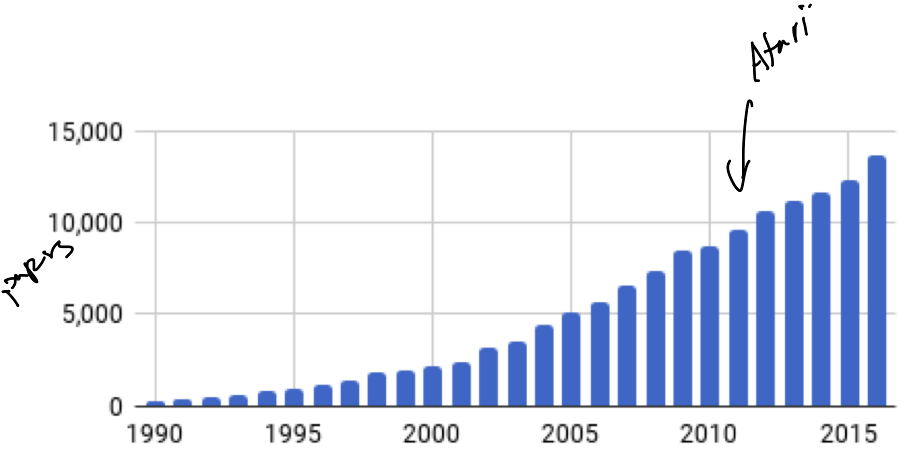
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

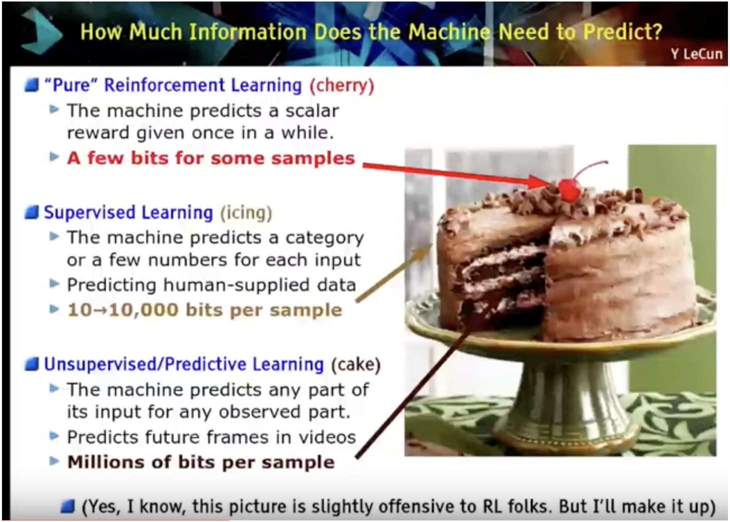


Huge Increase in Interest[4]



[4] Figure from Henderson et al. 2018 AAAI https://arxiv.org/pdf/1709.06560.pdf

But Also Contentious[5]



[5] https://www.youtube.com/watch?v=0unt2Y4qxQo

Lecture 1 Poll 1: Why Do You Want to Take This Class?

- Go to first link in Ed or http://PollEv.com/emmabrunskil381 (note only 1 "l")
- Skip registration
- Refresh if it is hanging
- Enter in your sunid as your screen id (bear with us– we will iron out these issues for Wed)
- Enter your answer!

Reinforcement Learning (Generally) Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

Optimization

- Goal is to find an optimal way to make decisions
  - Yielding best outcomes or at least very good outcomes
- Explicit notion of decision utility
- Example: finding minimum distance route between two cities given network of roads

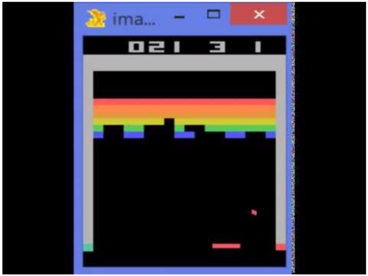
Delayed Consequences

- Decisions now can impact things much later...
  - Saving for retirement
  - Finding a key in video game Montezuma's revenge
- Introduces two challenges
  - When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer term ramifications
  - 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 failing)
- Decisions impact what we learn about
  - Only get a reward for decision made
  - Don't know what would have happened for other decision
  - If we choose to go to Stanford instead of MIT, we will have different later experiences...

- Policy is mapping from past experience to action $256^{300 \times 400}$
- Why not just pre-program a policy? 300



400

Figure: DeepMind Nature, 2015

RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	✓			✓	
Learns from experience				✓	
Generalization	✓			✓	
Delayed Consequences	✓			✓	
Exploration				✓	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning

RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X				
Learns from experience		✓			
Generalization	X	✓			
Delayed Consequences	X				
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- AI planning assumes have a model of how decisions impact environment

RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X				
Learns from experience		X	✓		
Generalization	X	X	✓		
Delayed Consequences	X				
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Supervised learning has access to the correct labels

RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X			✓	
Learns from experience		X	X	✓	
Generalization	X	X	X	✓	
Delayed Consequences	X			✓	
Exploration				✓	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Unsupervised learning has access to no labels

Imitation Learning

	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Imitation learning typically assumes input demonstrations of good policies
- IL reduces RL to SL. **For many good reasons, IL is very popular.**

Two Problem Categories Where RL is Particularly Powerful

1. No examples of desired behavior: e.g. because the goal is to go beyond human performance or there is no existing data for a task.
2. Enormous search or optimization problem with delayed outcomes:

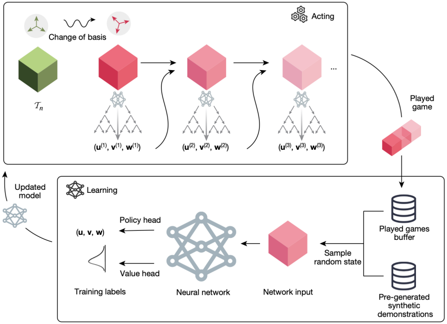


Figure: AlphaTensor. Fawzi et al. 2022

Today's Plan

- Overview of reinforcement learning
- **Course logistics**
- Introduction to sequential decision making under uncertainty

Course Outline

- Markov decision processes & planning
- Model-free policy evaluation
- Model-free control
- Policy Search
- Offline RL **including RL from Human Feedback and Direct Preference Optimization**
- Exploration
- Advanced Topics

MCTS

High Level Learning Goals[6]

- Define the key features of RL
- Given an application problem know how (and whether) to use RL for it
- Implement (in code) common RL algorithms
- Understand theoretical and empirical approaches for evaluating the quality of a RL algorithm

[6] For more detailed descriptions, see website

Course Structure Overview

- Live lectures
- Three homeworks
- 1 exam
- 1 multiple choice quiz
- Final project
- Check/Refresh your understanding exercises (Access through your Stanford poll everywhere account)
- Problem sessions (optional)

"Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC"[7]

- In a psychology Massive Open Online Class, doing more activities seemed to yield a **6 times larger** learning benefit compared to extra video watching or reading
- "...it appears students actually spend substantially less time per activity (3.4 min) than reading a page (5.0 min)"

[7] Koedinger et al. L@S 2015. https://dl.acm.org/doi/pdf/10.1145/2724660.2724681

"Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC"[8]

- In a psychology Massive Open Online Class, doing more activities seemed to yield a 6 times larger learning benefit compared to extra video watching or reading
- "...it appears students actually spend substantially less time per activity (3.4 min) than reading a page (5.0 min)"
- → **Engaged practice is likely to be a more efficient and effective way to learn material.**
- To achieve the class learning goals, I encourage you to: do homework, attend problem sessions, do the check your understandings, and try past quiz or exam problems for practice without referring to solutions before you complete them

[8] Koedinger et al. L@S 2015. https://dl.acm.org/doi/pdf/10.1145/2724660.2724681

Staff and Support

- Instructor: Emma Brunskill
- CAs: Dilip Arumugam, Joao Araujo, Saurabh Kumar, Jonathan Lee, Garrett Thomas
- Additional information
  - Course webpage: http://cs234.stanford.edu
  - Schedule, Ed (fastest way to get help), lecture slides
  - Prerequisites, grading details, late policy, see webpage
  - Office hour schedule will be announced by the end of today
- All of you can succeed if you put in the effort
- We, the class staff, and your fellow classmates, are here to help!

Today's Plan

- Overview of reinforcement learning
- Course logistics
- **Introduction to sequential decision making under uncertainty**

Refresher Exercise: AI Tutor as a Decision Process

- Student initially does not know either addition (easier) nor subtraction (harder)
- AI tutor agent can provide practice problems about addition or subtraction
- AI agent gets rewarded $+1$ if student gets problem right, -1 if get problem wrong
- Model this as a Decision Process. Define state space, action space, and reward model. What does the dynamics model represent? What would a policy to optimize the expected discounted sum of rewards yield?
- Write down your own answers (5 min) and then discuss in small breakout groups..

Lecture 1 Poll 2: Refresher Exercise: AI Tutor as a Decision Process

- Student initially does not know either addition (easier) nor subtraction (harder)
- AI tutor agent can provide practice problems about addition or subtraction
- AI agent gets rewarded +1 if student gets problem right, -1 if get problem wrong
- Model this as a Decision Process. Define state space, action space, and reward model.
- What does the dynamics model represent? What would a policy to optimize the expected discounted sum of rewards yield?
- Go to **2nd** link in Ed or `http://PollEv.com/emmabrunskil381` (note only 1 "l")
- If you haven't already: skip registration, refresh if it is hanging , enter in your sunid as your screen id (bear with us– we will iron out these issues for Wed)
- Enter your answer!

Refresher Exercise: AI Tutor as a Decision Process

history (observ, question, reward...)

how good student is at add & subtr
(.9, .4)

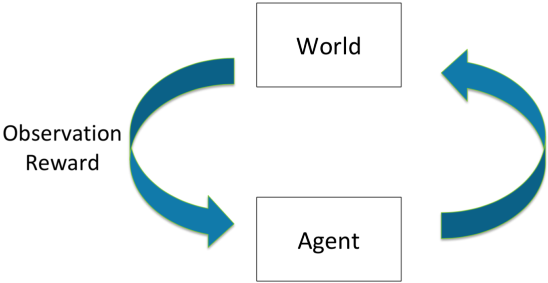
- State:
- Actions: addition question or subt
- Reward model: +1 if student gets right
- Meaning of dynamics model:

agent max its reward should only
give easy questions

Refresher Exercise: AI Tutor as a Decision Process

- Student initially does not know either addition (easier) nor subtraction (harder)
- Teaching agent can provide activities about addition or subtraction
- Agent gets rewarded for student performance: $+1$ if student gets problem right, -1 if get problem wrong
- Which items will agent learn to give to max expected reward? Is this the best way to optimize for learning? If not, what other reward might one give to encourage learning?

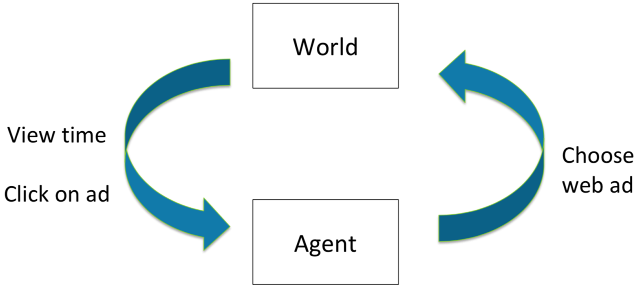
Sequential Decision Making



decisions

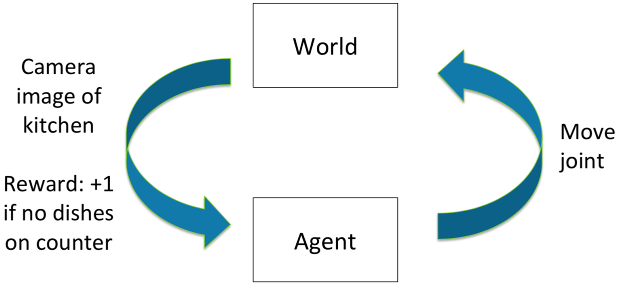
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

Example: Web Advertising



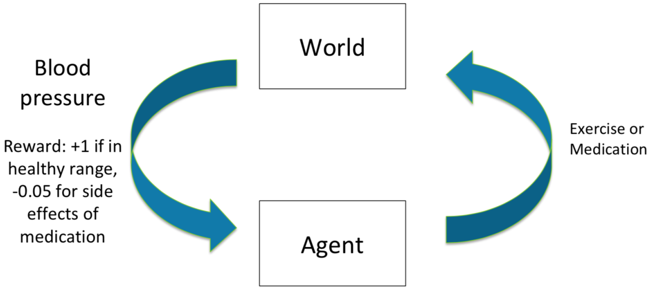
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

Example: Robot Unloading Dishwasher



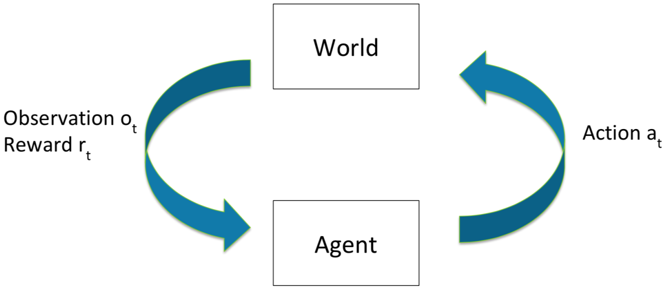
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

Example: Blood Pressure Control



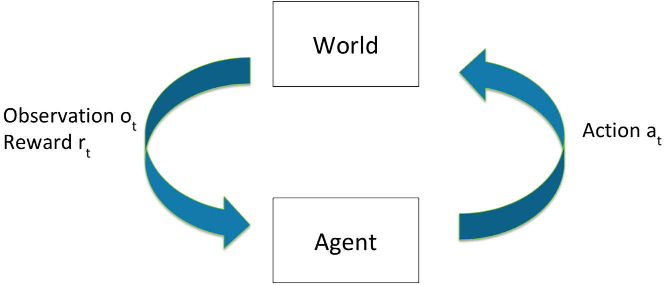
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

Sequential Decision Process: Agent & the World (Discrete Time)



- Each time step t:
  - Agent takes an action a_t
  - World updates given action a_t, emits observation o_t and reward r_t
  - Agent receives observation o_t and reward r_t

History: Sequence of Past Observations, Actions & Rewards



- History $h_t = (a_1, o_1, r_1, \ldots, a_t, o_t, r_t)$
- Agent chooses action based on history
- State is information assumed to determine what happens next
  - Function of history: $s_t = (h_t)$

Markov Assumption

- Information state: sufficient statistic of history
- State s_t is Markov if and only if:

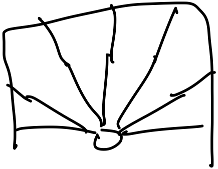
$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

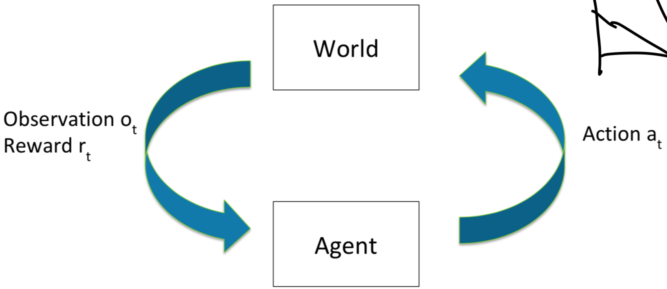
- Future is independent of past given present

Why is Markov Assumption Popular?

- Simple and often can be satisfied if include some history as part of the state
- In practice often assume most recent observation is sufficient statistic of history: $s_t = o_t$
- State representation has big implications for:
  - Computational complexity
  - Data required
  - Resulting performance

Types of Sequential Decision Processes





- Is state Markov? Is world partially observable? (POMDP)
- Are dynamics deterministic or stochastic?
- Do actions influence only immediate reward (bandits) or reward and next state ?

Example: Mars Rover as a Markov Decision Process

s_1	s_2	s_3	s_4	s_5	s_6	s_7
+1						+10

Figure: Mars rover image: NASA/JPL-Caltech

- States: Location of rover $(s_1, \ldots, s_7)$
- Actions: TryLeft or TryRight
- Rewards:
  - $+1$ in state s_1
  - $+10$ in state s_7
  - 0 in all other states

MDP Model

- Agent's representation of how world changes given agent's action
- Transition / dynamics model predicts next agent state

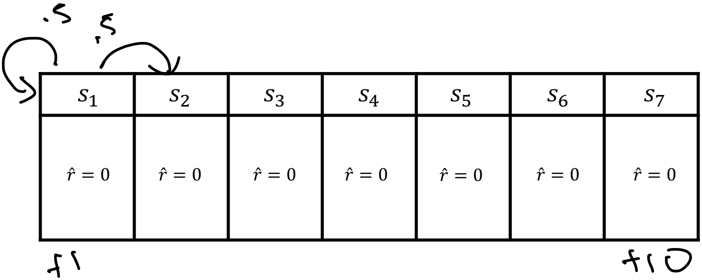
$$p(s_{t+1} = s' | s_t = s, a_t = a)$$

- Reward model predicts immediate reward

$$r(s_t = s, a_t = a) = \mathbb{E}[r_t | s_t = s, a_t = a]$$

$r(s)$
$r(s, a)$
$r(s, a, s')$

Example: Mars Rover Stochastic Markov Model



- Numbers above show RL agent's reward model
- Part of agent's transition model:
  - $0.5 = P(s_1|s_1, \text{TryRight}) = P(s_2|s_1, \text{TryRight})$
  - $0.5 = P(s_2|s_2, \text{TryRight}) = P(s_3|s_2, \text{TryRight}) \cdots$
- Model may be wrong

Policy

- Policy π determines how the agent chooses actions
- $\pi : S \rightarrow A$, mapping from states to actions
- Deterministic policy:

$$\pi(s) = a$$

- Stochastic policy:

$$\pi(a|s) = Pr(a_t = a|s_t = s)$$

Example: Mars Rover Policy

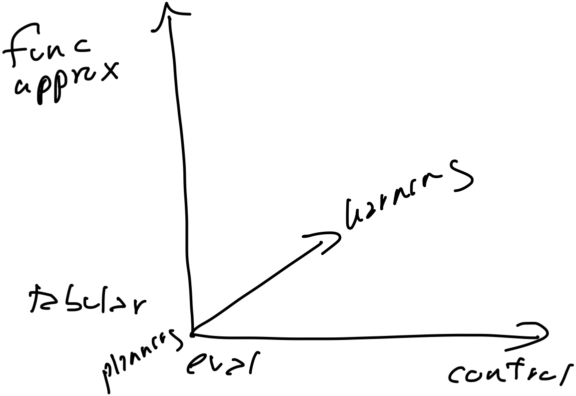
s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

- $\pi(s_1) = \pi(s_2) = \cdots = \pi(s_7) = \text{TryRight}$
- Quick check your understanding: is this a deterministic policy or a stochastic policy?

Evaluation and Control

- Evaluation
  - Estimate/predict the expected rewards from following a given policy
- Control
  - Optimization: find the best policy

Build Up in Complexity



Making Sequences of Good Decisions Given a Model of the World

- Assume finite set of states and actions
- Given models of the world (dynamics and reward)
- Evaluate the performance of a particular decision policy
- Compute the best policy
- This can be viewed as an AI planning problem

Making Sequences of Good Decisions Given a Model of the World

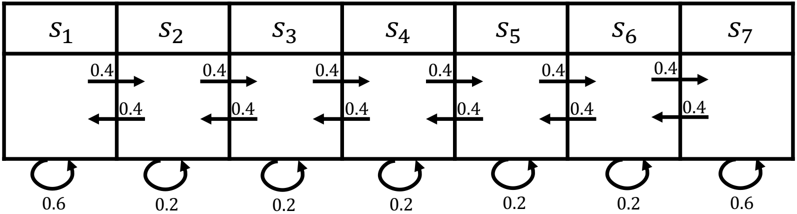
- Markov Processes
- Markov Reward Processes (MRPs)
- Markov Decision Processes (MDPs)
- Evaluation and Control in MDPs

Markov Process or Markov Chain

- Memoryless random process
  - Sequence of random states with Markov property
- Definition of Markov Process
  - S is a (finite) set of states ($s \in S$)
  - P is dynamics/transition model that specifices $p(s_{t+1} = s' | s_t = s)$
- Note: no rewards, no actions
- If finite number (N) of states, can express P as a matrix

$$P = \begin{pmatrix} P(s_1|s_1) & P(s_2|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & P(s_2|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1|s_N) & P(s_2|s_N) & \cdots & P(s_N|s_N) \end{pmatrix}$$

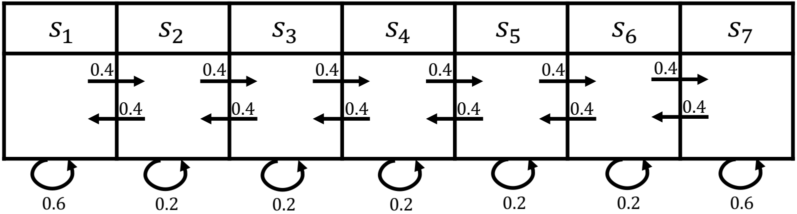
Example: Mars Rover Markov Chain Transition Matrix, P



$$P = \begin{pmatrix} 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.2 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.2 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.2 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \end{pmatrix}$$

rows sum to 1

Example: Mars Rover Markov Chain Episodes



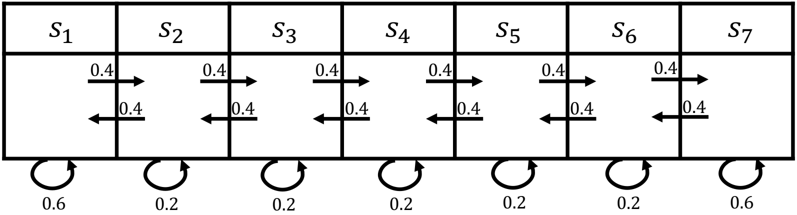
Example: Sample episodes starting from S4

- $s_4, s_5, s_6, s_7, s_7, s_7, \ldots$
- $s_4, s_4, s_5, s_4, s_5, s_6, \ldots$
- $s_4, s_3, s_2, s_1, \ldots$

Markov Reward Process (MRP)

- Markov Reward Process is a Markov Chain + rewards
- Definition of Markov Reward Process (MRP)
  - S is a (finite) set of states ($s \in S$)
  - P is dynamics/transition model that specifices $P(s_{t+1} = s' | s_t = s)$
  - R is a reward function $R(s_t = s) = \mathbb{E}[r_t | s_t = s]$
  - Discount factor $\gamma \in [0, 1]$
- Note: no actions
- If finite number (N) of states, can express R as a vector

Example: Mars Rover Markov Reward Process



+1 +10

- Reward: $+1$ in s_1, $+10$ in s_7, 0 in all other states

Return & Value Function

- Definition of Horizon (H)
  - Number of time steps in each episode
  - Can be infinite
  - Otherwise called **finite** Markov reward process
- Definition of Return, G_t (for a Markov Reward Process)
  - Discounted sum of rewards from time step t to horizon H

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{H-1} r_{t+H-1}$$

- Definition of State Value Function, $V(s)$ (for a Markov Reward Process)
  - Expected return from starting in state s

$$V(s) = \mathbb{E}[G_t | s_t = s] = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{H-1} r_{t+H-1} | s_t = s]$$

Discount Factor

- Mathematically convenient (avoid infinite returns and values)
- Humans often act as if there's a discount factor < 1
- If episode lengths are always finite ($H < \infty$), can use $\gamma = 1$

Discount Factor

- Mathematically convenient (avoid infinite returns and values)
- Humans often act as if there's a discount factor < 1
- $\gamma = 0$: Only care about immediate reward
- $\gamma = 1$: Future reward is as beneficial as immediate reward
- If episode lengths are always finite ($H < \infty$), can use $\gamma = 1$