DS 598
Introduction to RL

Xuezhou Zhang

Announcements

- Homework 1 is due this Friday (submit on blackboard).

- Course project is posted.
- **First task**: Make sure it installs correctly on your computer.

- Pytorch tutorial + project environment installation Helpdesk in the discussion section next week.

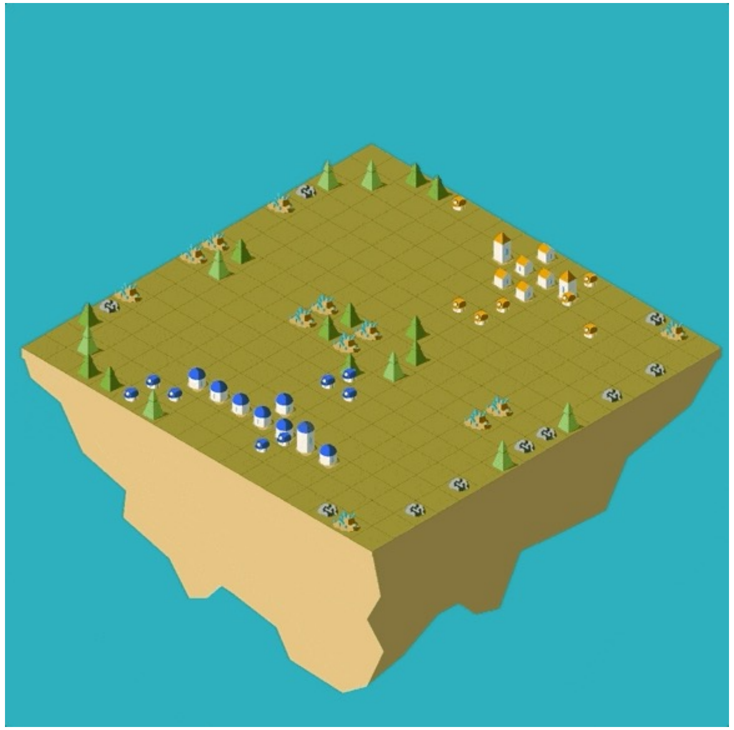
Team Signup So far

- 13 Teams
- 37/39 people

	Team 1	Team Members			Team 2	Team Members		
03/19	**Team Zero**	Mao Mao	Haotian Shangguan	Ziye Chen	**Team Go**	Zijian Guo	Yichen Song	
03/21	**Team RL**	Seunghwan Hyun	Osama Dabbousi	Zou(Zoey) Yang	**Team Carbon**	Bargav Jagatha	Akshat G	Mounika
03/26	**Team Alpha**	Ayush Sharma	Gauravdeep Singh Bindra		**Team Star**	Zhengyang Shan	Yi Liu	Jasmine Pham
03/28	**Team Gamma**	Wai Yuen Cheng	Andy Yang	Tariq Georges	**Team Best**	Minfeng Qian	Han Li	Qiji Zheng
04/02	**Team S**	Sahana Kowshik	Srishti Jain	Ruoxy Jin	**Team Terrier Threat**	Jack Campbell!	Carmen Pelayo!	Chenjia Li!
04/04					**Team Rocket**	Tejaswini S	Shreyas S	Abhaya Shukla
04/09	**Team Q**	Xavier Thomas	Shivacharan oruganti			YuCheng		
04/11	**Team ZGL**	Jasmine Dong	Yu Liang	Shuhan Wang				

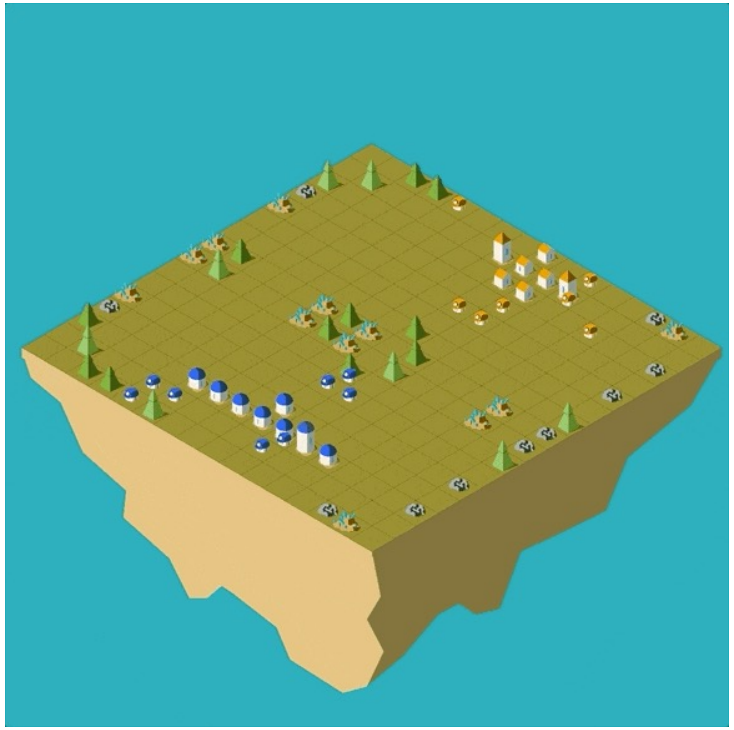
Project Description

- Goal: build more cities than your opponents!
- Several components: map, resources, units



Project Description: Map

- L x L grid.
- $L \in [12,16,24,32]$
- Resource positions randomly generated.



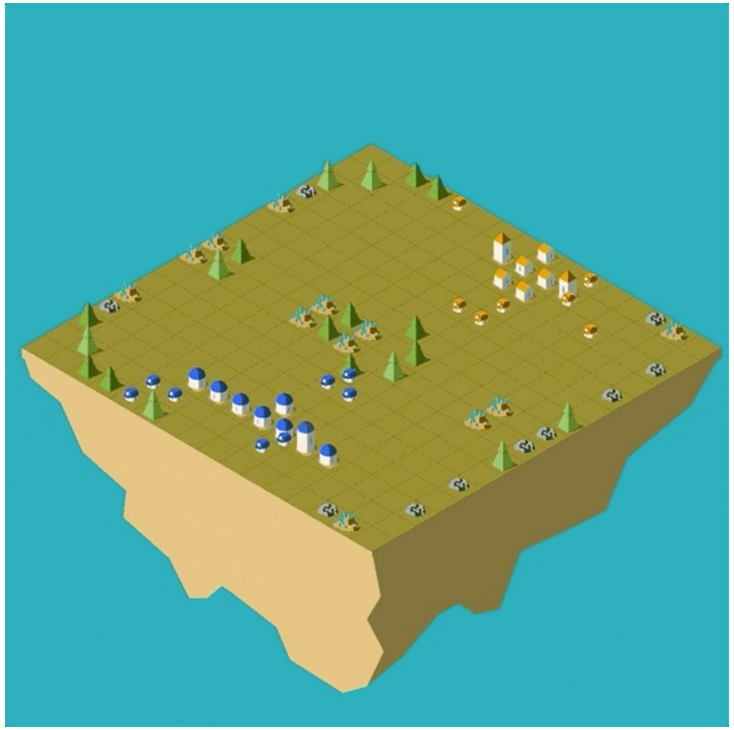
Project Description: Resources

- Wood (can regrow), Coal, Uranium

Resource Type	Research Points Pre-requisite	Fuel Value per Unit	Units Collected per Turn
Wood	0	1	20
Coal	50	10	5
Uranium	200	40	2

Project Description: Actionable Units

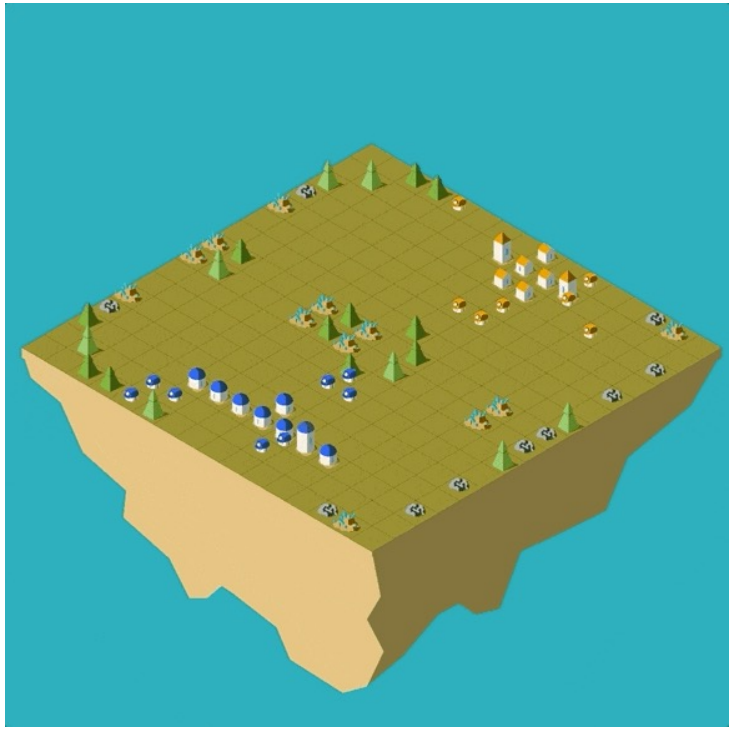
- Workers
  - Move
  - Pillage - Reduce the Road level
  - Transfer - Send resource to an adjacent Unit
  - Build City
- Carts:
  - Move
  - Transfer
- Cities:
  - Build Worker
  - Build Cart
  - Research



Project Description: Cooldown

- Every unit has a cooldown after action.
- Units on roads recover faster.

Unit Type	Base Cooldown
CityTile	10
Worker	2
Cart	3



Project Description: Day/Night Cycle

- Day: 30 turns   Night: 10 turns
- Total: 360 turns = 9 days/nights
- Units burn fuels to survive the night.

Unit	Fuel Burn in City	Fuel Burn Outside City
CityTile	23 - 5 * number of adjacent friendly CityTiles	N/A
Cart	0	10
Worker	0	4

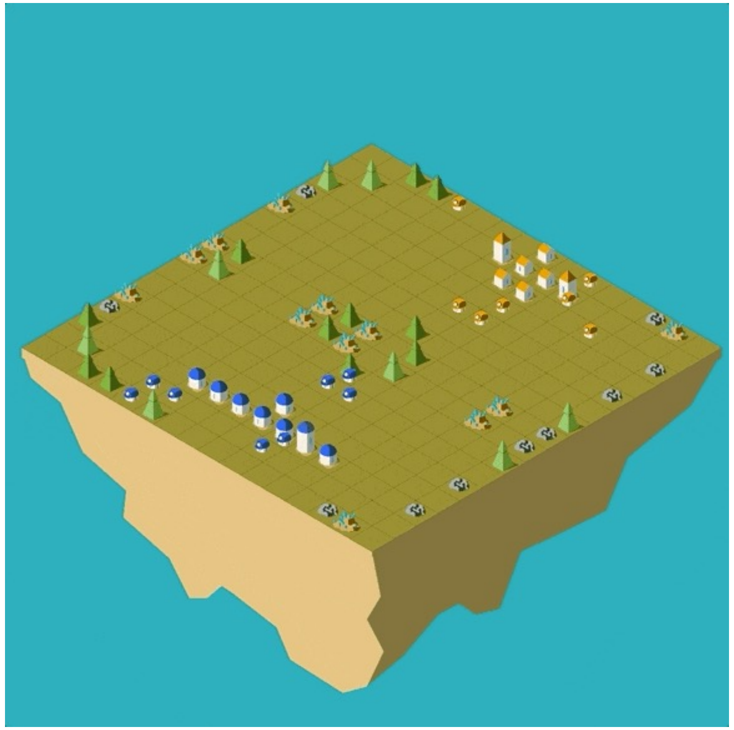
Finite Horizon MDPs

- Environment resets $s_0 \sim \mu(\cdot)$.
- For step $h = 0, \dots, H-1$
  - Agent perform action $a_h \sim \pi(\cdot|s_h)$.
  - Environment provide $s_{h+1} \sim P_h(\cdot|s_h, a_h), r_h = R_h(s_h, a_h)$.
- Q function: $Q_H(s,a) = 0, Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h) | \pi\right]$
- Bellman Equation: $Q_h^\pi(s,a) = R_h(s, \pi(s)) + \mathbb{E}_{s' \sim P_h(\cdot|s,\pi(s))}\left[Q_{h+1}^\pi(s', \pi(s'))\right]$

Finite Horizon MDPs

- Everything now depends on the time step h!

- Your strategy will differ at start vs. end of the game.



Challenges

1. Multi-agent
2. Large and Dynamical State/Action Space
3. Long Horizon

Ablation Study

Agent	**Gamer Median**	**Gamer Mean**	**Record Mean**	**Clipped Record Mean**
DreamerV2	1.64	11.33	0.36	0.25
No Layer Norm	1.66	5.95	0.38	0.25
No Reward Gradients	1.68	6.18	0.37	0.24
No Discrete Latents	1.08	3.71	0.24	0.19
No KL Balancing	0.84	3.49	0.19	0.16
No Policy Reinforce	0.69	2.74	0.16	0.15
No Image Gradients	0.04	0.31	0.01	0.01

Proper Acknowledgements

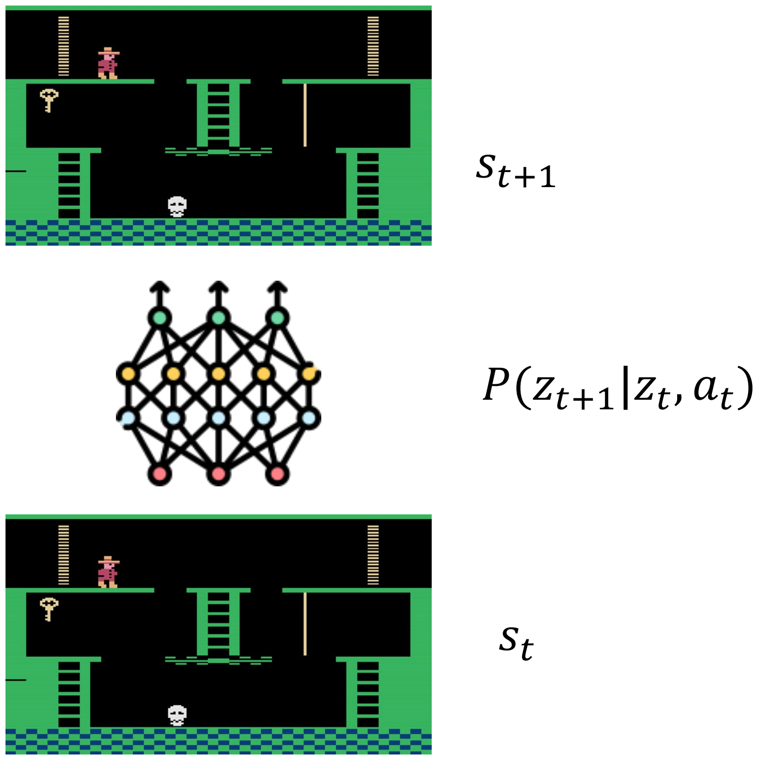
- You can use any resources that you can find online, given that you cite them properly in your presentation as well as your final reports.

Chapter 4: Value-based RL

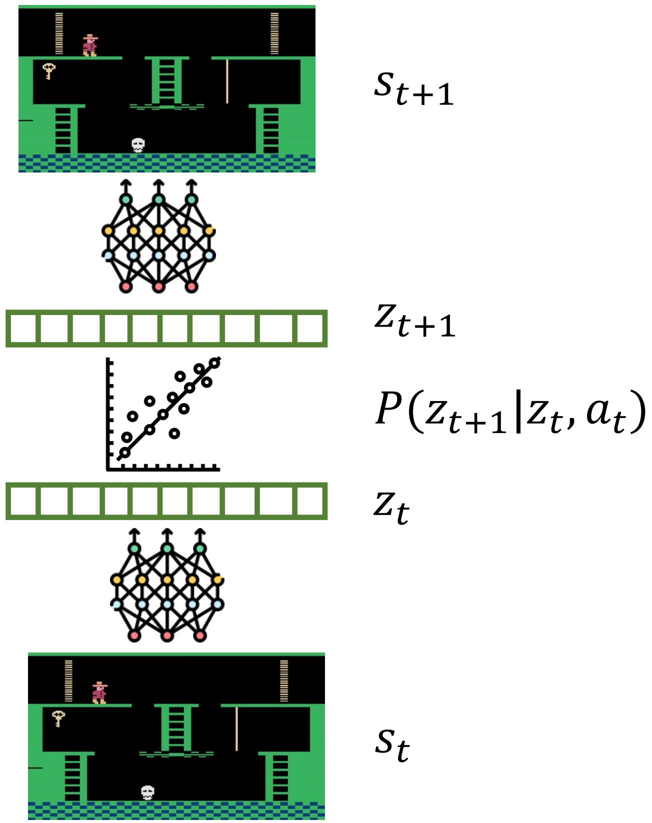
Recap: Online Reinforcement Learning

- Start by knowing nothing about the environment.
- Gather information while interacting with the environment.
- Gather reward / suffer costs along the way.


Last time: Model-based RL



A Naïve model is difficult to learn



Latent model: Dreamer, MuZero

However..

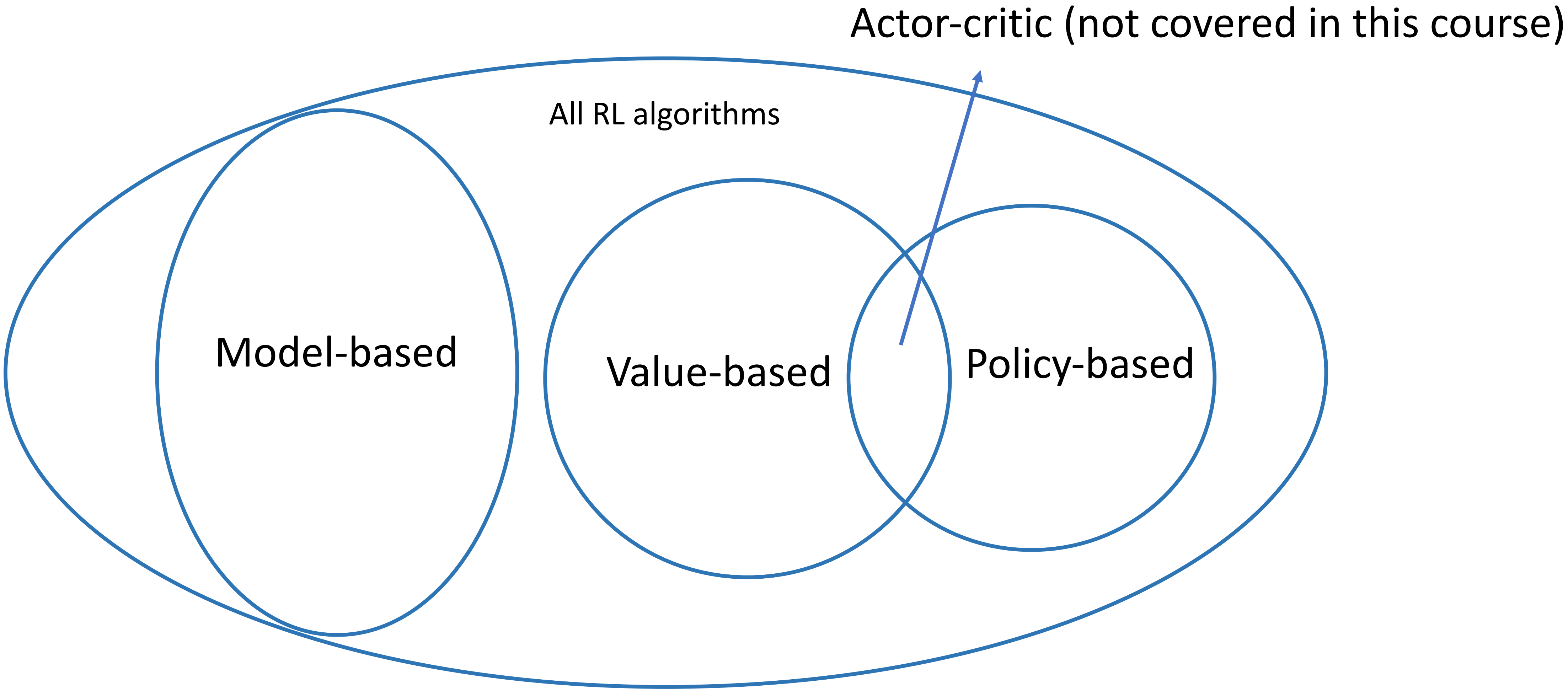
- Even with a good model, planning is still difficult.

🤔 Can we bypass learning the model at all?

Model-based vs. model-free RL

- **Model-free RL**: algorithms that avoid explicitly learning the transition model.

The RL Ontology



Value-based RL

- Estimate the Q^* function directly from data.

- Why the Q^* function?

- With a finite action space, one can make decisions directly from the Q^* function.

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

Solve for Q^* from data

Recall Value Iteration (VI):

1. Initialize $Q^{(0)}$ arbitrarily.
2. For $t = 1, \dots T$
   - $Q^{(i)}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\max_{a'} Q^{(i-1)}(s',a')\right]$
3. Return $Q^{(T)}$.

Solve for Q^* from data

Given a dataset $\mathrm{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$.

Fitted Q iteration (FQI):

1. Initialize $Q^{(0)}$ arbitrarily.
2. For $t = 1, \ldots T$
   - $Q^{(i)}(s,a) = \mathrm{argmin}_{f \in \mathcal{F}} \sum_{i=1}^n \left(f(s_i, a_i) - r_i - \gamma \max_{a'} Q^{(i-1)}(s_i', a') \right)^2$
3. Return $Q^{(T)}$.

Bellman Error

FQI solves for the equation ∂BE=0.

- $\mathcal{F}$ is the function class.
  - In the tabular setting, $\mathcal{F} = \{f: S \times A \to \mathbb{R}\}$
  - More generally, $\mathcal{F}$ can be a neural network mapping from (s,a) to $\mathbb{R}$.
- FQI is actually used in offline RL.

Recall model-based learning

Given a dataset $D = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{n}$.

Model-based RL:

1. Learn $\hat{P}(s'|s,a) = \frac{N_D(s,a,s')}{N\_D(s,a)}$.

2. Return $\hat{Q} = Q^*_{\hat{P}}$, e.g. via Value Iteration (VI).

🤔 $Q^{(T)}$ from FQI vs. $\hat{Q}$ from MBRL?

😳 They are the SAME! i.e. $\lim_{T\to\infty} Q^{(T)} = \hat{Q}$.

Hint: $Q^{(t)} = \hat{Q}^{(t)}$, where $\hat{Q}^{(t)}$ is from VI in $\hat{P}$.

FQI is a fake model-free method??

- Q: What’s the difference between FQI and MBRL?
- A: Computational/Space complexity.

- MBRL learn and save the model, which lives in $\mathbb{R}^{S\times A\times S}$.
- Value-based RL learn and save the Q function, which lives in $\mathbb{R}^{S\times A}$.

- In order to use FQI in online RL, one must store all historical data, of size $(S+A)T$, which sometimes dominates the space complexity.

A streaming algorithm: Q-Learning

- At time step t,
- Observes transition tuple (s_t, a_t, r_t, s_t')
- Q-learning:
- $Q^{(t+1)}(s_t, a_t) = Q^{(t)}(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma \max_{a'} Q^{(t)}(s_t', a') - Q^{(t)}(s_t, a_t)\right)$
- Recall FQI:
- $Q^{(t)}(s, a) = \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \left(f(s_t, a_t) - r_t - \gamma \max_{a'} Q^{(t)}(s_t', a')\right)^2$
- Q-learning is taking one gradient step w.r.t. the FQI objective with step size $\alpha_t(s_t, a_t)$.

When does Q-Learning converge to Q^*?

- Theorem: Given $\mathcal{M} = \{S, A, P, r, \gamma\}$, Q-learning given by the updated rule

$$Q^{(t+1)}(s_t, a_t) = Q^{(t)}(s_t, a_t) + \alpha_t(s_t, a_t)\left(r_t + \gamma \max_{a'} Q^{(t)}(s_t', a') - Q^{(t)}(s_t, a_t)\right)$$

converges w.p. 1 to Q^* if and only if

$$\sum_t^\infty \alpha_t(s,a) = \infty \quad \text{and} \quad \sum_t^\infty \alpha_t^2(s,a) < \infty.$$

for all $(s, a) \in S \times \mathrm{A}$.

When does Q-Learning converge to Q^*?

- If (s, a) is not visited at step t, then $\alpha_t(s_t, a_t) = 0$.

- So $\sum_t^\infty \alpha_t(s, a) = \infty$ implies that each (s, a) pair is visited infinitely often.

- $\sum_t^\infty \alpha_t^2(s, a) < \infty$ implies that the learning rate must takes a diminishing rate at least $\alpha_t(s, a) \propto 1/\sqrt{N_t(s, a)}$ and at most $1/N_t(s, a)$.

When does Q-Learning converge to Q^*?

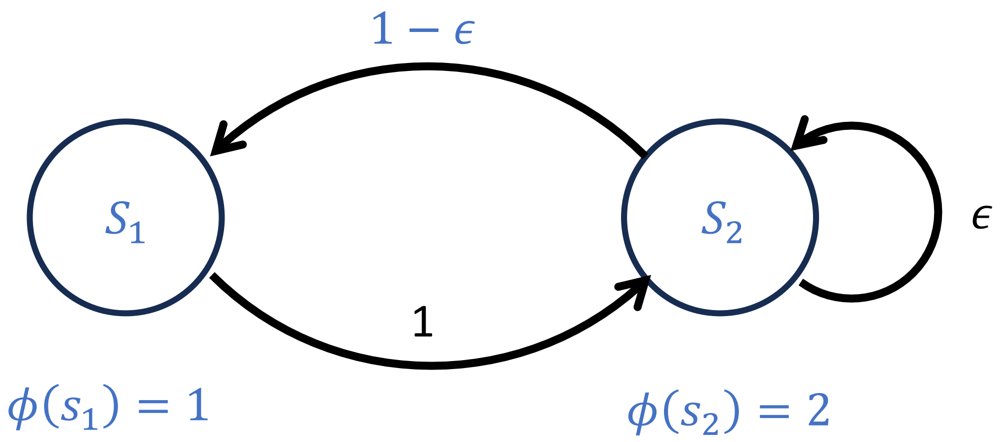
- However, this theorem only works for the tabular setting.
- When # state is large or infinite, you can’t hope to visit each state infinitely often.

When does Q-Learning not converge to Q^*?

- However, this theorem only works for the tabular setting.
- When # state is large or infinite, you can't hope to visit each state infinitely often.
- In fact, Q-learning is known to diverge under function approximation.

When does Q-Learning not converge to Q^*?

- Q-learning can fail under function approximation:



Next time..

- The heuristic solution that kinda(?) worked:
- DQN (2013) and its descendants