Efficient Reinforcement Learning: from the Idealized to the Realistic

FEI FENG

PH.D. THESIS DEFENSE

Committee:
Dr. Deanna Needell
Dr. Lieven Vandenberghe
Dr. Luminita Vese
Dr. Lin Yang (co-advisor)
Dr. Wotao Yin (co-advisor)

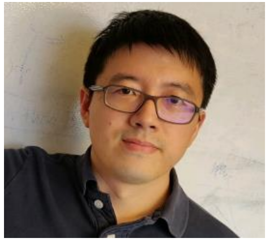
Thanks to my wonderful collaborators

Alekh Agarwal
@Microsoft Research

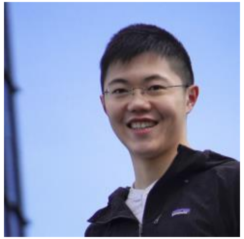
Simon S. Du
@ University of Washington

Ruosong Wang
@ Carnegie Mellon University

Lin Yang
@ UCLA



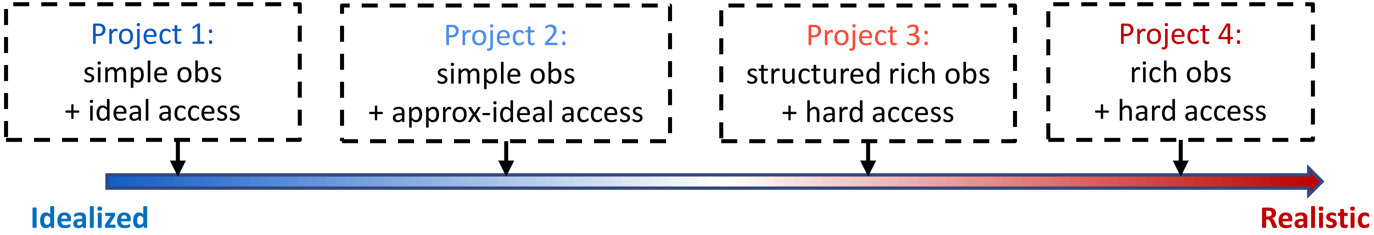
Wotao Yin
@ UCLA



Yibo Zeng
@ Columbia University

Outline

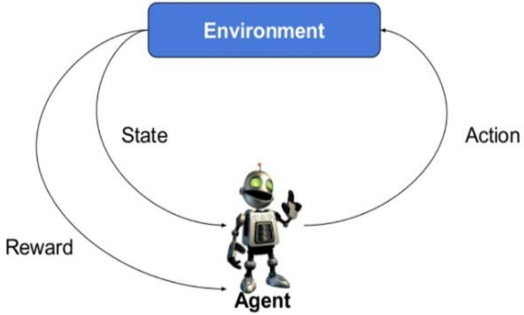
1. Background and Motivation
   - What is Reinforcement learning (RL)?
   - How to achieve efficient RL in various envs? [efficiency scales & challenges.]
2. A String of Answers:



3. Summary and Future Research

Part 1: Background and Motivation

Part 1: Background



- A **policy** $\pi: S \to \Delta(A)$
- The **goal** of RL is: without knowledge of p and r

$$\underset{\pi}{\text{maximize}} \;\; V^{\pi} = E\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid \pi\right]$$

[DeepMind 2017]
Super-human performance on Go.

- A Markov decision process $M := (S, A, p, r, \gamma)$.
- Transition kernel $p\,(s' \mid s\,, a\,)$, reward function $r(s\,, a)$.
- Sample trajectory: $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$.

Also trials in:

- Education
- Medical treatment
- Finance
- Self-driving

[OpenAI 2019]
Defeating **Dota 2** world champion.

An important tool for artificial intelligence.

Part 1: Motivation

Three classes of approaches:

- ❖ Value-based.
- ❖ Policy-based.
- ❖ Linear-programming-based

Simulate dynamic programming with stochastic estimation and function approximation.

Two efficiency scales:

- ❖ [**statistical efficiency/PAC-learnability**]:
  How many samples does it take to learn an ϵ-optimal policy with high probability?

- ❖ [**computational efficiency**]:
  Easy-to-implement? Time complexity? Memory complexity?

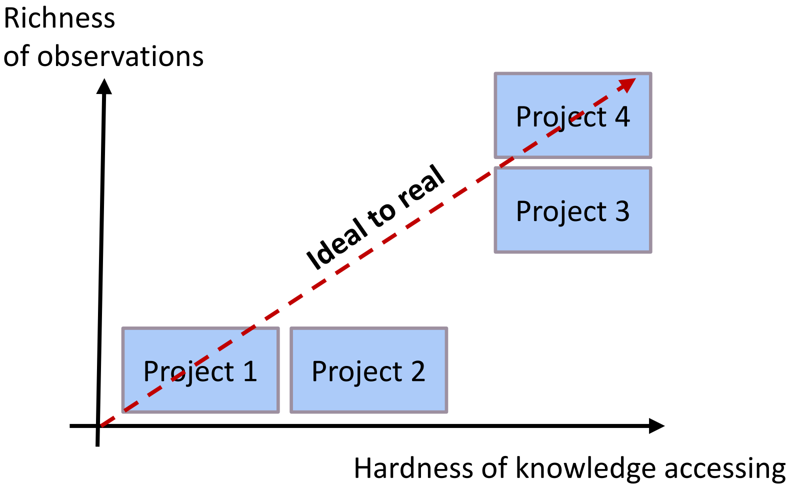
At most polynomial dependency.

Our research goal: improve both efficiencies over prior work in various environments.

Part 2: Efficient RL from the Idealized to the Realistic

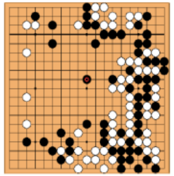
Part 2: Projects Overview

Descriptions of environments



The more realistic, the more challenging.

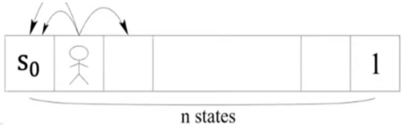
- **Richness of observations**:
  the number of states.

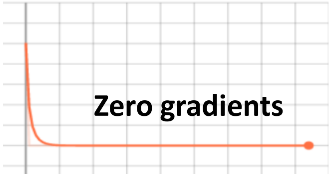


$|S| = 3^{361}$

$|S| \geq 256^{256\times240}$

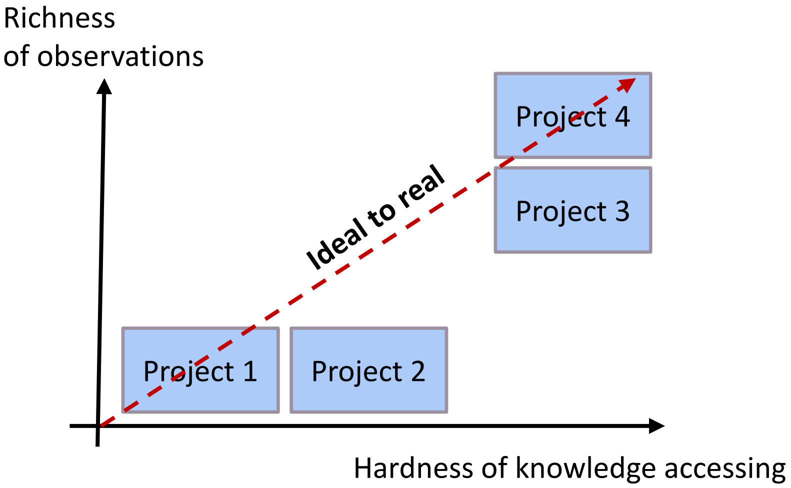
- **Hardness of knowledge accessing**:
  the difficulty of collecting high rewards.





Part 2: Projects Overview

Descriptions of environments



Efficient RL in various training environments.

Next, for each project:

1. Environment setting;
2. Prior results;
3. Our contribution on efficiency improvement;
4. Challenges/technique.

Part 2: Project 1

1. Env setting:

- A small number of states and actions.
- *A generative model:* $\mathrm{GM}(s, a) \to (s', r)$

2. Prior results:

Asynchronous Parallel

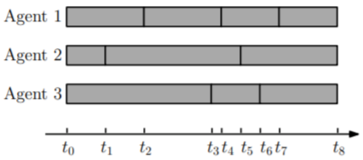


Image from [Peng et al. 2016]

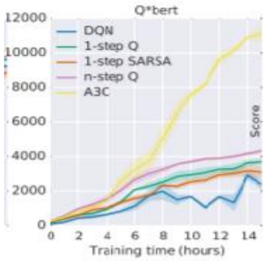
Can we develop an async-parallel RL algo with sample complexity results?

$\tilde{\Theta}\left(\frac{|S||A|}{(1-\gamma)^3\epsilon^2}\right)$ [Azar et al. 2012, Sidford et al. 2018, Agarwal et al. 2019]

Single-thread + $O(|S||A|)$ memory.

[Tsitsiklis 1994] **Async Q-learning**
No sample complexity result.

[Minh et al. 2016] A3C. Empirically successful.



Part 2: Project 1

3. High-level idea:

The original Q-value iteration:

$$Q_{s,a}(t+1) = \sum_{s'} p^a_{ss'} \left(r^a_{ss'} + \gamma \max_{a'} Q_{s',a'}(t)\right), \quad \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}.$$

Approximate with Samples:

$$Q_{s,a}(t+1) = \begin{cases} \frac{1}{K} \sum_{k=1}^{K} (r_k + \gamma \max_{a'} \hat{Q}_{s'_k,a'}) - c, & (s,a) = (s_{t+1}, a_{t+1}); \\ Q_{s,a}(t), & (s,a) \neq (s_{t+1}, a_{t+1}) \end{cases}$$

4. Main contribution:

- Near-optimal sample complexity: $\tilde{O}\left(\frac{|S||A|}{(1-\gamma)^5 \epsilon^2}\right)$.
- $O(|S|)$ memory

5. Key technique:

Math tools:

functional analysis + probability theory

Two error sources:

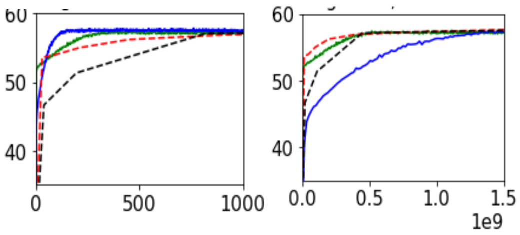
stochastic estimation + delayed information.

Convergence:

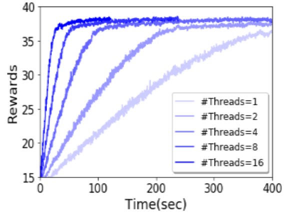
concentration inequality + bounded delay + contraction.

6. Experiment:

Parallel algorithms are run with 20 threads.

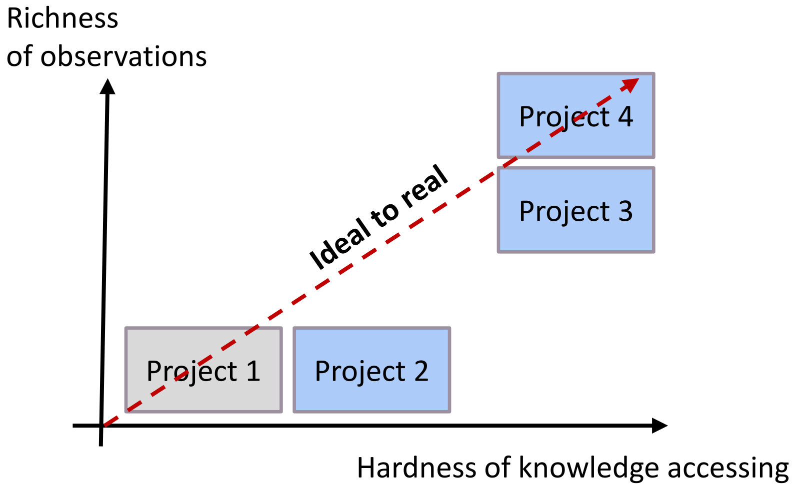


Blue is our algorithm.
[left]: Time; [right]: Sample.

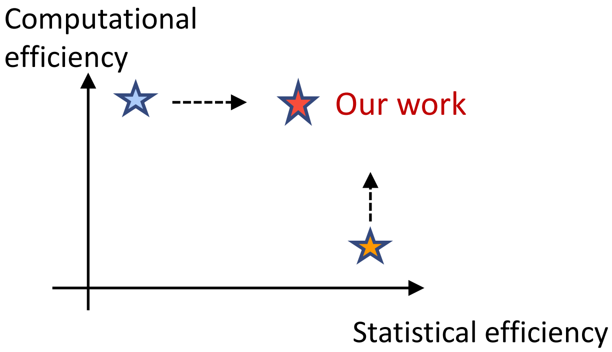


Linear speedup.

Part 2: Projects Overview



Efficient RL with various training environments.



[Project 1]:

- The first sample complexity result for async-parallel RL.
- Accepted by AISTATS 2020.
- Invited speaker at INFORMS 2019.
- Poster presentation at SOCAMS 2019.
- Poster presentation at IPAM Workshop 2020.

Part 2: Project 2

1. Env setting:

- A small number of states and actions.
- The full knowledge of an approximate model $M_0, d_{TV}(M_0, M) < \beta$.

$$d_{TV}(M_0, M) = \max\left\{\max_{(s,a)\in S\times A} \|p_0(\cdot\,|s,a) - p(\cdot\,|s,a)\|_1, \|r_0 - r\|_\infty\right\}$$

How does an approximate (under d_{TV}) model help?

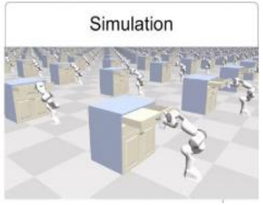
2. Prior results:

- multiple prior models;
- [Jiang 2018], another similarity measurement but is not statistical related.

No systematic answer to the above question.

Knowledge transfer is a widely adopted idea.

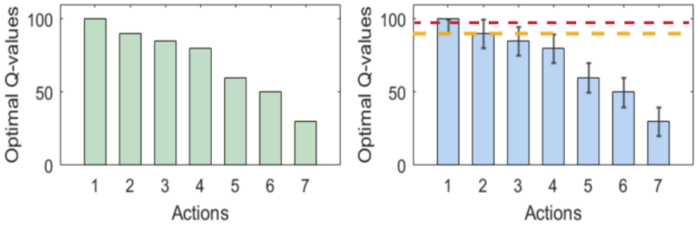
fast adaptation

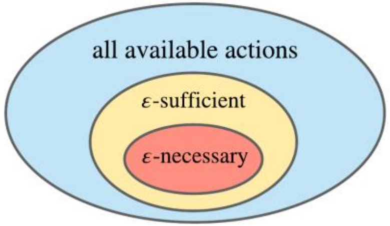


Part 2: Project 2

3. High-level idea:

- If $d_{TV}(M_0, M) \le \beta \Rightarrow \left\| Q^*_{M_0} - Q^*_M \right\|_\infty \le O(\beta)$.
- Induce action optimality information.



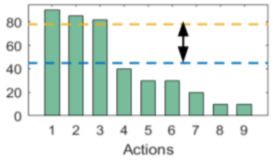


4. Main contribution:

A systematic answer: $\tilde{O}\left(\frac{\sum_s N_{\text{sufficient}}}{(1-\gamma)^3 \epsilon^2}\right)$ & $\Omega\left(\frac{\sum_s N_{\text{necessary}}}{(1-\gamma)^3 \epsilon^2}\right)$

Insights:

- Case I: $N_{\text{sufficient}} = 1.$
- Case II: $N_{\text{sufficient}} \approx N_{\text{necessary}}.$
- Case III: $N_{\text{necessary}} = \Omega(A_s).$



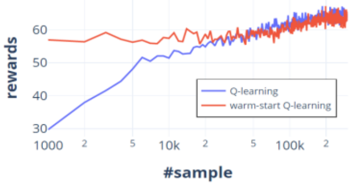
5. Key technique:

Math Tools:

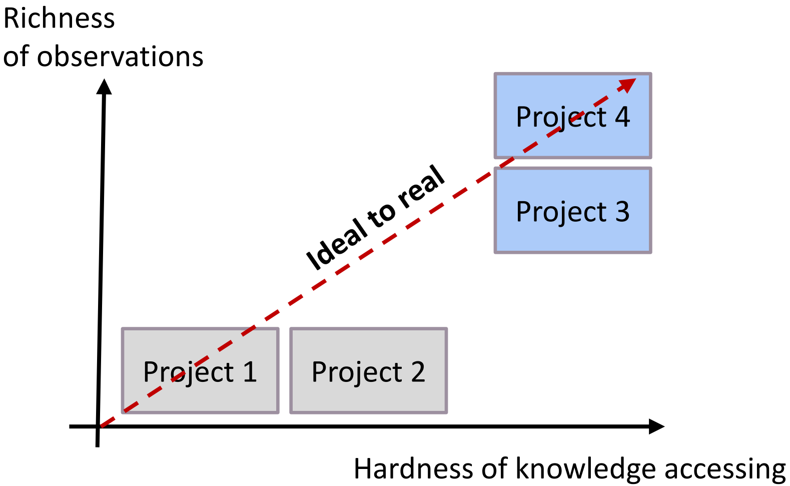
probability theory + information theory.

Lower bound:

construct a hard case.



Part 2: Projects Overview



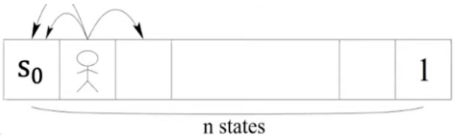
Efficient RL with various training environments.

[Project 2]:

- The first systematic answer to how an approximate model can help under d_{TV}.
- Submitted to JMLR.
- Poster presentation at IPAM Workshop 2020.
- Presently Cited by 3 theoretical RL papers.

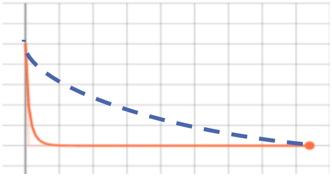
Preliminary of Hard Knowledge Accessing.

[The exploration problem]



1. Limited starting positions;
2. Sparse reward function.

Zero gradients



After rewards reshaping

- One generic solution:
  design <u>artificial rewards</u> to encourage visiting <u>unknown</u> area.

- **<u>Unknown</u>**: rarely visited
- **<u>Artificial rewards</u>**: high rewards on rarely visited area;
  low rewards on frequently visited area.

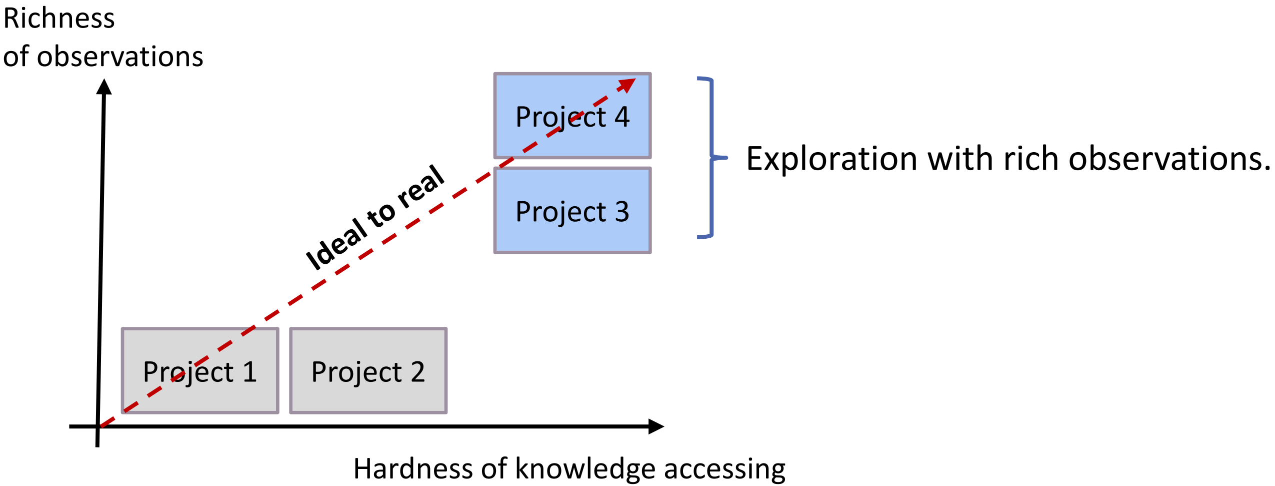
E.g., with finite obs, we log number of visits and let

$$r_{\text{artificial}} \propto \frac{1}{\sqrt{N_{\text{visit}}}} \propto \underline{\text{statistical uncertainty}}$$

$$\boxed{\tilde{\Theta}(|S||A| \cdot \text{poly}(H))}$$

Logging visitation number for every state is not applicable to large-scale state spaces.

Part 2: Projects Overview

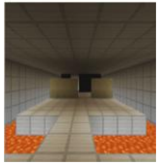


Efficient RL with various training environments.

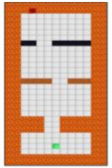
Part 2: Project 3

1. Env setting:

- Rich observations but with intrinsic low dimensional structure.

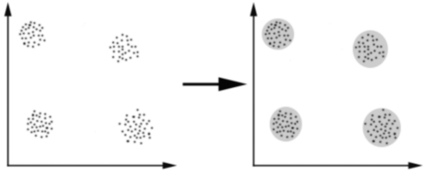


visual signal



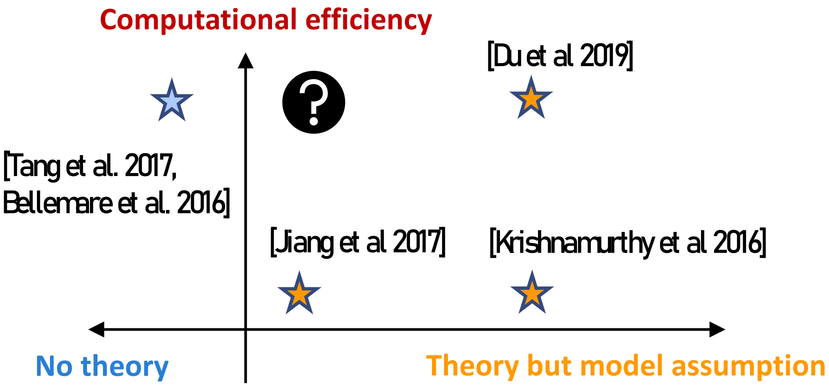
location

Observation similarity often occurs.



A better solution: congregate similar observations.

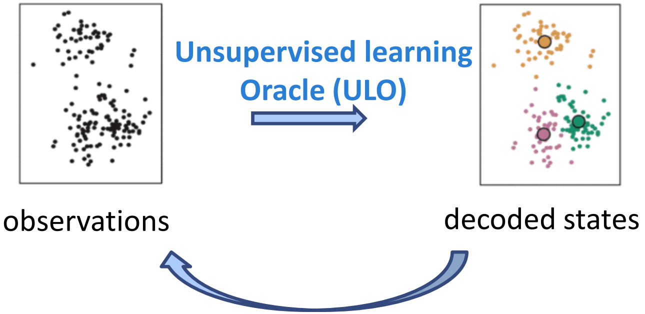
2. Prior results:



Can we develop an efficient algorithm with sound theory but no model assumption?

Part 2: Project 3

3. High-level Idea:



Exploration on a small number of decoded states.

5. Key technique:

- A novel mathematical abstract of ULO;
- A distribution view of RL;
- Statistical learning theory.

Core challenges:

1. No prior knowledge of the true latent states;
2. Interplay between RL and UL.

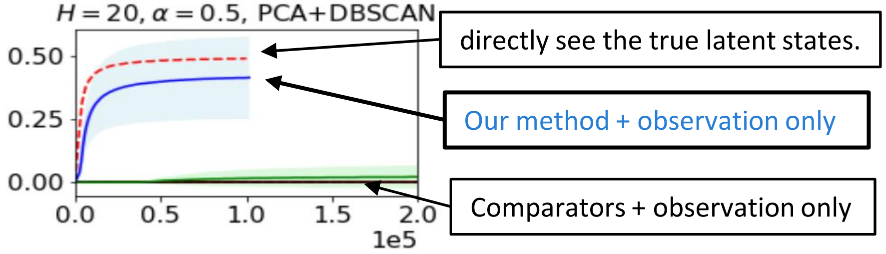
4. Main Contribution:

✓ Requires no additional dynamics assumption;

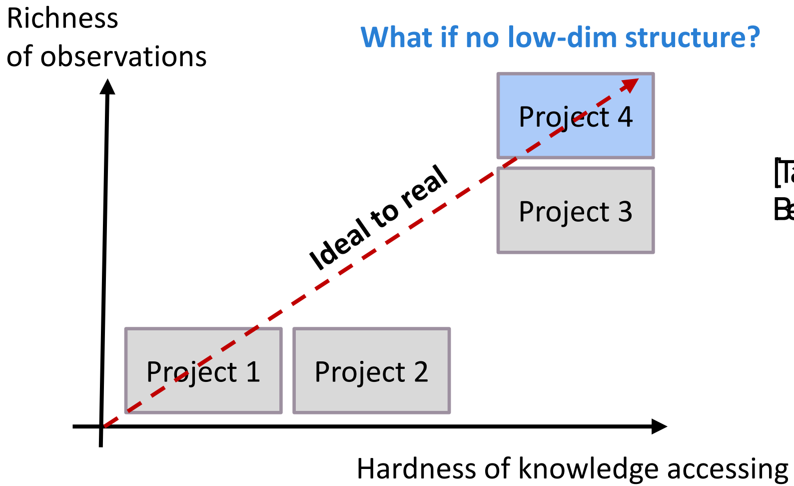
✓ Is PAC-learnable: Poly (|**S**|, |A|, H, $1/\epsilon$, $\log\frac{1}{\delta}$).

✓ Flexible and easy-to-implement

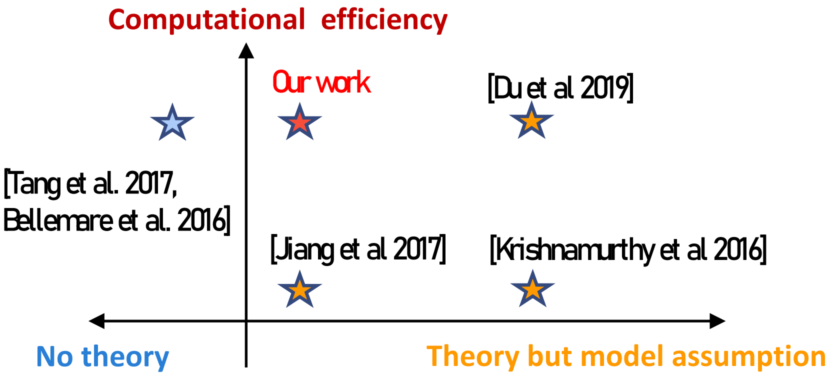
6. Experiment:



Part 2: Projects Overview



Efficient RL with various training environments.



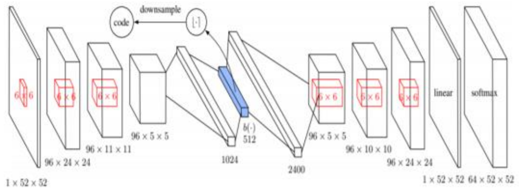
[Project 3]:

- Accepted by Neurips 2020 as Spotlight (Top 4% for ~10000 submissions).
- Invited speaker at RL Theory Seminar.
- Short version accepted by ICML 2020 Workshop: Theoretical Foundations of RL.

Part 2: Project 4

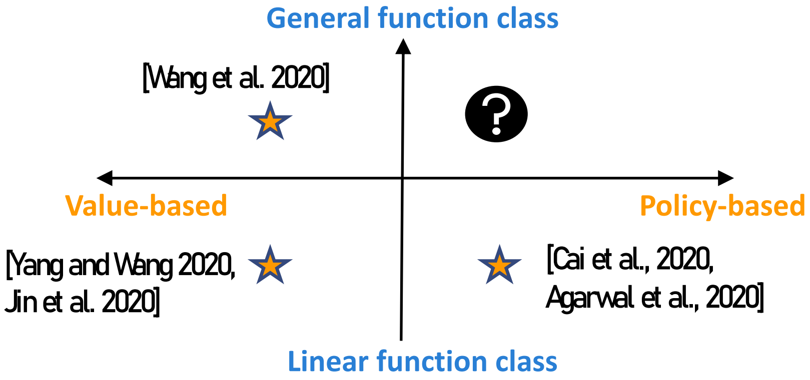
1. Env setting:

- A ‘good’ general function class (e.g. neural networks).



General function approximation is widely used in practice. But little theory is provided.

2. Prior related theoretical results:



<u>What about policy-based exploration with general function approximation?</u>

Chapter 2: Project 4

Core challenge:

how to explore with function approximation?

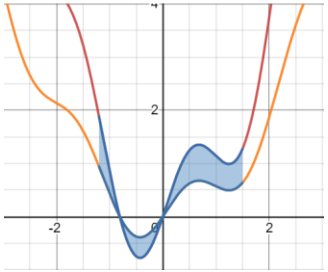
Recall exploration in small problem:

- Known/Unknown: frequently visited/rarely visited;
- Artificial rewards: low/high reward on frequently/rarely visited area $\propto \frac{1}{\sqrt{N_{\text{visit}}}}$

Key technique:

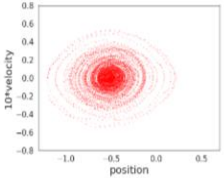
change visitation number to function approximation error.

- Known/Unknown: small/large function approximation error;
- Artificial rewards: low/high reward on small/large error area.

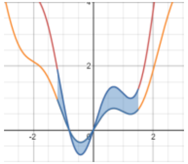


Part 2: Project 4

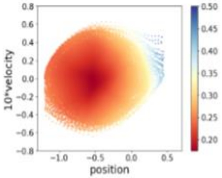
3. High-level Idea:



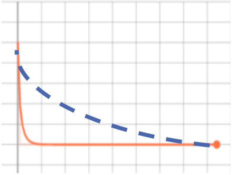
Current visitation



Define known/unknown region



Construct artificial rewards



Optimization.

4. Main Contribution:

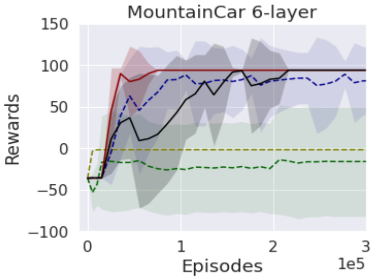
- ✓ Allows model misspecification.
- ✓ Is PAC-learnable: Poly (d_{eluder}, |A|, H, $1/\epsilon, \log\frac{1}{\delta}, C, \log(N_{cover})$).

d_{eluder} $\approx$ how many points does it take to approximately determine a function.

5. Key technique:

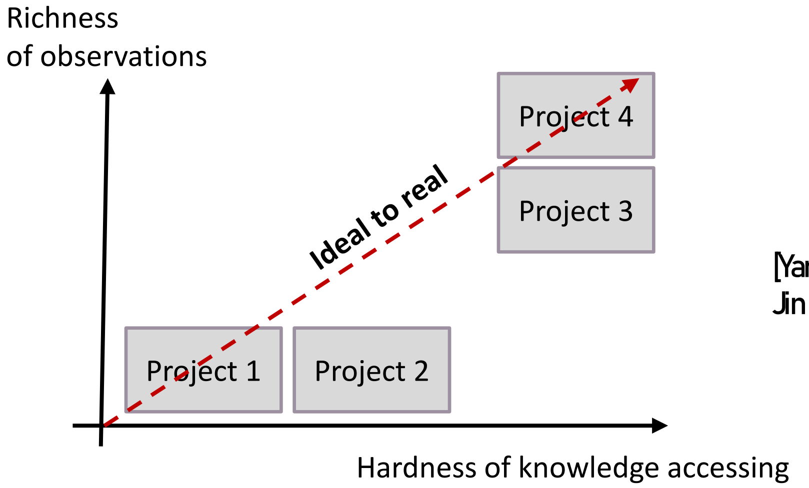
- Martingale concentration;
- Mirror descent convergence;
- Eluder dimension.

6. Experiment:

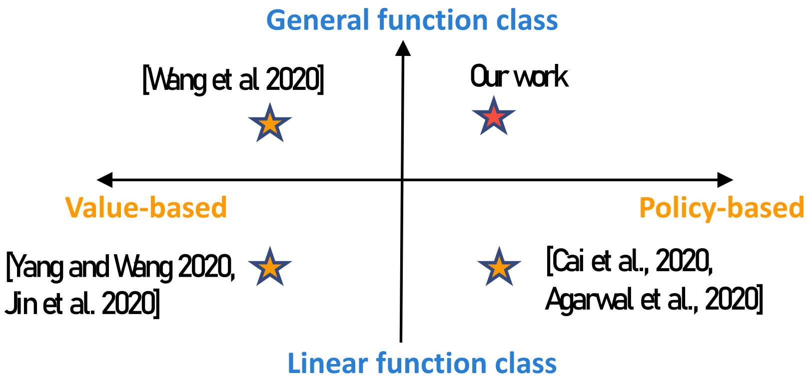


Red line is our algorithm.

Part 2: Projects Overview



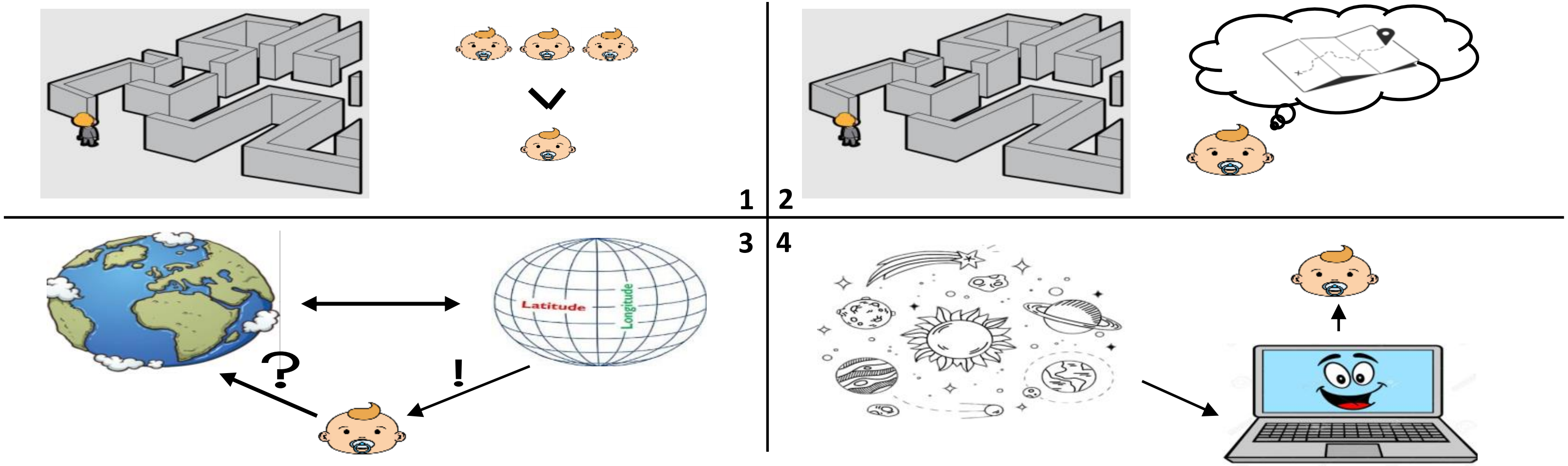
Efficient RL with various training environments.



[Project 4]:

- The first policy-based exploration method with general function approximation.
- Nice empirical performance.
- Submitted to ICML 2021, good initial reviews.

Summary



Our contribution:

A series of answers to improve statistical/computational efficiency of RL training in various environments.

Manuscripts

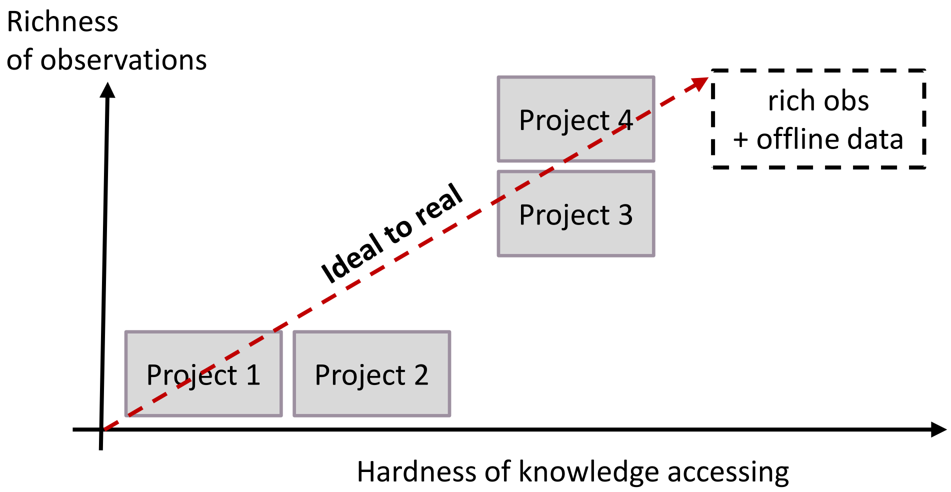
- Yibo Zeng, **Fei Feng**, and Wotao Yin.
  AsyncQVI: Asynchronous-Parallel Q-Value Iteration for Discounted Markov Decision Processes with Near-Optimal Sample Complexity.
  Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR 108:713-723, 2020.

- **Fei Feng**, Wotao Yin, and Lin F. Yang.
  How Does an Approximate Model Help in Reinforcement Learning?
  arXiv preprint arXiv:1912.02986. Submitted to JMLR.

- **Fei Feng**, Ruosong Wang, Wotao Yin, Simon S. Du, and Lin F. Yang.
  Provably Efficient Exploration for Reinforcement Learning Using Unsupervised Learning.
  In Advances in Neural Information Processing Systems, Volumn 33, 2020. Accepted as Spotlight.

- **Fei Feng**, Wotao Yin, Alekh Agarwal, and Lin F. Yang.
  Provably Correct Optimization and Exploration with Non-linear Policies.
  arXiv preprint arXiv:2103.11559. Submitted to ICML 2021.

Future Research



Also:

- Safe RL (RL with constraints),
- RL for optimization,
- Multi-agent RL, etc.

Thank you very much!

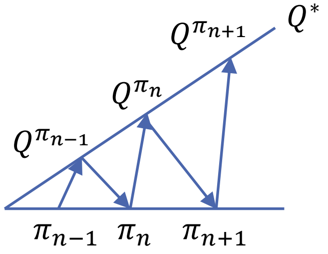
Backup Slides

- If p, r are given, solve MDP with **dynamic programming**.

Policy iteration

$$Q^{\pi_n}(s,a) := E\left[\sum_{t=1}^{\infty} \gamma^t r_t(s_t,a_t) \middle| s_1 = s, a_1 = a, \pi\right], \forall (s,a) \in S \times A$$

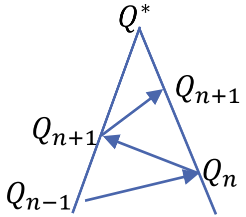
$$\pi_{n+1}(s) = argmax_a\, Q^{\pi_n}(s,a)$$



Value iteration

$$Q_n(s,a) = r(s,a) + \gamma \cdot E_{s' \sim p(\cdot|s,a)}\left[\max_{a' \in A} Q_{n-1}(s',a')\right], \forall (s,a) \in S \times A$$

$$\Rightarrow \pi^*(s) := argmax_a \lim_{n\to\infty} Q_n(s,a)$$



Linear Programming

- Without p, r, solve **RL** by simulating above procedures with stochastic estimation.

Uncertainty quantification using width

Introduce **Width**:

$$\sup_{f, f' \in F} f - f'$$

$$s.t. ||f - f'||_Z \leq \epsilon,$$

where Z is a given dataset.

- Width measures the controllability of function approximation with a finite dataset.
- Width can be used for uncertainty quantification. One can use SGD to estimate width.