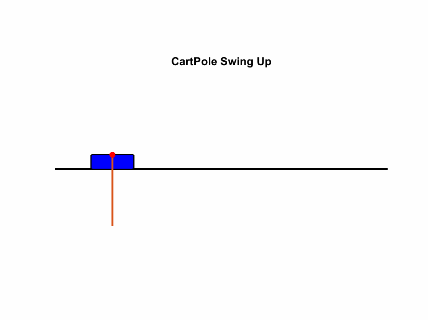
Model-Free Active Exploration in Reinforcement Learning

Alessio Russo, Alexandre Proutiere

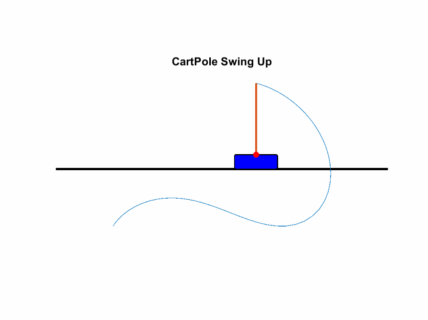
37th Conference on Neural Information Processing Systems (NeurIPS23), 2023.

Division of Decision and Control Systems
KTH Royal Institute of Technology, Stockholm, Sweden

How to efficiently explore?

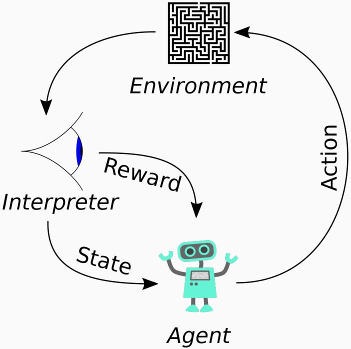


$\overset{?}{\Rightarrow}$



- Learning complex behaviors efficiently is hard... it ultimately comes down to how you explore the environment.
- What is an efficient way to explore an unknown environment?

Model

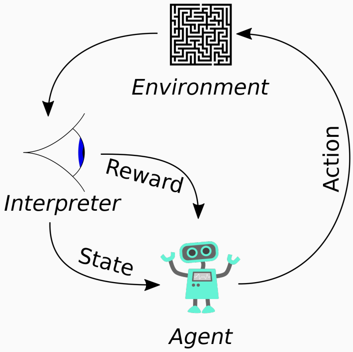


We consider an MDP $\phi = (S, A, P, q)$,

- S, A are, respectively, the state and action spaces.
- $P : S \times A \mapsto \Delta(S)$ is the transition function.
- $q : S \times A \mapsto \Delta([0, 1])$ is the reward distribution.

- Discounted value of a Markov policy π: $V^\pi(s) = \mathbb{E}^\pi[\sum_{t \geq 0} \gamma^t r_t | s_0 = s]$ with $s_{t+1} \sim P(\cdot|s_t, a_t), r_t \sim q(\cdot|s_t, a_t)$ and $a_t \sim \pi(\cdot|s_t)$. $V^\star(s) = \max_\pi V^\pi(s)$ is the optimal value.
- Action-value function of π: $Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s'}[V^\pi(s')]$ (sim. $Q^\star$).

Model



We consider an MDP $\phi = (S, A, P, q)$,

- S, A are, respectively, the state and action spaces.
- $P : S \times A \mapsto \Delta(S)$ is the transition function.
- $q : S \times A \mapsto \Delta([0, 1])$ is the reward distribution.

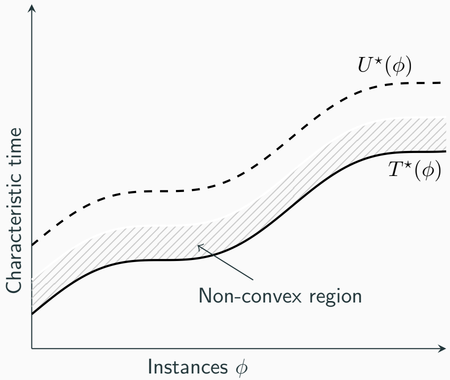
- Discounted value of a Markov policy π: $V^\pi(s) = \mathbb{E}^\pi[\sum_{t\geq 0} \gamma^t r_t | s_0 = s]$ with $s_{t+1} \sim P(\cdot|s_t, a_t), r_t \sim q(\cdot|s_t, a_t)$ and $a_t \sim \pi(\cdot|s_t)$. $V^\star(s) = \max_\pi V^\pi(s)$ is the optimal value.
- Action-value function of π: $Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s'}[V^\pi(s')]$ (sim. $Q^\star$).

Sample complexity lower bound

Sample complexity of learning an optimal policy [Marjani and Proutiere, 2021]:

$$\liminf_{\delta\to 0} \underbrace{\mathbb{E}_\phi[\tau]}_{\text{Sample Complexity}} / \log(1/\delta) \geq \underbrace{T^\star(\omega_{\text{opt}})}_{\text{Characteristic time}},$$

where $\omega_{\text{opt}} = \arg\sup_\omega T(\omega)^{-1}$ is the optimal exploration strategy.



- For a specific MDP ϕ computing the lower bound $T^\star$ is a non-convex problem.
- An alternative way is to find an upper bound $U^\star = \max_\omega U^\star(\omega)$ by convexifying the original problem.

An approximate upper bound

An approximation of this upper bound U is given by

$$U(\omega) \approx \max_{s,a\neq\pi^\star(s)} \frac{H(s,a)}{\omega(s,a)} + \frac{H^\star}{\min_{s'} \omega(s',\pi^\star(s'))},$$

where $H(s,a) := \frac{2+8\varphi^2 \mathrm{Var}_{sa}[V^\star]}{\Delta(s,a)^2}$ and $H^\star \propto \frac{\max_{s'} \mathrm{Var}_{s',\pi^\star(s')}[V^\star](1+\gamma)^2}{\Delta_{\min}{}^2(1-\gamma)^2}$.

- $\Delta(s,a) = V^\star(s) - Q^\star(s,a)$ is the *sub-optimality gap* (with $\Delta_{\min} = \min_{s,a\neq\pi^\star(s)} \Delta(s,a)$).

An approximate upper bound

An approximation of this upper bound U is given by

$$U(\omega) \approx \max_{s,a\neq\pi^\star(s)} \frac{H(s,a)}{\omega(s,a)} + \frac{H^\star}{\min_{s'} \omega(s',\pi^\star(s'))},$$

where $H(s,a) := \frac{2+8\varphi^2 \mathrm{Var}_{sa}[V^\star]}{\Delta(s,a)^2}$ and $H^\star \propto \frac{\max_{s'} \mathrm{Var}_{s',\pi^\star(s')}[V^\star](1+\gamma)^2}{\Delta_{\min}{}^2(1-\gamma)^2}$.

- $\Delta(s,a) = V^\star(s) - Q^\star(s,a)$ is the *sub-optimality gap* (with $\Delta_{\min} = \min_{s,a\neq\pi^\star(s)} \Delta(s,a)$).
- $\mathrm{Var}_{sa}[V^\star] := \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\left(V^\star(s') - \mathbb{E}_{\bar{s}\sim P(\cdot|s,a)}[V^\star(\bar{s})]\right)^2\right]$ is the variance of the optimal value.

An approximate upper bound

An approximation of this upper bound U is given by

$$U(\omega) \approx \max_{s,a \neq \pi^\star(s)} \frac{H(s,a)}{\omega(s,a)} + \frac{H^\star}{\min_{s'} \omega(s', \pi^\star(s'))},$$

where $H(s,a) := \frac{2+8\varphi^2 \mathrm{Var}_{sa}[V^\star]}{\Delta(s,a)^2}$ and $H^\star \propto \frac{\max_{s'} \mathrm{Var}_{s',\pi^\star(s')}[V^\star](1+\gamma)^2}{\Delta_{\min}{}^2(1-\gamma)^2}$.

Corollary

In the generative model the optimal allocation $\omega^\star$ satisfies

$$\omega^\star(s,a) \propto \begin{cases} H(s,a) & a \neq \pi^\star(s), \\ \sqrt{H^\star \sum_{s,a\neq\pi^\star(s)} H(s,a)/|S|} & \textit{otherwise.} \end{cases}$$

Algorithm idea

The idea is to explore according to $\omega^\star$, but we do not know $H(s,a)$ and $H^\star$!

1. Learn the Q-values and the variance of the optimal policy in a model-free way
   - Compute $\Delta_t(s,a) = V_t^\star(s) - Q_t^\star(s,a)$ and $\mathrm{Var}_{sa,t}[V_t^\star]$, where $V_t^\star(s) = \max_a Q_t^\star(s,a)$.
   - Using Δ_t and $\mathrm{Var}_{sa,t}$ compute $H_t(s,a)$ and $H_t^\star$.
2. Using these values, compute $\omega^\star$ (use certainty equivalence), and use it to explore the environment

$$\omega_t^\star(s,a) \propto \begin{cases} H_t(s,a) & a \neq \pi_t^\star(s), \\ \sqrt{H \sum_{s,a\neq\pi_t^\star(s)} H_t^\star(s,a)/|S|} & \text{otherwise.} \end{cases}$$

However, ...

Algorithm idea

The idea is to explore according to $\omega^\star$, but we do not know $H(s,a)$ and $H^\star$!

1. Learn the Q-values and the variance of the optimal policy in a model-free way
   - Compute $\Delta_t(s,a) = V_t^\star(s) - Q_t^\star(s,a)$ and $\mathrm{Var}_{sa,t}[V_t^\star]$, where $V_t^\star(s) = \max_a Q_t^\star(s,a)$.
   - Using Δ_t and $\mathrm{Var}_{sa,t}$ compute $H_t(s,a)$ and $H_t^\star$.
2. Using these values, compute $\omega^\star$ (use certainty equivalence), and use it to explore the environment

$$\omega_t^\star(s,a) \propto \begin{cases} H_t(s,a) & a \neq \pi_t^\star(s), \\ \sqrt{H \sum_{s,a\neq\pi_t^\star(s)} H_t^\star(s,a)/|S|} & \text{otherwise.} \end{cases}$$

However, ...

Algorithm idea

The idea is to explore according to $\omega^\star$, but we do not know $H(s,a)$ and $H^\star$!

1. Learn the Q-values and the variance of the optimal policy in a model-free way
   - Compute $\Delta_t(s,a) = V_t^\star(s) - Q_t^\star(s,a)$ and $\mathrm{Var}_{sa,t}[V_t^\star]$, where $V_t^\star(s) = \max_a Q_t^\star(s,a)$.
   - Using Δ_t and $\mathrm{Var}_{sa,t}$ compute $H_t(s,a)$ and $H_t^\star$.
2. Using these values, compute $\omega^\star$ (use certainty equivalence), and use it to explore the environment

$$\omega_t^\star(s,a) \propto \begin{cases} H_t(s,a) & a \neq \pi_t^\star(s), \\ \sqrt{H \sum_{s,a\neq\pi_t^\star(s)} H_t^\star(s,a)/|S|} & \text{otherwise.} \end{cases}$$

However, ...

Boostrapped MF-BPI

- $\omega_t^\star(s,a)$ explores according to the current estimate of the aleatoric uncertainty of the MDP $(\Delta_t(s,a), \mathrm{Var}_{sa,t}[V^\star])$. It does not account for parametric uncertainty (uncertainty of the model).
- MDP-NAS [Marjani and Proutiere, 2021] requires a forced exploration step (e.g., mix $\omega^\star$ with a uniform distribution) to reduce the parametric uncertainty asymptotically.
- Instead, we quantify the parametric uncertainty about $Q^\star, \mathrm{Var}_{sa}[V^\star]$ using an ensemble of models.
  - We approximately sample from this uncertainty and use it to compute $\omega_t^\star$.

Boostrapped MF-BPI

- $\omega_t^\star(s,a)$ explores according to the current estimate of the aleatoric uncertainty of the MDP ($\Delta_t(s,a), \mathrm{Var}_{sa,t}[V^\star]$). It does not account for parametric uncertainty (uncertainty of the model).
  - Parametric uncertainty: lack of data, random parameters initialization, randomness in the algorithm, etc...
- MDP-NAS [Marjani and Proutiere, 2021] requires a forced exploration step (e.g., mix $\omega^\star$ with a uniform distribution) to reduce the parametric uncertainty asymptotically.
- Instead, we quantify the parametric uncertainty about $Q^\star, \mathrm{Var}_{sa}[V^\star]$ using an ensemble of models.
  - We approximately sample from this uncertainty and use it to compute $\omega_t^\star$.

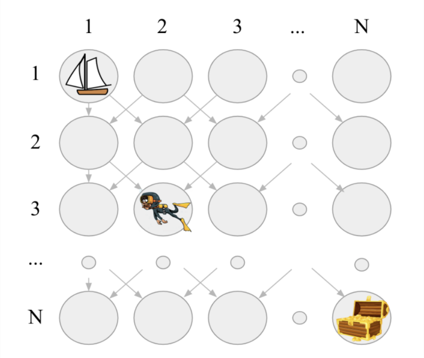
Boostrapped MF-BPI

- $\omega_t^\star(s,a)$ explores according to the current estimate of the aleatoric uncertainty of the MDP ($\Delta_t(s,a), \mathrm{Var}_{sa,t}[V^\star]$). It does not account for parametric uncertainty (uncertainty of the model).
- MDP-NAS [Marjani and Proutiere, 2021] requires a forced exploration step (e.g., mix $\omega^\star$ with a uniform distribution) to reduce the parametric uncertainty asymptotically.
- Instead, we quantify the parametric uncertainty about $Q^\star, \mathrm{Var}_{sa}[V^\star]$ using an ensemble of models.
    - We approximately sample from this uncertainty and use it to compute $\omega_t^\star$.

Boostrapped MF-BPI

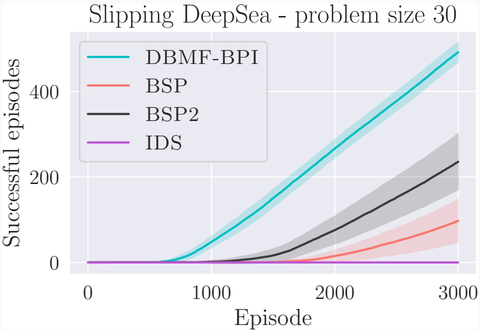
- $\omega_t^\star(s,a)$ explores according to the current estimate of the aleatoric uncertainty of the MDP $(\Delta_t(s,a), \mathrm{Var}_{sa,t}[V^\star])$. It does not account for parametric uncertainty (uncertainty of the model).
- MDP-NAS [Marjani and Proutiere, 2021] requires a forced exploration step (e.g., mix $\omega^\star$ with a uniform distribution) to reduce the parametric uncertainty asymptotically.
- Instead, we quantify the parametric uncertainty about $Q^\star, \mathrm{Var}_{sa}[V^\star]$ using an ensemble of models.
  - We approximately sample from this uncertainty and use it to compute $\omega_t^\star$.

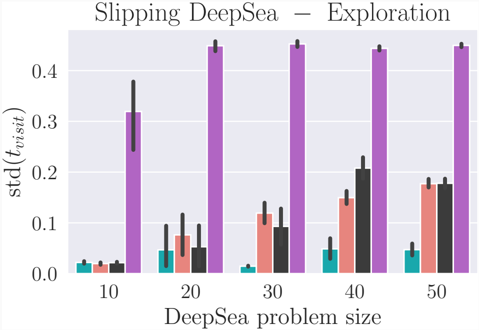
Environment: Slipping DeepSea problem



- Only diagonal movements + negative reward at each step (except for the last row).
- Last row: zero reward unless the agent reaches the last column.
- Probability of *slipping*, i.e. the agent goes in the wrong direction, is 5%.

Results: DeepSea problem





Slipping DeepSea problem. On the left: total number of successful episodes for a grid 30×30. On the right: standard deviation of t_{visit} at the last episode, depicting how much each agent explored (the lower the better).

Conclusion

Exploration needs to be tailored according to the difficulty of the underlying MDP:

- Leverage instance-specific results.
- Explore according to both aleatoric $(\Delta(s, a), \mathrm{Var}_{sa}[V^\star])$ and parametric uncertainty.

Check the paper for more results and information! Thank you for listening!