A naïve survey of Exploration in Reinforcement Learning

Minghuan Liu

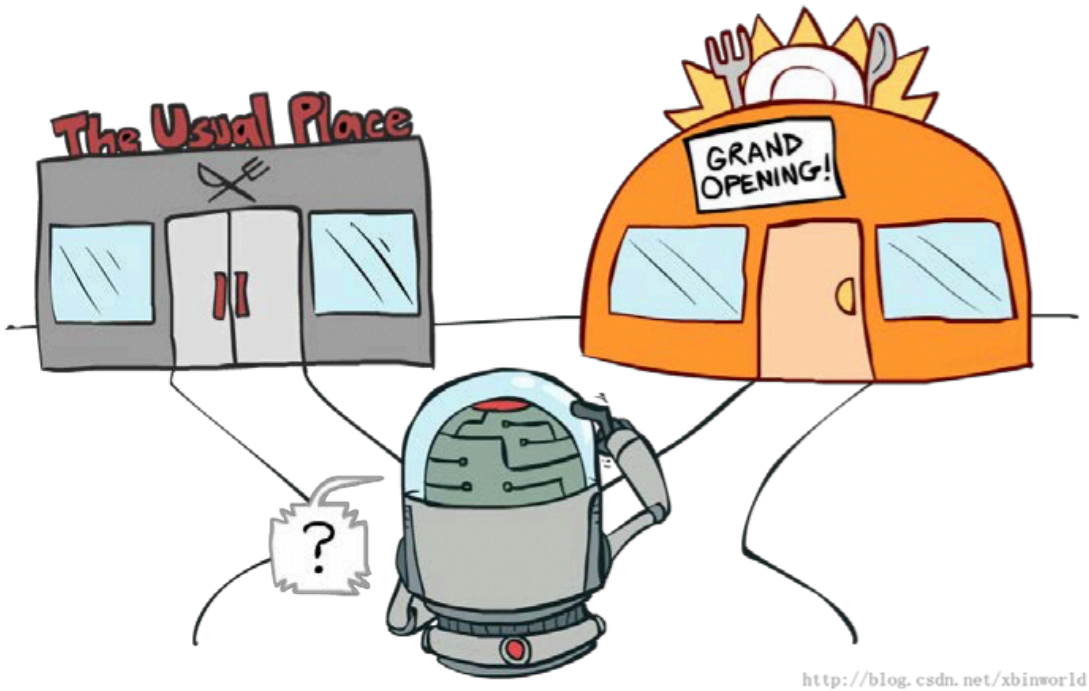
Apex Data & Knowledge Management Lab

Shanghai Jiao Tong University

Content

- What is exploration-exploitation dilemma?
- Exploration in MAB
  - An introduction
  - Naïve methods
  - Complicated methods
- Exploration in RL (MDP)
  - Count based methods
  - Information theory based methods
  - Prediction error methods
  - Other methods

What is exploration-exploitation dilemma?



An example:
Imagine there are ten restaurants near your home. So far, you have only eaten at five restaurants and know how good the five restaurants are. So, if you want to find the best restaurant, where will you go next time?

What is exploration-exploitation dilemma?

- You want to **exploit** known actions/trajectories to get higher rewards, however you still want to **explore** the other actions/trajectories to find if there are better choices, which comes to **exploration-exploitation dilemma**.

- You can choose to explore until the results of every solution is known and then exploit the best all along. With infinite time and chances, you can finally find the optimal solution.

- But how can you get better solutions as soon as possible? Or if the possible choices is large or infinite, can you explore all the time?

What is exploration-exploitation dilemma?

- If an algorithm contains exploration only, then no effective action will be choose.

- If an algorithm contains exploration only, then it becomes the greedy algorithm and we can not get the optimal solution.

- Thus, every algorithm should contains both.

- In RL algorithm, stochastic policy are always used and the exploration belongs to the **Random Variable /Random Noise**.

What is exploration-exploitation dilemma?

Regular exploration choices contains:

1. ϵ-greedy (for discrete action space)

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|}, & if a = argmax_a Q(s,a) \\ \frac{\epsilon}{|A(s)|}, & if a \neq argmax_a Q(s,a) \end{cases}$$

What is exploration-exploitation dilemma?

Regular exploration choices contains:

2. Botzman (Softmax) Policy (for discrete action space)

$$\pi(a|s) = \frac{exp(kQ(s,a))}{\sum_{a'} exp(kQ(s,a')}$$

What is exploration-exploitation dilemma?

Regular exploration choices contains:

3. Gaussian Policy (always for continous action space)

$$\pi(a|s) = \mu_\theta + \epsilon, \epsilon \sim N(0, \sigma^2)$$

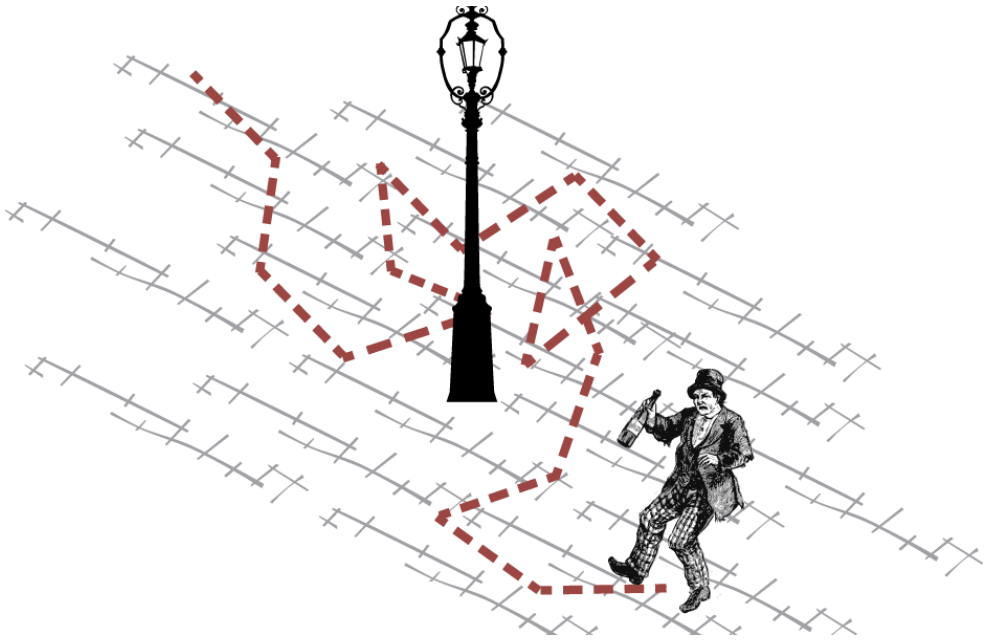
What is exploration-exploitation dilemma?

These policy are essentially

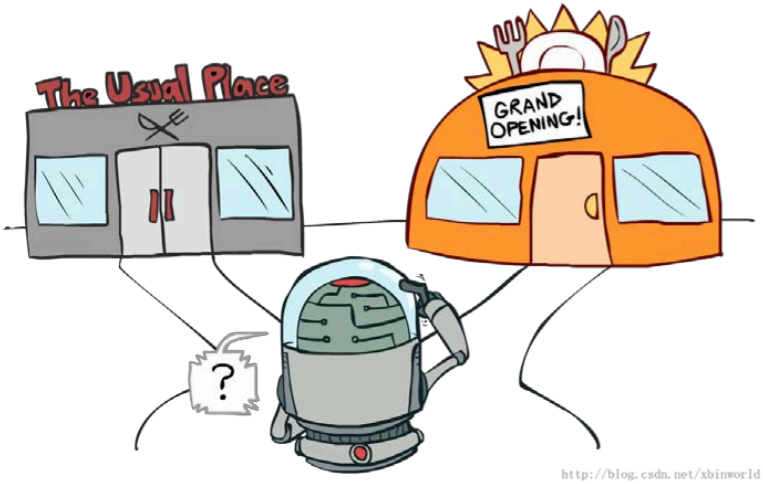
deterministic/greedy policy + random noise

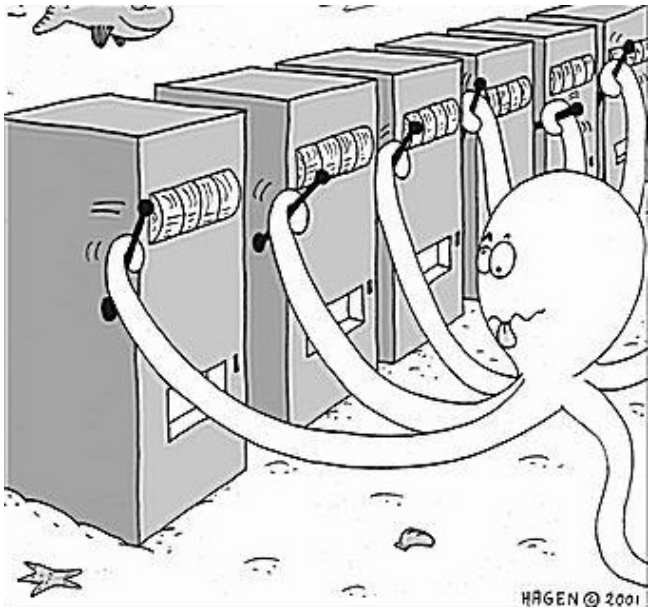
Also called dithering strategy. Like random walking.

Such naïve methods are obviously not make the use of history experience, thus have a low sample efficiency.

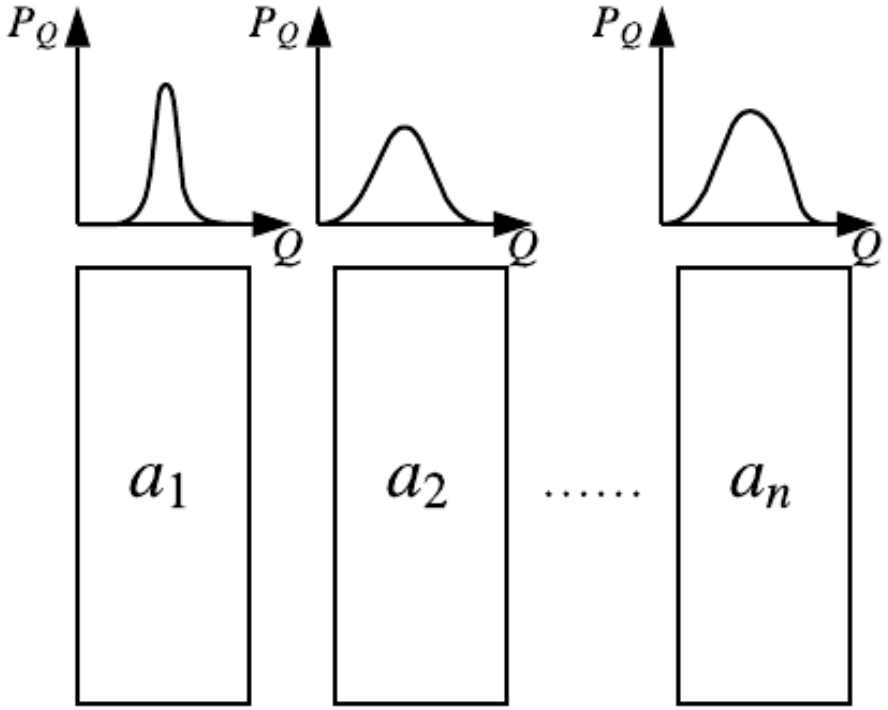


Exploration in MAB
- An Introduction





$$Q(a) = \mathbb{E}[r|a]$$



Exploration in MAB
- An Introduction

Difference between MAB and MDP (RL)

1) MAB seeks optimal policy of a single situation or a situation changes with time (no concept of states); MDP seeks optimal policy of different states, needs sequential decision making.
2) The action choice of MAB only influence current reward, however the action choice of timestep *t* influence the rewards after *t*.

MAB provides a **clear and simple form** for e-e dilemma.

Exploration in MAB
- Naïve Methods

Evaluation of the e-e dilemma solution

- Define **regret** to as the averaged loss of each step:

$$l_t = \mathbb{E}[V^* - Q(a_t)]$$

$$V^* = Q^*(a^*) = \max_{a \in \mathcal{A}} Q^*(a)$$

- Define **total regret** as the total loss:

$$\begin{aligned} L_t &= \mathbb{E}[\sum_{\tau=1}^{t} V^* - Q(a_\tau)] \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)](V^* - Q(a)) \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[N_t(a)]\Delta_a \end{aligned}$$

Exploration in MAB
- Naïve Methods

Evaluation of the e-e dilemma solution

- A good algorithm should achieve a **sublinear** total regret instead of a **linear** total regret
- A suboptimal policy's asymptotic total regret is at least **logarithmic (Lai and Robbins lower bound)**:

$$\lim_{t\to\infty} L_t \leq 8 \ln t \sum_{a|\Delta_a>0} \frac{\Delta_a}{KL(R^a||R^{a^*})}$$

Exploration in MAB
- Naïve Methods

1. ϵ-greedy policy

$$l_t \geq \frac{\epsilon}{\mathcal{A}} \sum_{a \in \mathcal{A}} \Delta_a$$

has a linear total regret

2. Softmax policy

$$\mathbb{P}_t(a) = \frac{exp(Q_t(a)/\tau)}{\sum_{i=1}^{n} exp(Q_t(i)/\tau}$$

$$l_t = \sum_{a \in \mathcal{A}} P_t(a) \Delta_a$$

The effect of these two policy depends on tasks and fine-tune. Some said that ϵ is easier to set than τ because the latter always needs a priori. Others said that it is unreasonable to transfer Q into probabilities.

Exploration in MAB
- Naïve Methods

3. Optimistic initial values

$$Q_1(a) = r_{max}$$

- Make active exploration in the beginning
- Unfit for a dynamic situation
- Has linear total regret

Exploration in MAB
- Complicated Methods

- These naïve methods are always useful in practical experiments, because the evaluation of Q is getting more accuracy during experiments, however the total regrets are limited in a linear increment.

- Complicated methods always have sublinear total regret, these methods can be divided into **Frequency Theory** based methods and **Beyesian Methods**.

Exploration in MAB
- Complicated Methods

1. Decaying ϵ-greedy policy

Consider

$$c > 0$$

$$d = \min_{a|\Delta_a > 0} \Delta_a$$

$$\epsilon_t = \min\left\{1, \frac{c|\mathcal{A}|}{d^2 t}\right\}$$

Then

$$l_t = \frac{c}{d^2 t} \sum_{a \in \mathcal{A}} \Delta_a$$

$$L_t \approx \int_t l_t dt$$

$$= \int_t \frac{1}{t} \frac{c}{d^2} \sum_{a \in \mathcal{A}} \Delta_a$$

$$= \frac{c}{d^2} \sum_{a \in \mathcal{A}} \Delta_a \ln t$$

The total regret is **sublinear**.

Such needs priori of the gap.

Exploration in MAB
- Complicated Methods

1. Decaying ϵ-greedy policy

- Decaying τ for Botzman policy can also lead to a sublinear total regret.

- Some works focus on parameter (ϵ) adaptive adjustment.

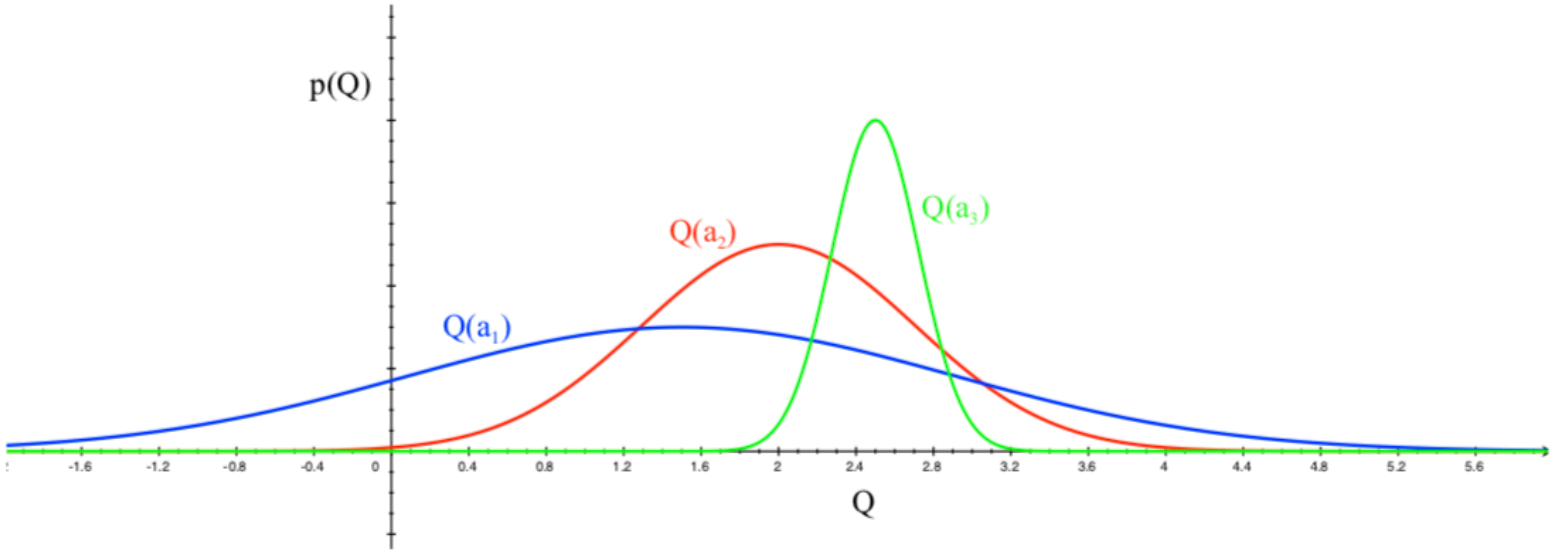
- Some works consider contextual information.

Exploration in MAB
- Complicated Methods

2. Upper Confidence Bound(UCB)

$$a_t = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$$

$$\hat{U}_t(a) \propto \frac{1}{N_t(a)}$$



Exploration in MAB
- Complicated Methods

2. UCB1 (2002)

$$a_t = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$$

$$\hat{U}_t(a) \propto \frac{1}{N_t(a)}$$

$$\mathbb{P}(Q(a) > \hat{Q}_t(a) + \hat{U}_t(a)) \leq e^{-2N_t(a)U_t^2(a)}$$

$$U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

$$a_t = \arg\max_{a \in \mathbb{A}} Q(a) + \frac{2\log t}{N_t(a)}$$

Exploration in MAB
- Complicated Methods

3. Pursuit (1984)

$$p_i(0) = 1/k$$

$$p_i(t+1) = \begin{cases} p_i(t) + \beta(1 - p_i(t)), & if i = argmax_j \hat{\mu}_j(t) \\ p_i(t) + \beta(0 - p_i(t)), & otherwise \end{cases}$$

$$\beta \in (0,1)$$

There are proofs with PAC in automaton that Pursuit can converge to optimal policy.

Exploration in MAB
- Complicated Methods

4. POKER (Price of Knowledge and Estimated Reward) (2005)

$$p_{a_t} = Q_{a_t} + \mathbb{P}[q_a \geq \hat{V}_t^* + \sigma_{\mu_t}]\sigma_{\mu_t} H$$

Q_{a_t} is the reward evaluation of current action.

$\hat{V}_t^*$ is the current evaluation of the best reward.

$\sigma_{u_t} = \mathbb{E}[V^* \hat{V}_t^*]$ is the reward evaluation of current action.

V_t^* is the best reward.

H is horizon.

The concept of **Information Value** sometimes are called **exploration bonus**.

Are proved to be **zero-regret**.

Knowledge Acquirement leading exploration.

Exploration in MAB
- Complicated Methods

5. Probability Matching (2010-2018)

$$\begin{aligned}\pi(a|h_t) &= \mathbb{P}[Q(a) > Q(a'), \forall a' \neq a|h_t] \\ &= \mathbb{E}_{\mathcal{R}|h_t}\left[\mathbb{I}(a = \arg\max_{a \in A} Q(a))\right] \\ &= \int \left[\mathbb{I}(\mathbb{E}(r|a^*, x, \theta) = \max_{a \in A} \mathbb{E}(r|a, x, \theta)\right] P(\theta|h_t) d\theta\end{aligned}$$

1. Get posterior $P(\theta|h_t)$
2. Sample θ^*
3. Select action $a^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}[r|\theta^*, a, x]$

Achieves Lai and Robbins lower bound.
Lead to study on Beyesian Bandit.

Exploration in MAB
- Complicated Methods

5. Beyesian UCB (2012)

- Each time a is selected, update posterior $\pi_a^t(\theta_j)$
- Consider the reward $Q(a) \sim N(\mu_a, \sigma^2)$ and the priori is $\pi_a^0 = \frac{1}{\sigma^2}$
- Then the Beyesian UCB policy is:

$$a_t = \arg\max_{a \in \mathcal{A}} \frac{S_a(t)}{N_a(t)} + \sqrt{\frac{S_a^{(2)}(t)}{N_a(t)}} \mathcal{Q}\left(1 - \frac{1}{t(\log n)^c}, \mathcal{T}(N_a(t) - 1)\right)$$

$\mathcal{Q}(t, \rho)$ is the quantile function of ρ such that $\mathbb{P}_\rho(X \leq \mathcal{Q}(t, \rho)) = t$

$\lambda_a^t (1 \leq j \leq K)$ is the posterior of μ_a, which is the average of $Q(a)$

$\mathcal{T}(k)$ is the Student-t distribution with df k.

Exploration in MAB
- Complicated Methods

5. Gittin Indices (1979, 2002, 2010)

- Each time a is selected, update posterior for Q_a.
- Bayes-adaptive Reinforcement Learning
- With experiment going on, the reward distribution evolves into different information state.
- Given a priori, the reward distribution can be calculated.
- The computation cost is high.
- Problems:
  - Incomplete learning
  - Require actions independent
  - Not fit for fixed strategies
  - The discounting scheme must be geometric

Exploration in MAB
- Complicated Methods

Warning!!!

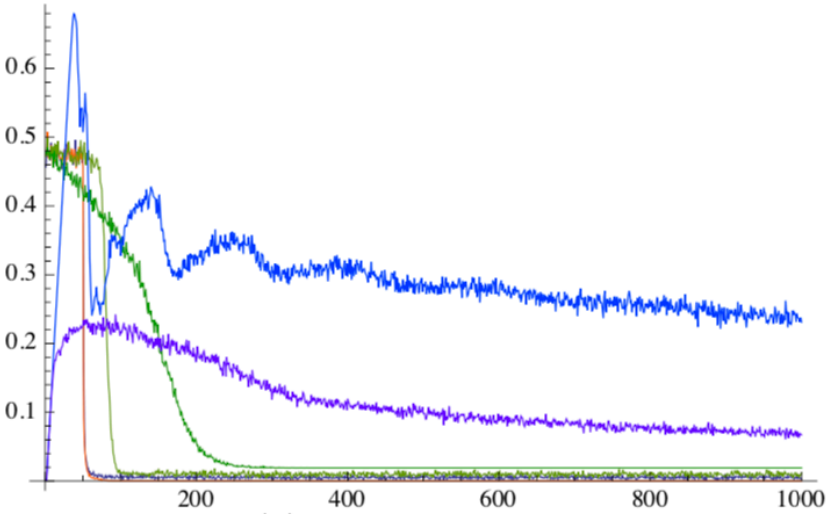
Although these complicated methods have convincing bound, some researchers find that empirically these methods can not get beyond the naïve methods.

@article{kuleshov2014algorithms,
title={Algorithms for multi-armed bandit problems},
author={Kuleshov, Volodymyr and Precup, Doina},
journal={arXiv preprint arXiv:1402.6028}, year={2014} }

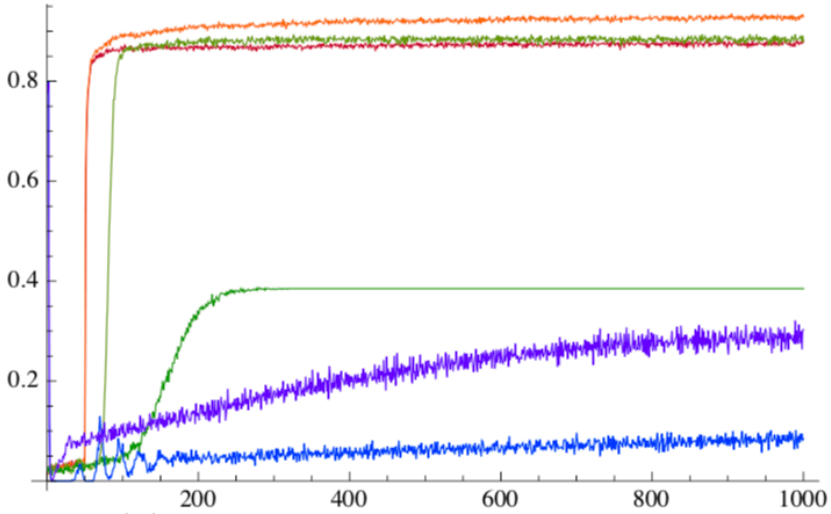
$\sigma = 0.01$

ϵ-greedy, $\epsilon = 0.005$ (29.2) — Pursuit, $\beta = 0.5$ (47.2) — UCB1 (296)

Softmax, $\tau = 0.001$ (24.3) — Reinforcement comparison, $\alpha = 0.01, \beta = 0.98$ (78.3) — UCB1-Tuned (119)



(a) Regret per turn

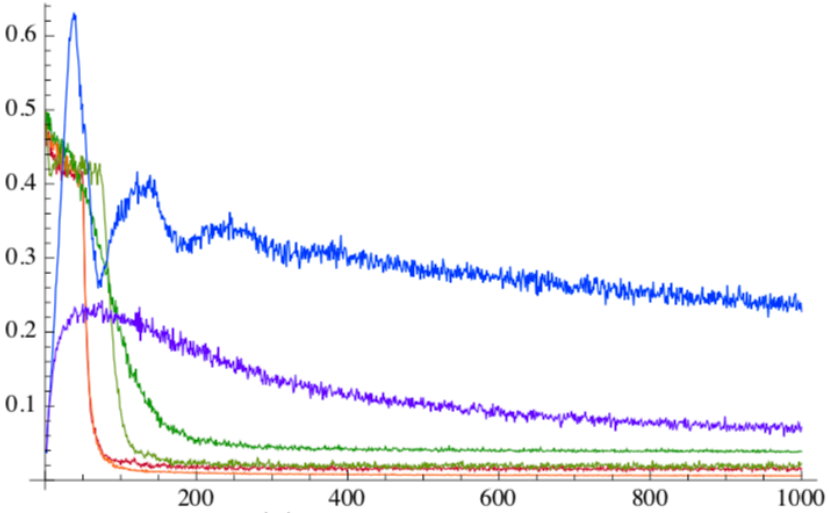


(b) Percentage of optimal arm plays

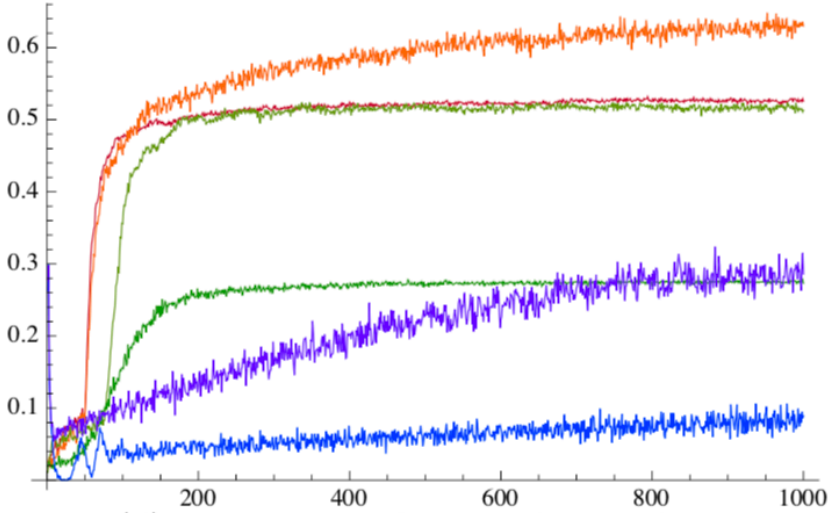
$\sigma = 0.1$

ϵ-greedy, $\epsilon = 0.005$ (39.5) — Pursuit, $\beta = 0.5$ (56.7) — UCB1 (296)

Softmax, $\tau = 0.01$ (32.0) — Reinforcement comparison, $\alpha = 0.1, \beta = 0.98$ (80.0) — UCB1-Tuned (120)



(c) Regret per turn



(d) Percentage of optimal arm plays

Exploration in RL (MDP)
- Count based methods

Keep the count of visited states or state-action pairs.

1. PAC-MDP Methods

- MBIE (model based internal estimation) [version1](1998)
  - Directed Exploration
  - Recency-based reward: $R = \frac{-t}{K_T}$,t is current time step
  - Frequency-based reward: $R = \frac{-C_{s_t}(a_t)}{K_C}$, $C_{s_t}(a_t)$ is the execution number of action a_t
  - These reward encourages agents to explore the least frequency action
  - MBIE build a transition model and use the model to compute the upper bound of Q-values

Exploration in RL (MDP)
- Count based methods

Keep the count of visited states or state-action pairs.

1. PAC-MDP Methods

- MBIE (model based internal estimation) [version2](2005)
  - Proof MBIE is PAC
  - Give the reward upper bound $\tilde{R}(s,a) = \hat{R}(s,a) + \sqrt{\frac{\ln 2/\delta_R}{2n(s,a)}}$
  - δ_R are given by Hoeffding bound
- MBIE-EB (2008)
  - Evaluate the Q use $R(s,a) + \frac{\beta}{\sqrt{n(s,a)}}$
- E^3(2002), $R_{\max}$ (2002) …

Exploration in RL (MDP)
- Count based methods

Summary idea of PAC-MDP Methods

If an agent has observed some state-action pairs sufficient times, then we can use bias inequalities such as Hoeffding bound to the empirical estimate is near to make the real dynamic model of env.

If some state-action pairs haven't be seen so many times, then assume it has a high value, which will encourage agents to try more such state-action pairs until we have an accuracy system model.

The model is used to compute the transition probability.

$$T(s'|s,a)$$

This can be seen as the **Optimism to Uncertainty**.

Exploration in RL (MDP)
- Count based methods

Keep the count of visited states or state-action pairs.

2. Bayesian Reinforcement Learning
   - Propose a concept of belief state
   - Optimal policy choose actions based on not noly how it effect the next state of the env but also the belief state.
   - Bayesian policy will very naturally trade off between exploring the system to gain more knowledge, and exploiting its current knowledge of the system
   - Usually not solvable

- Combine Beyesian and PAC-MDP (2009)
   - Evaluate Beyesian value function using reward $R(s,a) + \frac{\beta}{1+n(s,a)}$

Exploration in RL (MDP)
- Count based methods

Keep the count of visited states or state-action pairs.

3. Pseudo count methods

The contribution is **pseudo count**

- Use a hash function to decrease state space (2017)

$$R(s,a) + \frac{\beta}{\sqrt{n(\phi(s))}}$$

- Use a density model of states (2016)

$$R(x,a) + \frac{\beta}{\sqrt{\hat{n}(x)+0.01}}$$

- Use a PixelCNN to compute the pseudo count (2018)

$$R(x,a) + \frac{1}{\sqrt{\hat{n}_n(x)}}$$

Exploration in RL (MDP)
- Information Theory based methods

Use information theory concepts to compute the reduction of uncertainty or the information gain. Most use **variational inference** methods.

1. Mutual Information methods

- Maximize MI to compute the 'empowerment' (2015)

$$\epsilon(s) = \max_w \mathcal{I}^w(a, s' \mid s) = \max_w \mathbb{E}_{p(s'|a,s)w(a|s)} \left[\log \left(\frac{p(a,s'|s)}{w(a|s)p(s'|s)} \right) \right]$$

  - Can choose action to maximize empowerment or use empowerment to make reward shaping

- Minimize MI as exploration bonus (2012)
  - Take actions as representations of states and the mapping from states to actions as lossy compression
  - Find the most 'compact' action

Exploration in RL (MDP)
- Information Theory based methods

Use information theory concepts to compute the reduction of uncertainty or the information gain. Most use **variational inference** methods.

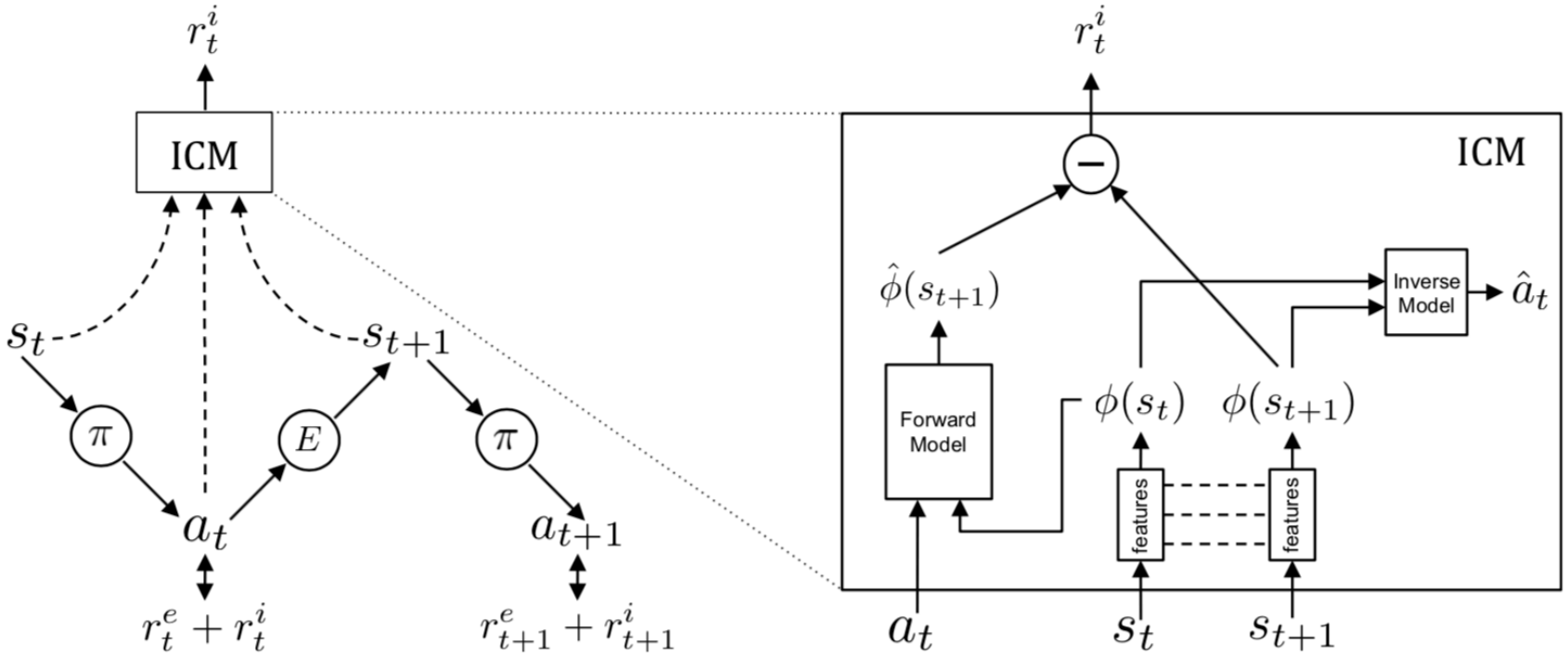
2. Information gain methods (2016)

- The IG of the belief of env
- Use IG of knowing a_t and s_{t+1} after history ξ_t as bonus

$$R(s_t, a_t) + \eta D_{KL}[p(\theta \mid \xi_t, a_t, s_{t+1} \parallel p(\theta \mid \xi_t)]$$

Exploration in RL (MDP)
- Prediction Error based methods

1. ICM+PPO (2017)



$$\frac{\eta}{2} \parallel \hat{\phi}(s_{t+1}) - \phi(s_{t+1}) \parallel_2^2$$

Exploration in RL (MDP)
- Prediction Error based methods

2. RND+PPO (SotA) (2019)

- Minimize $\|\hat{f}(\mathrm{x};\theta) - f(\mathrm{x})\|^2$ w.r.t θ
- Use prediction error as exploration bonus
- Use a separate non-episodic V and γ for intrinsic reward

$$R = R_E + R_I$$
$$V = V_E + V_I$$

Exploration in RL (MDP)
- Summary

Intrinsic Reward / Surprise / Curiosiy / Uncertainty / IG

- Count based methods
- Information theory based methods
- Prediction error methods

Exploration-Exploitation Dilemma

Exploration in high dimensional space

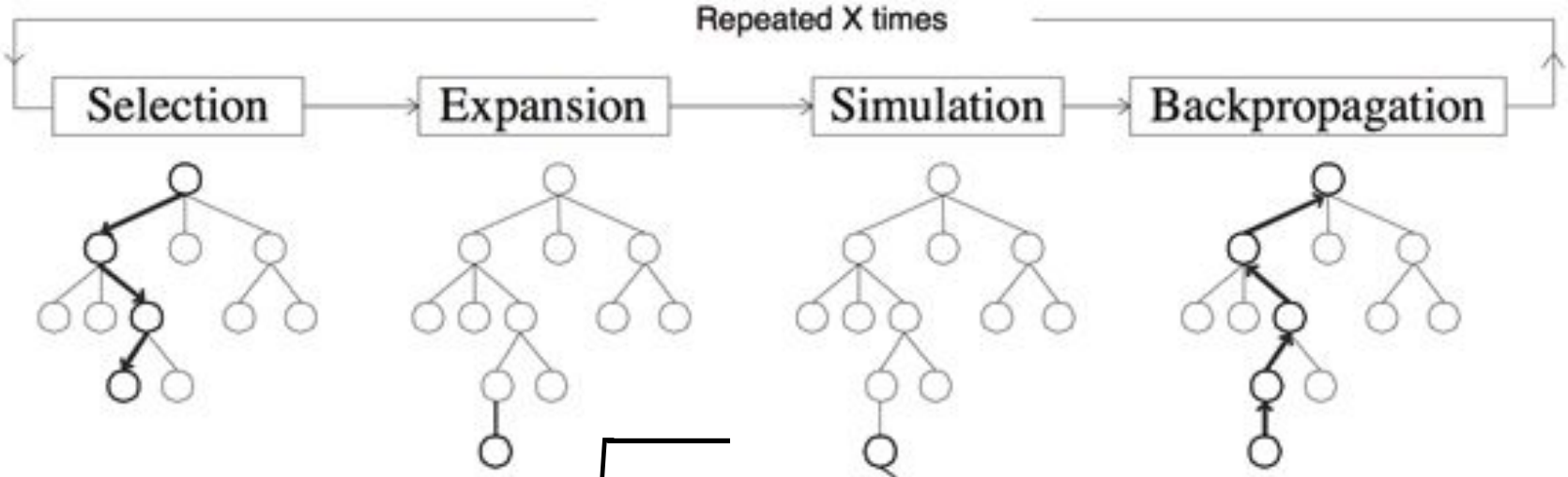
Exploration in sparse reward environment

…

Exploration in RL (MDP)
- Other Methods

1. UCT (Upper Confidence Bounds for Tree) (2006)

- MCTS + UCB
- The action selection problem as treated as a separate multi-armed bandit for every (explored) internal node.

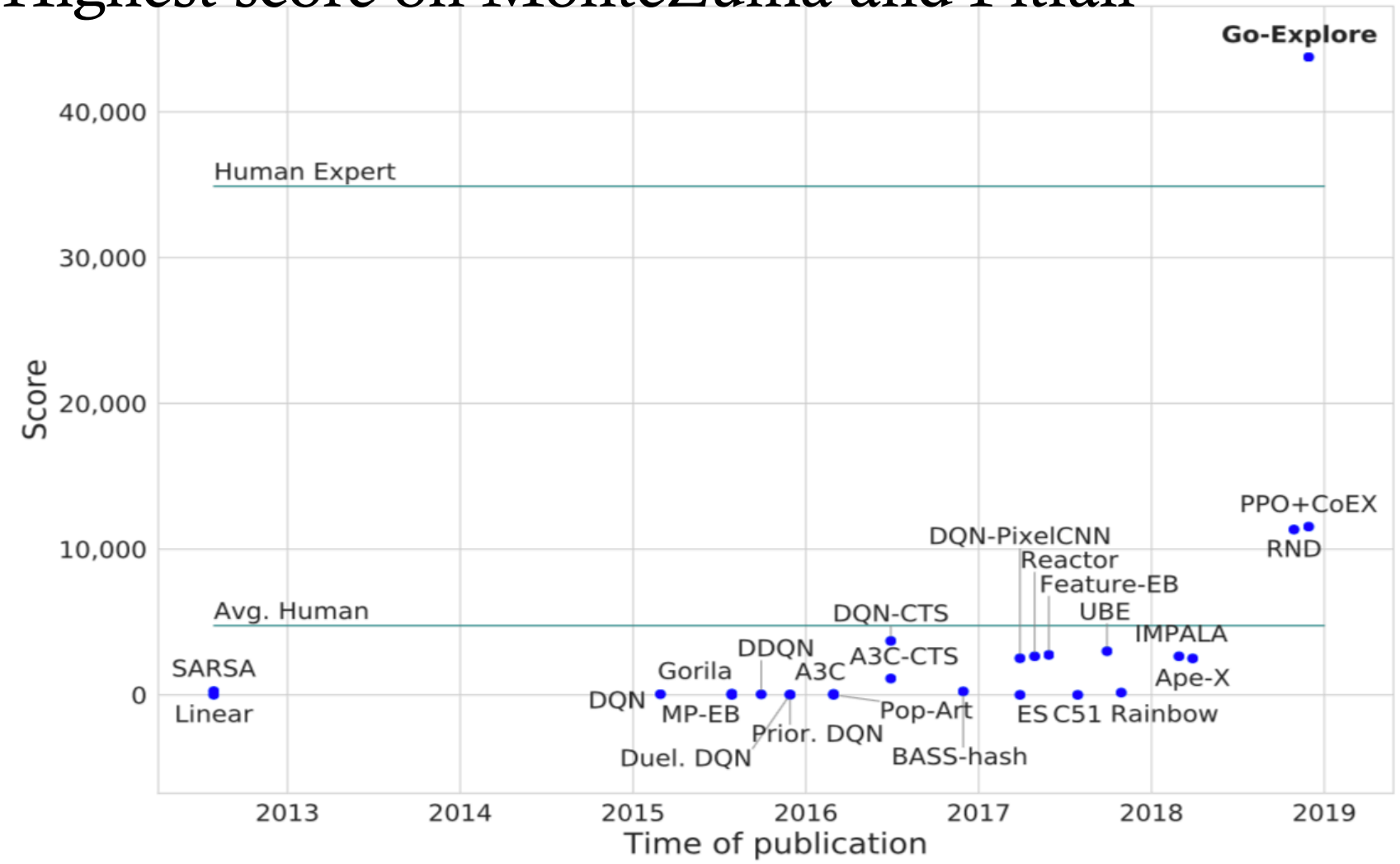


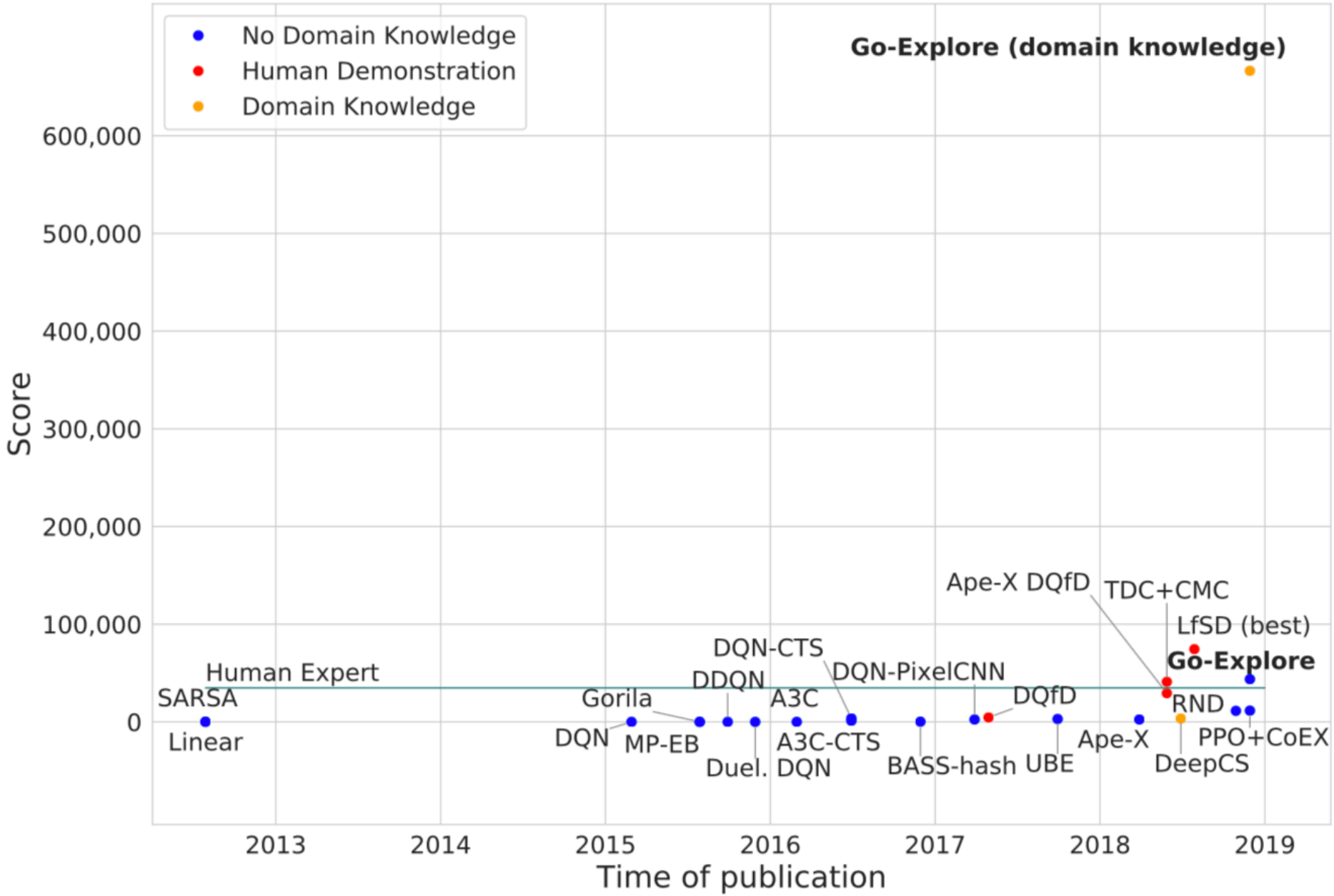
$$V_i = v_i + c \times \sqrt{\frac{lnN}{n_i}}$$

Figure from Chaslot (2006)

Exploration in RL (MDP)
- Other Methods

2. Go-Explore (Uber, 2019)
- Highest score on MonteZuma and Pitfall





Exploration in RL (MDP)
- Other Methods

2. Go-Explore (Uber, 2019)

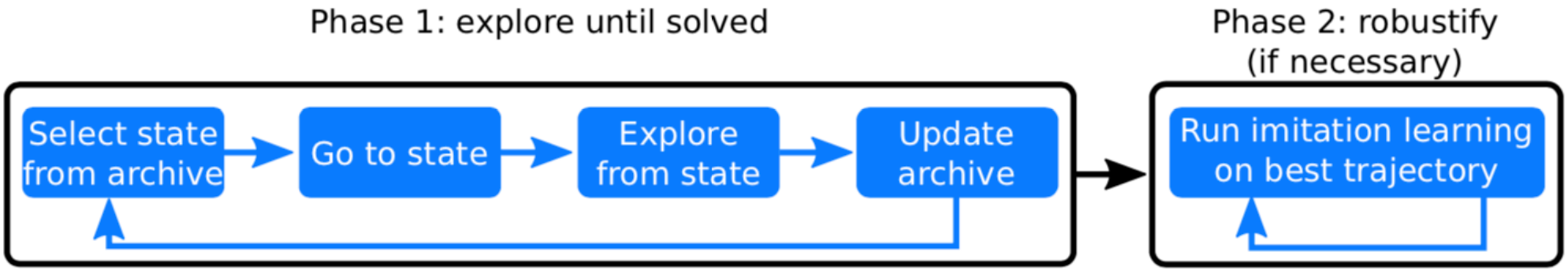


Figure 2: **A high-level overview of the Go-Explore algorithm.**

- Starting the agent near the last state in the trajectory, and then running an ordinary RL algorithm from there (in this PPO)
- Once the algorithm has learned to obtain the same or a higher reward than the example trajectory from that starting place near the end of the trajectory, the algorithm backs the agent's starting point up to a slightly earlier place along the trajectory.
- Repeats the process until eventually the agent has learned to obtain a score greater than or equal to the example trajectory all the way from the initial state.

Exploration in RL (MDP)
- Other Methods

3. CoEX (2019)

- Use attention to localize agents' position.
- PPO+CoEX / A2C+CoEX

$$r^{+}(s) = 1/\sqrt{\#(\psi(s))}$$

$\#(\psi(s))$ denotes the visitation count of the (discrete) mapped state $\psi(s)$, which consists of the contingent region (x, y)

$$(x, y) \;=\; \mathrm{argmax}_{(j,i)}\, \alpha_t(i, j)$$

$$\mathcal{R} = \mathbb{E}_{\pi}\left[\sum_t \gamma^t \left(\beta_1 r^{\mathrm{ext}}(s_t, a_t) + \beta_2 r^{+}(s_t)\right)\right]$$

- Also use intrinsic reward

Exploration in RL (MDP)
- Other Methods

4. Soft Q /Soft AC (2017 / 2018)

$$\pi^*_{\text{std}} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) \right].$$

$$\pi^*_{\text{MaxEnt}} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot|\mathbf{s}_t)) \right]$$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) \propto \exp\left(-\mathcal{E}(\mathbf{s}_t, \mathbf{a}_t)\right)$$

- α is an optional but convenient parameter that can be used to determine the relative importance of entropy and reward.
- $\mathcal{H}$ is the entropy function.

Benefits:

- improved exploration
- compositionality that allows transferring skills between tasks

Exploration in RL (MDP)
- Other Methods

5. Parameter Space Noice (2018)

For off-policy methods:

- Perturb the policy for exploration and train the non-perturbed network on this data by replaying it

For on-policy methods:

$$\nabla_{\phi,\Sigma}\mathbb{E}_{\tau}[R(\tau)] \approx \frac{1}{N}\sum_{\epsilon^i,\tau^i}\left[\sum_{t=0}^{T-1}\nabla_{\phi,\Sigma}\log\pi(a_t|s_t;\phi+\epsilon^i\Sigma^{\frac{1}{2}})R_t(\tau^i)\right]$$

$$\epsilon^i \sim \mathcal{N}(0,I) \qquad \tau^i \sim (\pi_{\phi+\epsilon^i\Sigma^{\frac{1}{2}}},p)$$

Exploration in RL (MDP)
- Other Methods

6. PCID (Policy Cover via Inductive Decoding) (2019)

- <Provably efficient RL with Rich Observations via Latent State Decoding>
- Block MDP
- Identify latent states from observed contexts
- $p(s'|s,a)$ and $q(x|s)$
- Exploration problem in episodic MDPs with rich observations generated from a small number of latent states
- Prove to be sample efficiency

Exploration in MARL (Stochastic Game) - Future Work

Intrinsic Reward?
Exploration noise?
Exploration policy?
…?

Not everything is exploration,
But exploration is in everywhere.