Explicit Explore-Exploit Algorithms in Continuous State Spaces

Mikael Henaff
Microsoft Research NYC

Introduction

- Many RL algorithms have high sample complexity
- Explicit Explore-Exploit ($E^3$): first provably sample-efficient RL algorithm
- Sample complexity: $\text{poly}(|S|, |A|, H)$

![Diagram](image)

- Assumes small discrete $S$
- Goal: handle large/infinite $S$

Algorithm

Compute exploration policy
(high disagreement in model predictions)

\[ \pi_{\text{explore}} = \arg\max_{\pi \in \Pi} v_{\text{explore}}(\pi, M_i) \]

Gather experience

\[ a_t \sim \pi_{\text{explore}}(s_t), s_{t+1} \sim P_{M_i}(\cdot|s_t, a_t) \]

\[ R_t \leftarrow R_t \cup \{(s_t, a_t, s_{t+1})\}_t \]

Refine dynamics models

\[ M_{i+1} = \{M \in \mathcal{M}_i : W(\pi_{\text{explore}}, M, R) \leq \delta \text{ on } R\}, \]

\[ M_{i+1} = \text{ensemble of models with low error on } R \]

DREEM (idealized)

- Start with full version space
- Exact policy search
- Eliminate models over time

Optimize models over time

- Ensemble of neural nets
- MCTS/DQN
- Neural-E3

Explore

\[ \pi_{\text{exploit}} = \arg\max_{\pi \in \Pi} v_{\text{exploit}}(\pi, M_i, R) \]

Exploit

\[ \pi_{\text{exploit}} = \arg\max_{\pi \in \Pi} v_{\text{exploit}}(\pi, M_i, R) \]

Theorem

Assume that $M^* \in \mathcal{M}$. With probability at least $1 - \delta$, DREEM outputs an $\epsilon$-optimal exploitation policy after collecting at most $O\left(\frac{H^d |\mathcal{A}|^d \log (|\mathcal{M}|/\delta)}{\epsilon^d} \right)$ samples, where $d$ is the max rank of the misfit matrices.

Use ranks of model misfit matrices as complexity measure: $d = \max_{M \in \mathcal{M}} \text{rank}(A_h)$

Proof sketch (simplified, errors are 0/1):

- High disagreement between $M, M' \implies$ at least one must have high error
- At iteration $t$, there is a model in $M_t$ with high error or all models give a good exploitation policy
- Row $\pi_t$ of $A_h$ is linearly independent of rows of previous $\pi_t \implies$ at most $\text{rank}(A_h) \leq d$ iterations

Experiments

- Small ensemble of NN models to approximate version space (4-8 models)
- MCTS/BFS for planning during exploration
- DQN on replay buffer for exploitation

Related Work

- Kearns and Singh, 2002
- Error matrices: Jiang et al, 2017; Sun et al, 2019
- Practical algorithm: Shyam et al, Pathak et al 2019

Links

- Code: https://github.com/mbenhaff/neural-e3
- Contact: mibenaff@microsoft.com