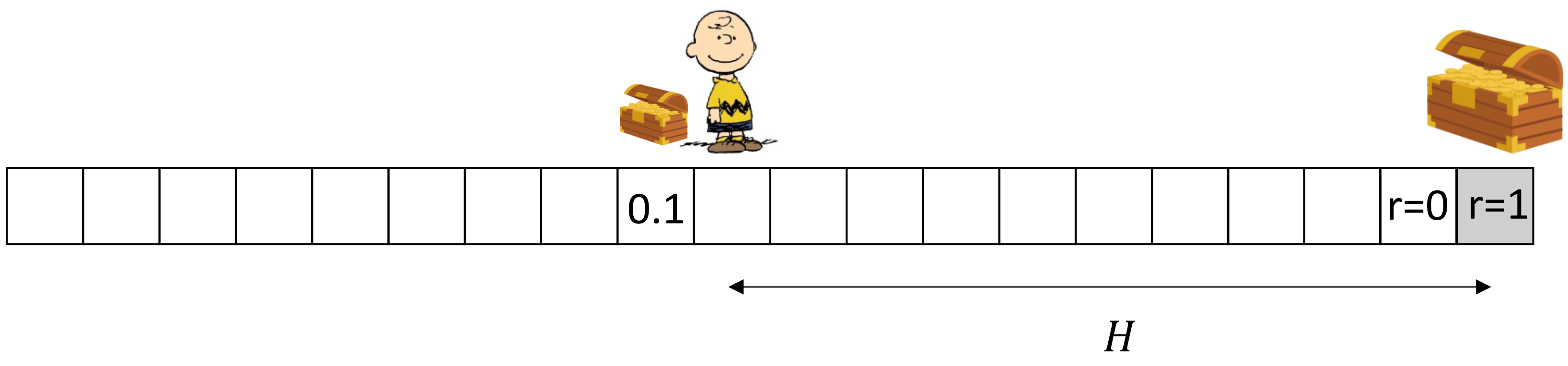
Exploration in MDPs

Chen-Yu Wei

State-Space Exploration in MDPs



Environment:

- Fixed-horizon MDP with episode length H
- Initial state at 0
- A single rewarding state at state H
- Actions: Go LEFT or RIGHT

Suppose we perform DQN with ϵ-greedy with random initialization
⇒ On average, we need 2^H episodes to see the reward

Regret Analysis for MDPs?

- We have done regret analysis for several bandit algorithms:
  - Regression oracle + (ϵ-greedy or inverse gap weighting)
  - UCB
  - EXP3

$$\sum_{\boxed{(s,a,r,s')\in B}} \left(Q_\theta(s,a) - R(s,a) - \gamma \mathbb{E}_{s'\sim} \left[\max_{a'} Q_\theta(s',a')\right] \right)^2$$

- We did not really establish regret bounds for MDPs. We only argued:
  - Approximate value iteration: under the assumption that the data in replay buffer is exploratory small-
  - Approximate policy iteration: monotonically improvement

$$\boxed{\max_{s,a} \left| Q(s,a) - R(s,a) - \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} \left[\max_{a'} Q(s',a')\right] \right| \le \epsilon}$$

$$\Downarrow$$

$$\max_{s,a} \left| Q(s,a) - Q^\star(s,a) \right| \le \frac{\epsilon}{1-\gamma}$$

$$\boxed{Q^{\pi_{t+1}}(s,a) \geq Q^{\pi_t}(s,a)}$$

Regret Analysis for MDPs?

$$\mathbb{E}_{s\sim\rho}\left[V^{\pi^\star}(s)\right] - \mathbb{E}_{s\sim\rho}[V^{\pi}(s)]$$

$$= \sum_{s,a} d_\rho^{\pi}(s,a)\,(V^\star(s) - Q^\star(s,a))$$

For VI-based algorithm (approximating $Q^\star$)

Approximating $Q^\star(s,a)$ requires the replay buffer to cover **wide range of** state-actions.

$$\boxed{(s,a) \sim d_\rho^{\pi^*}(s,a)}$$

make $Q^\pi(s,a) - \sum_{a'} \pi(a'|s)\, Q^\pi(s,a')$

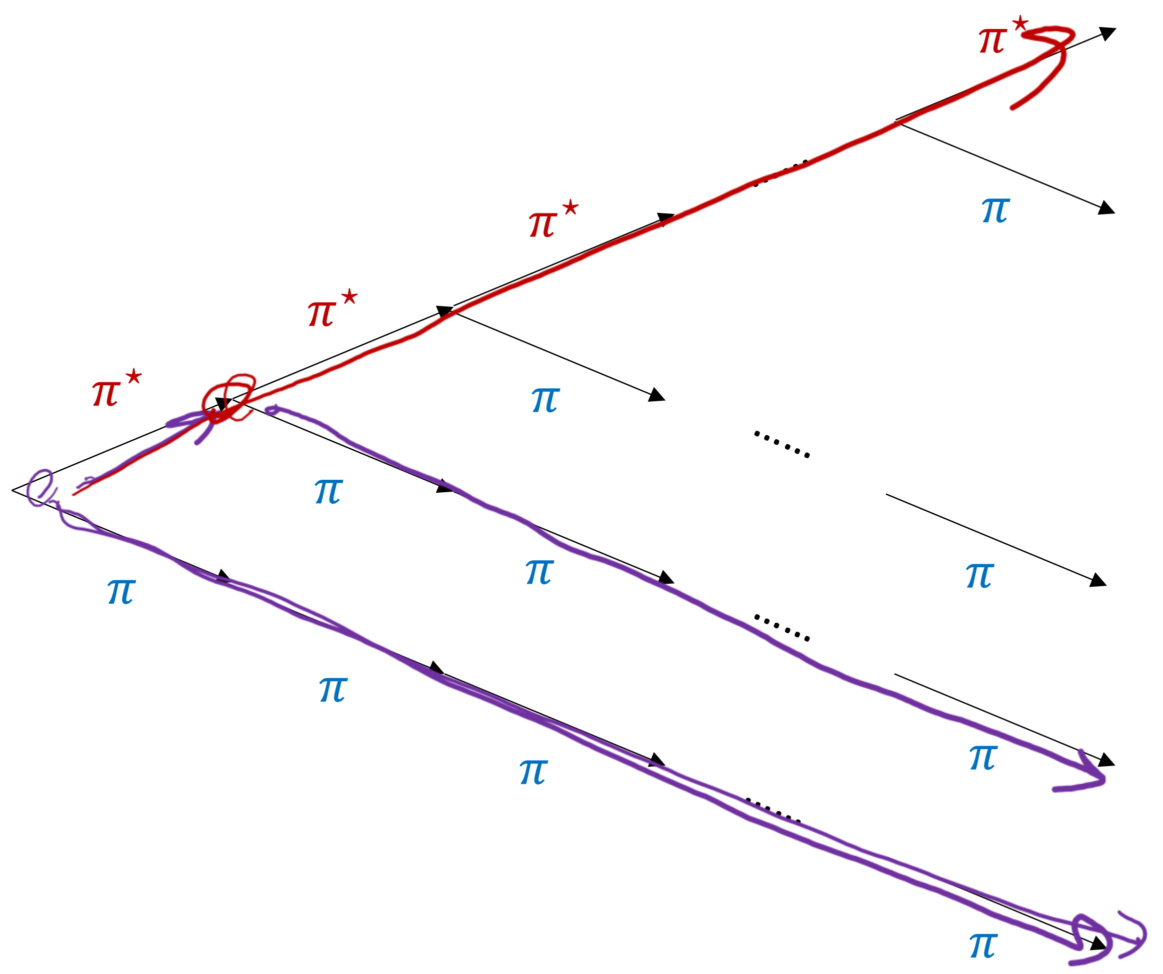
$$= \sum_{s,a} \boxed{d_\rho^{\pi^\star}(s,a)}\,(Q^\pi(s,a) - V^\pi(s))$$

For PI-based algorithm (approximating Q^π)

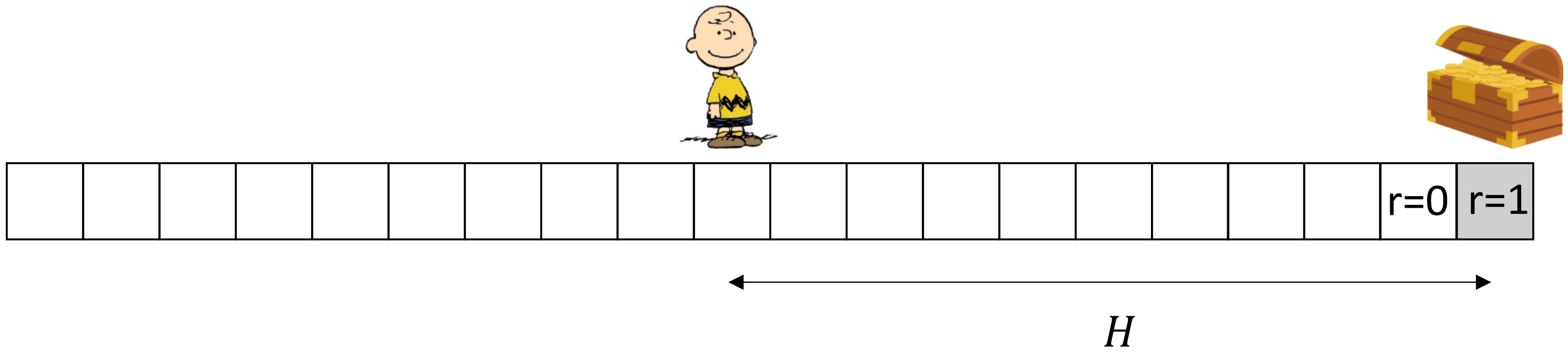
Approximating $Q^\pi(s,a)$ only requires state-actions generated from current policy

But…

$$\sum_{h=1}^{H}\sum_{s,a} d_{\rho,h}^{\pi^\star}(s)\left(\pi'_h(a|s)-\pi_h(a|s)\right)Q_h^{\pi}(s,a) \quad = \quad \sum_{h=1}^{H}\sum_{s,a} d_{\rho,h}^{\pi}(s)\left(\pi'_h(a|s)-\pi_h(a|s)\right)Q_h^{\pi^\star}(s,a)$$



Regret Analysis for MDPs?



$$\sum_{s,a} d_\rho^\pi(s,a)\,(V^\star(s) - Q^\star(s,a))$$

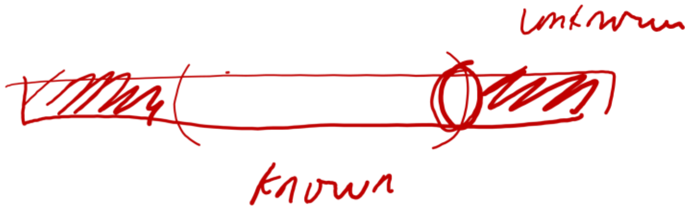
$$\sum_{s,a} d_\rho^{\pi^\star}(s,a)\,(Q^\pi(s,a) - V^\pi(s))$$

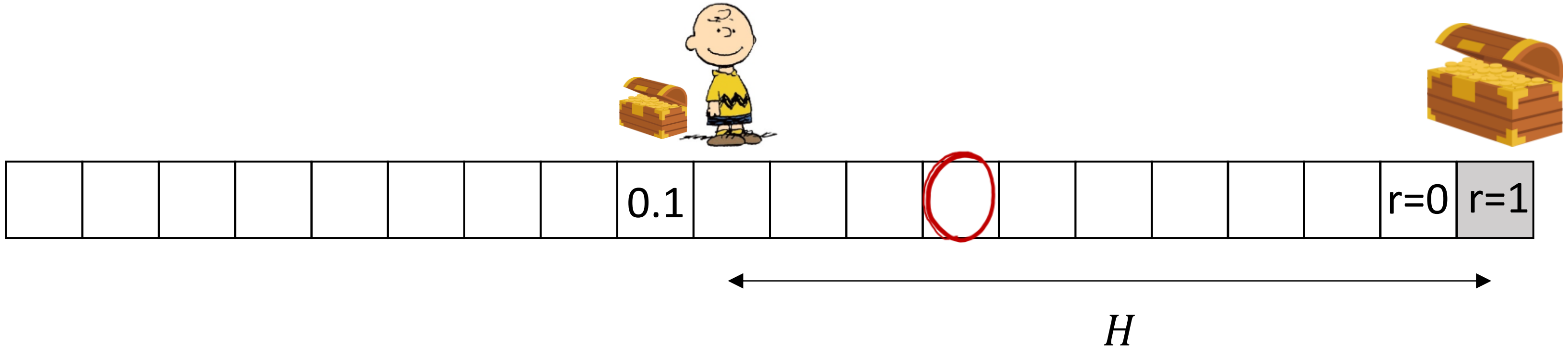
PI-based algorithm only tries to make $\sum_{s,a} d_\rho^{\pi_k}(s,a)\,(Q^\pi(s,a) - V^\pi(s))$ small.

It can only quickly find optimal policy when $d_\rho^{\pi_k} \approx d_\rho^{\pi^\star}$

Insufficiency of algorithms we have discussed for MDPs

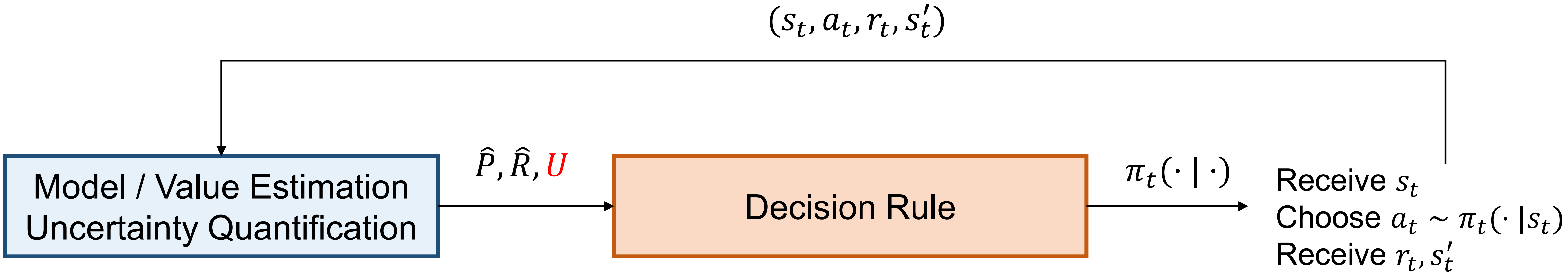
- Lack of **exploration over the state space** (we need **deep exploration**)
- This issue is particularly critical if
  - Local reward does not provide any information
  - Local reward provide misleading information





- Solution
  - Try to make the data (i.e., state-action) distribution close to $d^{\pi^\star}$
  - Try to visit as many states as possible (by quantifying the leaner's **uncertainty** about a state)

Exploration via Uncertainty Quantification



Exploration Bonus for Bandits (Optimism Principle)

- We have discussed this idea for action exploration – UCB.

$\hat{R}_t$

Upper Confidence Bound

$$a_t = \text{argmax}_a \;\; \hat{R}_t(a) + \sqrt{\frac{2\log(2/\delta)}{N_t(a)}}$$

$|R(a) - \hat{R}_t(a)|$

$\lesssim \sqrt{\frac{2\log(\;)}{N_t(a)}}$

$\hat{R}_t(a)$ = the empirical mean of arm a up to time $t - 1$.
$N_t(a)$ = the number of times we draw arm a up to time $t - 1$.

$\hat{R}_t(a) + \sqrt{\frac{2\log(\;)}{N_t(a)}} \gtrsim R(a)$ w.h.p.

Exploration Bonus for MDPs

UCB Value Iteration (UCBVI)

For episode $1, 2, \ldots, T$:

$\tilde{Q}_{H+1}(s, a) = 0 \quad \forall s, a$

For step $H, H-1, \ldots, 1$:

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s', a') + H\sqrt{\frac{2S\log(2/\delta)}{N_t(s,a)}} \quad \forall s, a$$

$$\frac{N_t(s,a,s')}{N_t(s,a)}$$

times we visit s,a and see next state s'

times we visit s,a

$\hat{R}(a) + \sqrt{\frac{1}{N_t(a)}}$

$\boxed{\tilde{Q}_h(s,a) \geq Q^*_h(s,a)}$

Receive $s_1 \sim \rho$

For step $1, 2, \ldots, H$:

Take action $a_h = \mathrm{argmax}_a\, \tilde{Q}_h(s_h, a)$

Receive $r_h = R(s_h, a_h) + \text{noise}, \quad s_{h+1} \sim P(\cdot\,|s_h, a_h)$

$-1 \le R(s,a) \le 1$

$\left|\hat{R}_t(s,a) - R(s,a)\right| \lesssim \sqrt{\frac{1}{N_t(s,a)}}$

Exploration Bonus for MDPs

$\left\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1 \lesssim \sqrt{\frac{S}{N_t(s,a)}}$

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + H\sqrt{\frac{2S\log(2/\delta)}{N_t(s,a)}} \quad \forall s,a$$

$$Q_h^*(s,a) = R(s,a) + \sum_{s'} P(s'|s,a) \max_{a'} Q_{h+1}^*(s',a')$$

$$\boxed{\tilde{Q}_h(s,a) \overset{?}{\ge} Q_h^*(s,a)}$$

$$\tilde{Q}_h(s,a) - Q_h^*(s,a)$$

$$= \hat{R}(s,a) - R(s,a) + \sum_{s'} \left(\hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') - P(s'|s,a) \max_{a'} Q_{h+1}^*(s',a') \right) + b_t(s,a)$$

$$\ge \hat{R}(s,a) - R(s,a) + \sum_{s'} \left(\hat{P}(s'|s,a) \max_{a'} Q_{h+1}^*(s',a') - P(s'|s,a) \max_{a'} Q_{h+1}^*(s',a') \right) + b_t(s,a) \ge 0$$

This holds as long as

$$\left|\hat{R}(s,a) - R(s,a)\right| + \sum_{s'} \left|\hat{P}(s'|s,a) - P(s'|s,a)\right| \underbrace{\max_{a'} Q_{h+1}^*(s',a')}_{\le H} \le b_t(s,a)$$

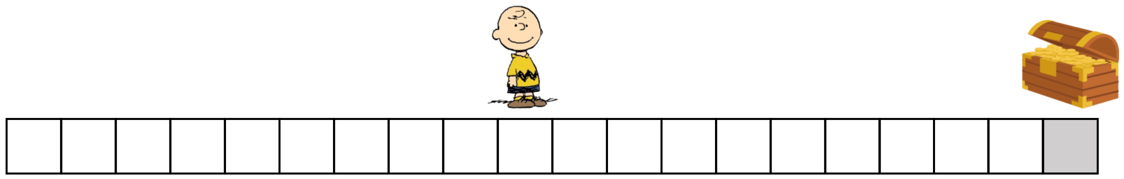
Exploration Bonus for MDPs

Theorem. Regret Bound of UCBVI

Proven in HW4

UCBVI ensures with high probability,

$$\text{Regret} = \sum_{t=1}^{T} \left(V^\star(s_{t,1}) - V^{\pi_t}(s_{t,1}) \right) \lesssim HS\sqrt{AT}.$$



Improving the required number of episodes from 2^H to $\mathrm{poly}(H)$

Jaksch, Ortner, Auer. Near-Optimal Regret Bounds for Reinforcement Learning. 2010.

Azar, Osband, Munos. Minimax Regret Bounds for Reinforcement Learning. 2017.

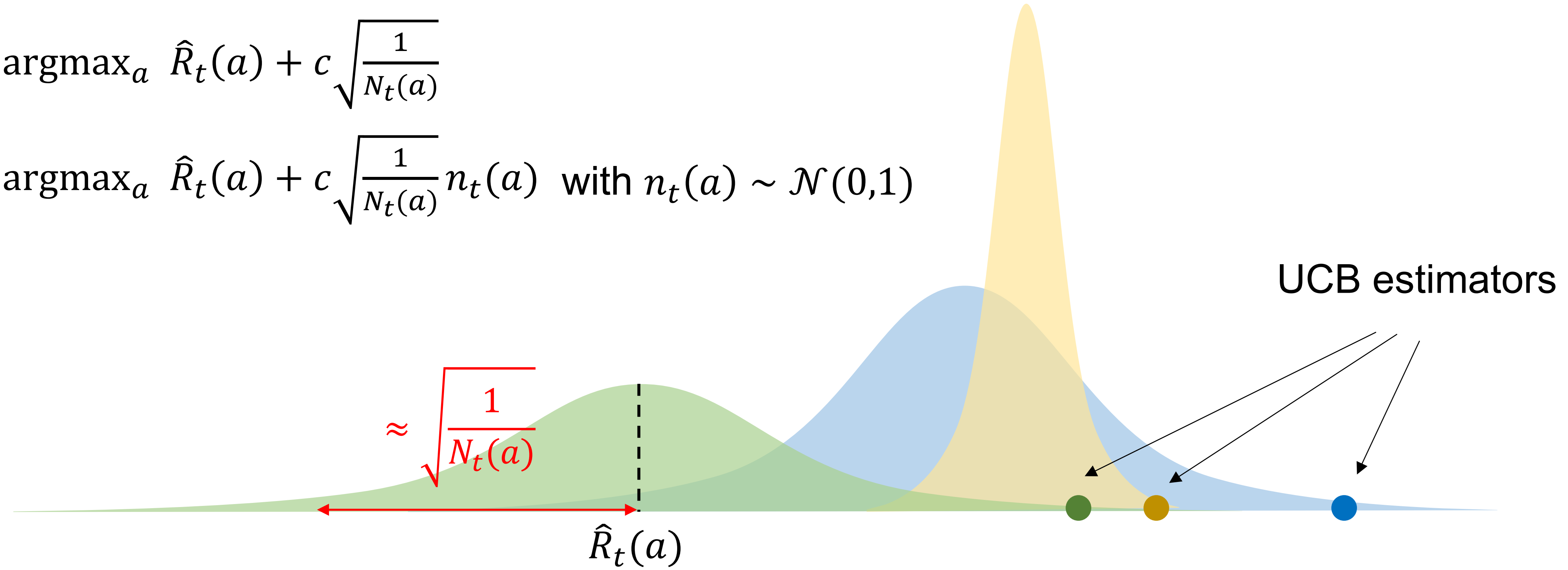
Thompson Sampling
(Posterior Sampling)

Bayesian interpretation:
Assume the reward mean $(\theta(1), \dots, \theta(A))$ is drawn from a Gaussian distribution (prior distribution).
Then the **posterior distribution** is

$$P(\theta(a)|\mathcal{H}_t) = \mathcal{N}\left(\hat{R}_t(a), \frac{1}{N_t(a)}\right)$$

UCB: $a_t \approx \mathrm{argmax}_a \ \hat{R}_t(a) + c\sqrt{\frac{1}{N_t(a)}}$

TS: $a_t \approx \mathrm{argmax}_a \ \hat{R}_t(a) + c\sqrt{\frac{1}{N_t(a)}} n_t(a)$ with $n_t(a) \sim \mathcal{N}(0,1)$



Randomized Exploration for MDPs

Randomized Value Iteration

For episode $1, 2, \dots, T$:

$\tilde{Q}_{H+1}(s,a) = 0 \quad \forall s, a$

For step $H, H-1, \dots, 1$:

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + H\sqrt{\frac{2S\log(2/\delta)}{N_t(s,a)}} \underbrace{n_t(s,a)}_{\sim \mathcal{N}(0,1)}$$

Receive $s_1 \sim \rho$

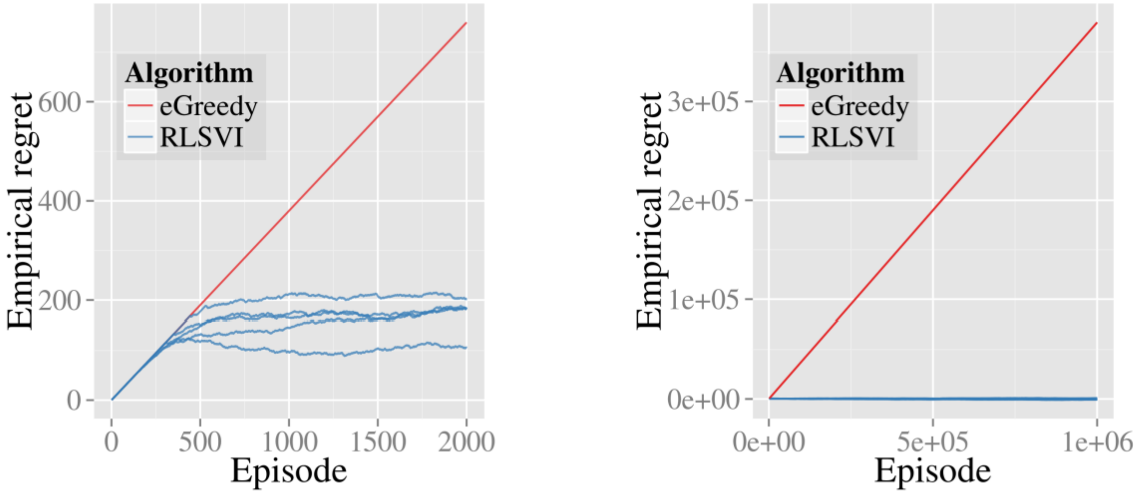
For step $1, 2, \dots, H$:

Take action $a_h = \mathrm{argmax}_a \tilde{Q}_h(s_h, a)$

Receive $r_h = R(s_h, a_h) + \text{noise}, \quad s_{h+1} \sim P(\cdot|s_h, a_h)$

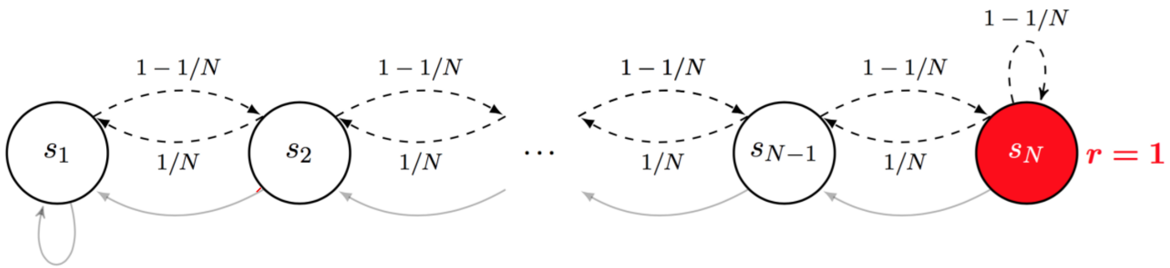
Osband, Van Roy, Wen. Generalization and Exploration via Randomized Value Functions. 2014.

Randomized Exploration for MDPs



(a) First 2000 episodes (b) First 10^6 episodes

Figure 2. Efficient exploration on a 50-chain



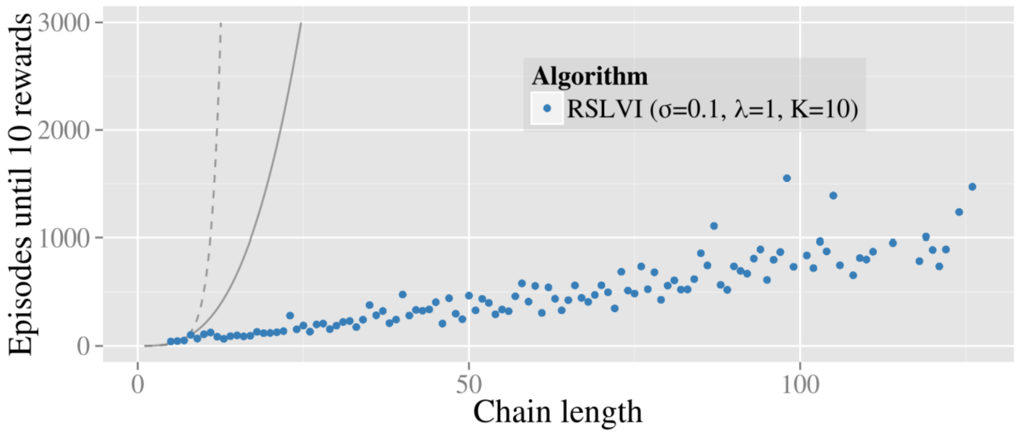


Figure 3. RLSVI learning time against chain length.

Osband, Van Roy, Wen. Generalization and Exploration via Randomized Value Functions. 2014.

Recap: Exploration in Finite-State Finite-Action MDPs

Find exploration bonus $B(s,a)$ such that

$$\left|\hat{R}(s,a) - R(s,a)\right| \le B(s,a) \quad \text{reward uncertainty}$$

$$\left|\mathbb{E}_{s'\sim\hat{P}(\cdot|s,a)}[V(s')] - \mathbb{E}_{s'\sim P(\cdot|s,a)}[V(s')]\right| \le B(s,a) \quad \text{transition uncertainty}$$

$A^{\pi}(s,a)$ is under the true reward R.

Then perform VI (e.g. DQN) over the reward $r(s,a) + \alpha\, B(s,a)$

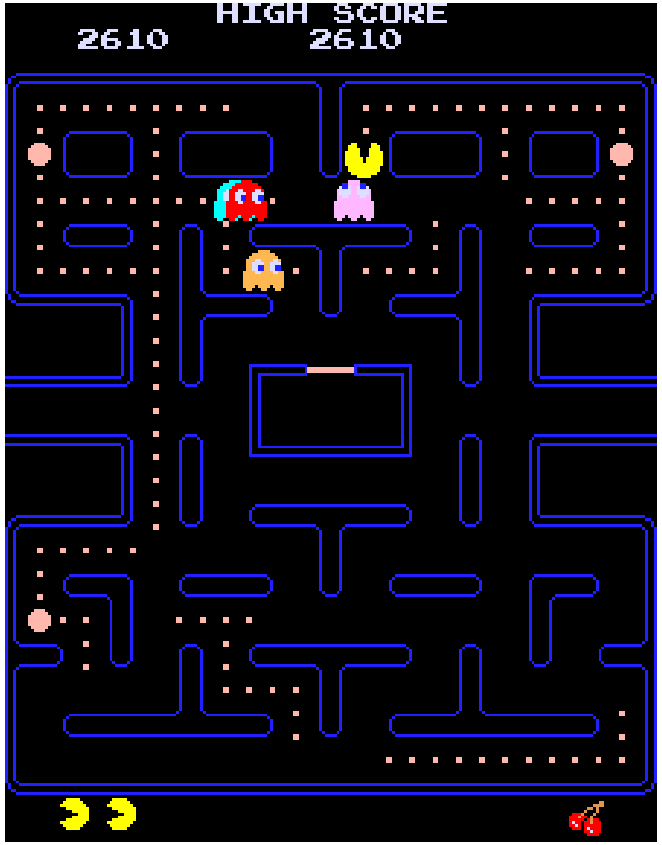
or PI (e.g. PPO, PG) with reward estimator

$$\frac{\pi_\theta(a_i|s_i)}{\pi_{\text{old}}(a_i|s_i)} A^{\pi_{\text{old}}}(s_i, a_i;\ R + \alpha B) \quad \text{or} \quad \frac{\pi_\theta(a_i|s_i)}{\pi_{\text{old}}(a_i|s_i)} A^{\pi_{\text{old}}}(s_i, a_i;\ R) + \alpha \sum_a \pi_\theta(a|s_i) A^{\pi_{\text{old}}}(s_i, a; B)$$

The advantage function of π_{old} with reward function $R + \alpha B$

$$\frac{\pi_\theta(a_i|s_i)}{\pi_{old}(a_i|s_i)} A^{\pi_{old}}(s_i, a_i; \phi) + \alpha \sum_a \pi_\theta(a|s_i) \underline{B(s_i, a)} \leftarrow X$$

$$+ \alpha \sum_a \pi_\theta(a_i|s_i) \underline{Q^{\pi_{old}}(s_i, a; B)}$$



Common Approaches of Exploration

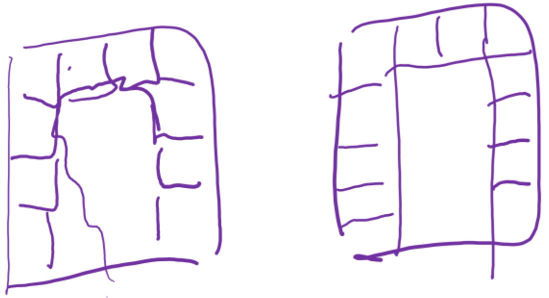
- Optimistic Exploration
  - Upper Confidence Bound
- Randomized Exploration
  - Thompson Sampling (Posterior Sampling)
- Information-Directed Exploration
  - Information-Directed Sampling

$|\hat{R}(a) - R(a)|$

$0.02 \times$

$\leq \frac{1}{gap} = \frac{1}{0.02}$

$O(1)$



① Navigation

0.51 0.49

Ber(0.51) Ber(0.49) 0

World 1

0.49 0.51

Ber(0.49) Ber(0.51) 0.01

world 2

Exploration in Large State Spaces

$B(s,a)$: simultaneous exploration for state and action

$B(s)$: need also a separate exploration mechanism for action space

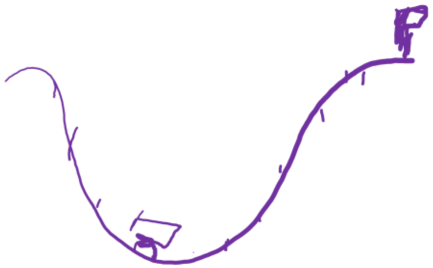
UCB / TS with State(-Action) Discretization

HW4 Task

Partition the state-action space into a finite number of groups

Then instead of counting the #visits to individual state-action, we only count the #visits to each group

$g(s,a)$: the group (s,a) belongs to



$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + c \cdot \frac{1}{\sqrt{N_t(g(s,a))}}$$

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + c \cdot \frac{\mathcal{N}(0,1)}{\sqrt{N_t(g(s,a))}}$$

$\phi(s,a) = e_{g(s,a)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

$\phi(s,a) = e_{s,a} \in \mathbb{R}^{S \times A}$ $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

$\phi(s,a)^\top \Lambda^{-1} \phi(s,a) = \frac{1}{N(s,a)}$

UCB / TS with State(-Action) Features

Suppose for any (s,a), we have access to a feature vector $\phi(s,a) \in \mathbb{R}^d$.

Then instead of counting the #visits to every state-action, we can evaluate the **novelty of the feature**.

$$\Lambda_t = \sum_{i<t} \sum_{h=1}^{H} \phi(s_{ih}, a_{ih}) \phi(s_{ih}, a_{ih})^\top$$

$$\Lambda = \sum_{(s,a)\in D} \phi(s,a)\phi(s,a)^\top$$

$$\phi(s,a)^\top \Lambda^{-1} \phi(s,a) = \frac{1}{N(g(s,a))}$$

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + c \cdot \sqrt{\phi(s,a)\Lambda_t^{-1}\phi(s,a)}$$

Jin et al. Provably efficient reinforcement learning with linear function approximation. 2019.

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + c \cdot \mathcal{N}(0, \phi(s,a)\Lambda_t^{-1}\phi(s,a))$$

Zanette et al. Frequentist Regret Bounds for Randomized Least-Squares Value Iteration. 2019.

Ideas for Exploration

$|\hat{R}(s,a) - R(s,a)| \lesssim \cancel{\frac{1}{\sqrt{N(s,a)}}}$

Ideas from UCB:

1. $\tilde{R}(s,a) = \hat{R}(s,a) + \frac{1}{\sqrt{N(s,a)}}$ where $N(s,a) \approx$ Amount of prior visit to (s,a)
2. $\tilde{R}(s,a) = \hat{R}(s,a) + \underline{e(s,a)}$ where $e(s,a) \approx$ Prediction error on $\hat{R}(s,a)$ and $\hat{P}(\cdot\,|s,a)$

Ideas from TS:

3. $\tilde{R}(s,a) = \hat{R}(s,a) +$ noise whose variance scales with the uncertainty of $\hat{R}(s,a)$ and $\hat{P}(\cdot\,|s,a)$

Ideas from Information-directed Sampling:

4. $\tilde{R}(s,a) = \hat{R}(s,a) + \lambda\, \underbrace{\mathrm{KL}\big(\mathcal{P}(\cdot\,|\mathcal{H}_t, s, a, s'), \mathcal{P}(\cdot\,|\mathcal{H}_t)\big)}_{\text{Information gain}}$

After these modifications, just perform standard RL algorithm over $\tilde{R}$.

1. Bonus from Prediction Error

$\tilde{s}'(s,a)$: next state prediction from (s,a)

Bonus from Prediction Error

Ideally, we would like to quantify $\|\hat{P}(\cdot\mid s,a) - P(\cdot\mid s,a)\|_1$ and set it as bonus.

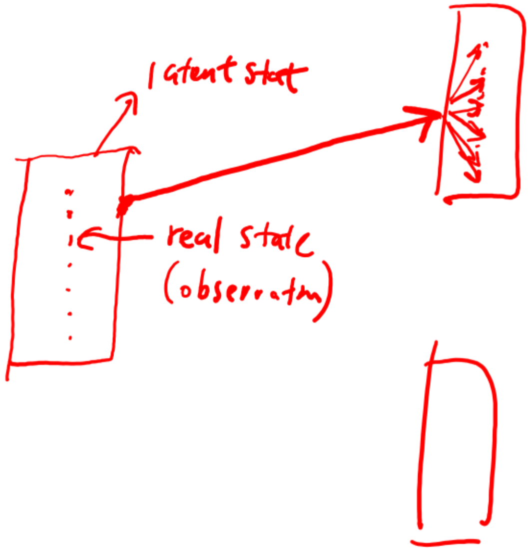
However, modeling transition is not always easy. Sometimes, what we can do is just **predicting the next state** and measure $\|\hat{s}'(s,a) - s'(s,a)\|$

There are some issues if we naively do this:

1. For environments with stochastic transitions, we will never have small prediction error for the next state.
2. For many environments, some part of the state is uncontrollable by the learner (e.g., movement of the clouds in the background).

Bonus from Prediction Error

In some special cases, the world can be modeled as a **deterministic** latent-state MDPs.



1. we have mapping from real state → latent state
2. latent state transition is deterministic

Intrinsic Curiosity Module (ICM)

Inverse model:

State representation $\phi(s)$
↑
real state

$$\hat{a}_t = \boldsymbol{f_I}\left(\phi(s_t), \phi(s_{t+1})\right)$$

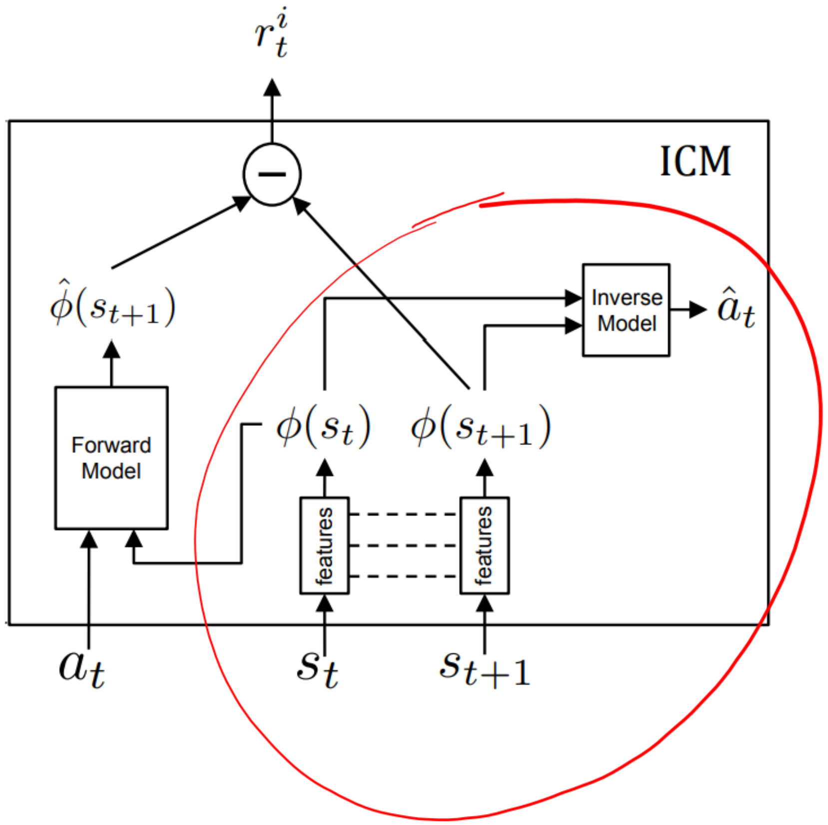
$$\underset{\boldsymbol{f_I}, \phi}{\text{minimize}} \|\hat{a}_t - a_t\|_2^2$$

Forward model:

$$\hat{\phi}(s_{t+1}) = \boldsymbol{f_F}\left(\phi(s_t), a_t\right)$$

$$\underset{\boldsymbol{f_F}}{\text{minimize}} \left\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$$

Bonus $B(s_t, a_t) \triangleq \left\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$



Pathak et al. Curiosity-driven Exploration by Self-supervised Prediction. 2017.

Random Network Distillation (RND)

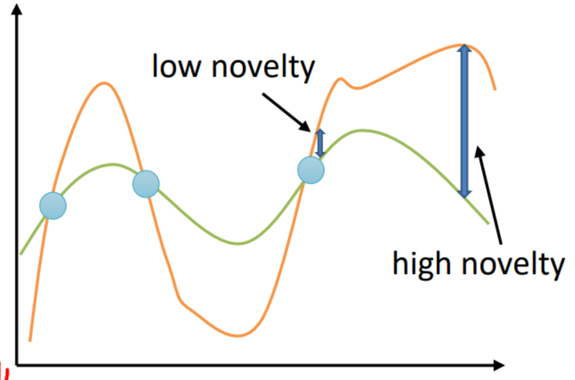
HW4 Task

Given a target function $f^\star(s,a)$ and buffer data $\mathcal{B} = \{(s_i, a_i)\}_{i=1}^n$

Minimize $\frac{1}{n}\sum_{i=1}^{n} \|f_\theta(s_i,a_i) - f^\star(s_i,a_i)\|^2$

θ

Use $B(s,a) = \|f_\theta(s,a) - f^\star(s,a)\|^2$ as the bonus



$\min_\theta \| \underbrace{f_\theta(s_i,a_i)}_{\hat{P}(\cdot|s_i,a_i)} - P(\cdot|s_i,a_i)\|$

Ideally, we want $f^\star(s,a) \approx P(\cdot\,|s,a) \in \mathbb{R}^S$

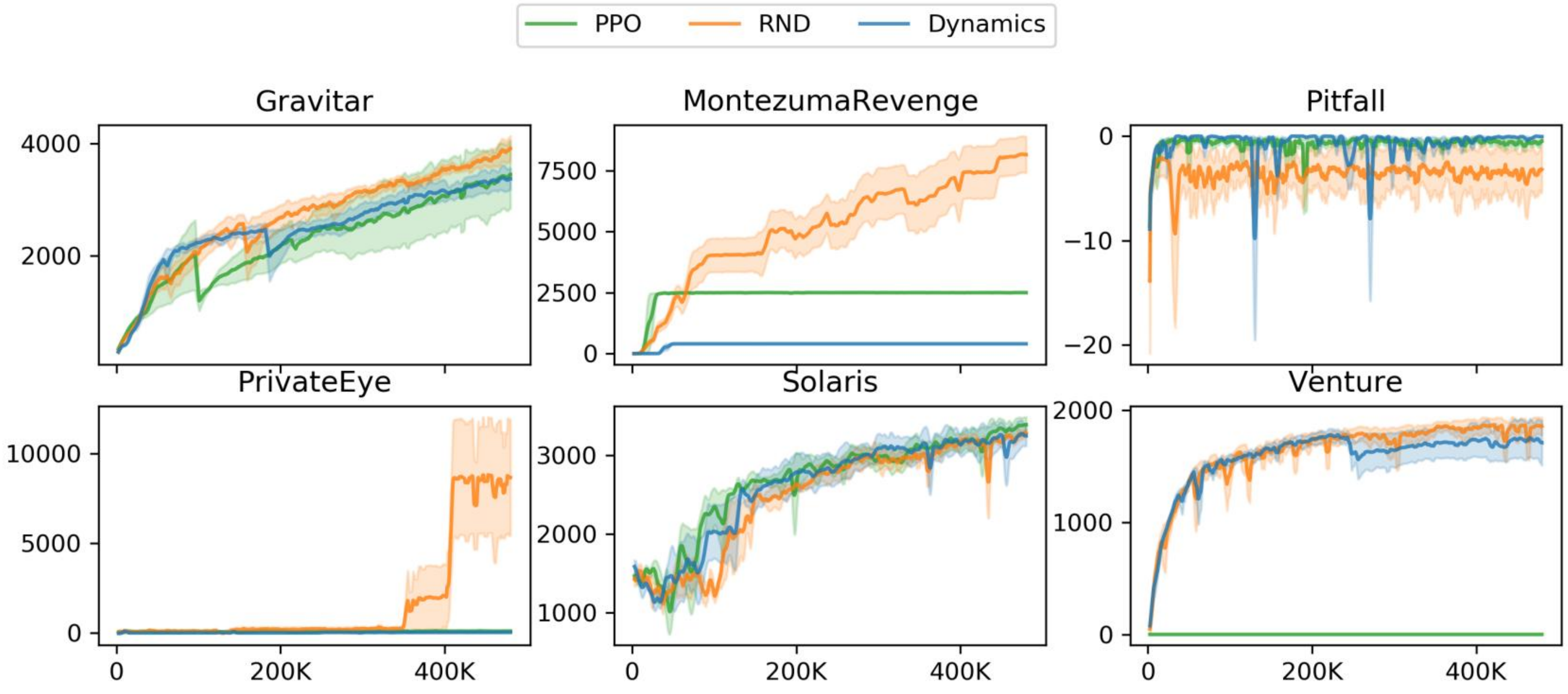
$\| \hat{P}(\cdot|s_i,a_i) - P(\cdot|s_i,a_i)\|$

But we can simply use a random network $f^\star(s,a) = f_\phi(s,a)$

fixed

Burda et al. Exploration by Random Network Distillation. 2018.

Random Network Distillation



Burda et al. Exploration by Random Network Distillation. 2018.

2. Thompson Sampling

Recall: Randomized Value Iteration

Randomized Value Iteration

For episode $1, 2, \ldots, T$:

distribution over $\tilde{Q}_h$

$\tilde{Q}_{H+1}(s, a) = 0 \quad \forall s, a$

For step $H, H-1, \ldots, 1$:

$$\tilde{Q}_h(s, a) \triangleq \hat{R}(s, a) + \sum_{s'} \hat{P}(s'|s, a) \max_{a'} \tilde{Q}_{h+1}(s', a') + H \sqrt{\frac{2\log(2/\delta)}{N_t(s, a)}} \underbrace{n_t(s, a)}_{\sim \mathcal{N}(0,1)}$$

Draw $\tilde{Q}$ from distribution

Receive $s_1 \sim \rho$

For step $1, 2, \ldots, H$:

Take action $a_h = \mathrm{argmax}_a \tilde{Q}_h(s_h, a)$

Receive $r_h = R(s_h, a_h) + \text{noise}, \quad s_{h+1} \sim P(\cdot | s_h, a_h)$

Osband, Van Roy, Wen. Generalization and Exploration via Randomized Value Functions. 2014.

Recall: Randomized Value Iteration

$$\tilde{Q}_h(s,a) \triangleq \hat{R}(s,a) + \sum_{s'} \hat{P}(s'|s,a) \max_{a'} \tilde{Q}_{h+1}(s',a') + n_t(s,a)$$

Adapting this idea to DQN:

$$\theta = \underset{\theta}{\operatorname{argmin}} \sum_{(s,a,r,s') \in \mathcal{B}} \left(r + \max_{a'} Q_{\bar{\theta}}(s',a') + n_t(s,a) - Q_\theta(s,a)\right)^2 \qquad (*)$$

Notice that different noise gives different θ.

Direct generalization from Randomized VI (not easy to implement)

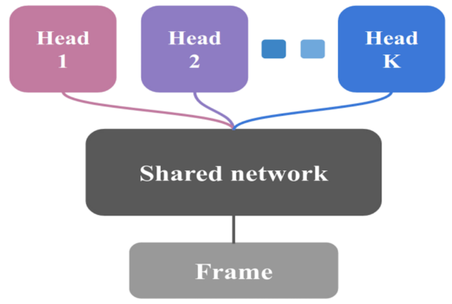
Θ = Space of θ's

In each episode, sample a $\theta \in \Theta$ with the distribution following (*), and execute $\pi(s) = \operatorname{argmax}_a Q_\theta(s,a)$

Bootstrapped DQN

Osband et al. Deep Exploration via Bootstrapped DQN. 2016.
Osband et al. Randomized Prior Functions for Deep Reinforcement Learning. 2018.

Randomly initialize K instances of DQN $\theta_1, \ldots, \theta_K$
(each θ_i has their own target network $\bar{\theta}_i$ and replay buffer $\mathcal{B}_i$).



(a) Shared network architecture

For each episode:

Randomly sample $i \sim \text{Unif}\{1,2, \ldots, K\}$

Execute $\pi(s) = \max_a Q_{\theta_i}(s, a)$ in the whole episode.

Randomly place the obtained (s, a, r, s') in some/all replay buffers.

Update all DQN parameters.

Bootstrapped DQN

Osband et al. Deep Exploration via Bootstrapped DQN. 2016.
Osband et al. Randomized Prior Functions for Deep Reinforcement Learning. 2018.

Some intuitions:

- The random initialization makes $Q_{\theta_1}(s,a), \ldots, Q_{\theta_K}(s,a)$ all very different. We can view them as associated with different initial noise $n_1(s,a)$.
- Over the course of training, for (s,a)'s that are more often visited, their effective magnitude of $n_t(s,a)$ decreases (because we train those DQNs without adding more noise).

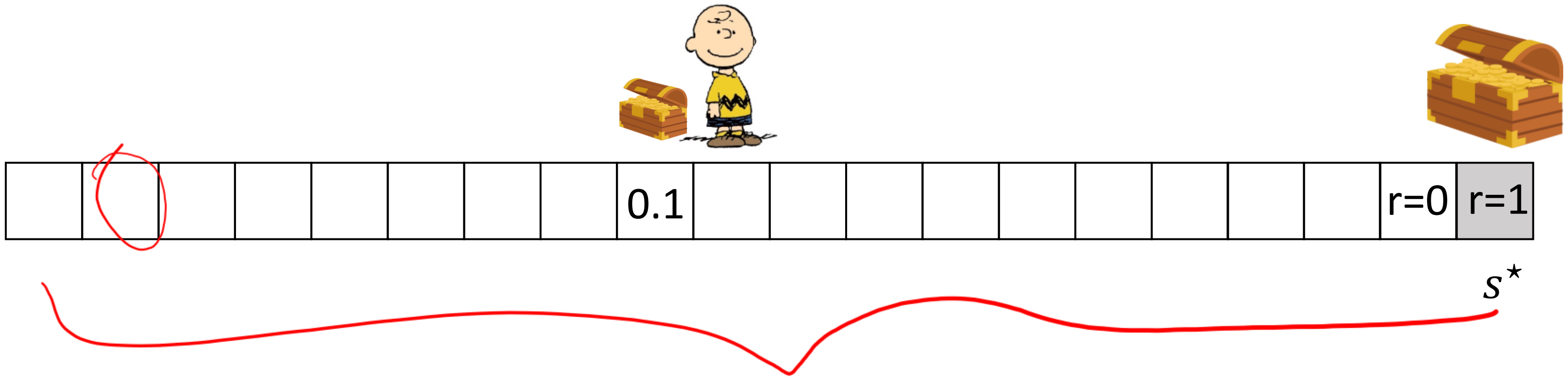
$\boxed{Q(s,a)} = R(s,a) + \mathbb{E}_{s'} Q(s',a')$
$+ n_t(s,a)$

- For (s,a)'s that are not often visited, their effective magnitude of $n_t(s,a)$ remains high.
- **Why does this perform deep exploration?** For a particular state s, if $\max_a Q_{\theta_i}(s,a)$ is initialized high but has not been visited many times before, the training of θ_i will propagate this high value to other state and encourage the learner to reach s from other states.

Bootstrapped DQN

Osband et al. Deep Exploration via Bootstrapped DQN. 2016.
Osband et al. Randomized Prior Functions for Deep Reinforcement Learning. 2018.

- In the toy example, as long as **one of the K DQNs** initializes $s^\star$ (or some states close to it) with a high value, then it can help the learner explore to $s^\star$.
- In this example, roughly we need $K =$ O(number of states) to achieve this effect.

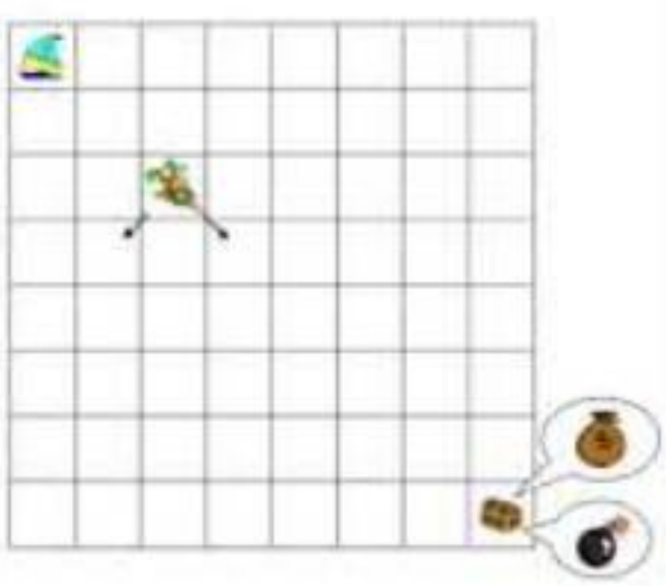


Bootstrapped DQN

Osband et al. Deep Exploration via Bootstrapped DQN. 2016.
Osband et al. Randomized Prior Functions for Deep Reinforcement Learning. 2018.

"Deep Sea" Exploration

- **Stylized "chain" domain testing "deep exploration":**
  - *State = N x N grid, observations 1-hot.*
  - *Start in top left cell, fall one row each step.*
  - *Actions {0,1} map to left/right in each cell.*
  - ***"left" has reward = 0, "right" has reward = -0.1/N***
  - ***... but if you make it to bottom right you get +1.***
- Only one policy (out of more than 2^N) positive return.
- **ε-greedy / Boltzmann / policy gradient / are useless.**



3. Information-Directed Exploration

Houthooft et al. VIME: Variational Information Maximizing Exploration. 2017.