Explicit Explore-Exploit Algorithms in Continuous State Spaces
Mikael Henaff
Microsoft Research

Introduction

- Many RL algorithms have high sample complexity
- This can limit their practical application
- Explicit Explore-Exploit (E3): First provably efficient RL algorithm [1]
- Sample complexity polynomial in |S|
- Model-based:
  - Use current model to plan to reach unknown states
  - Once model is sufficiently accurate, optimize rewards
- Assumes small discrete state space

E3 with continuous states

- Instead of maintaining set of states, maintain set of dynamics models M_i
- Treat states which induce high disagreement between models in M_i as unknown
- Execute policies which induce high disagreement
- No dependence on number of states

Algorithm

Model class M_i, policy class Π
D(π, M, M') = ||P^π,M_i(·) - P^π,M'_i(·)|| TV

\[ v_{\text{exploit}}(\pi, M) = \max_{M'' \in M, M'' \not= M} \sum_{h=1}^{H} D(\pi, M, M'', h) \]

Algorithm 1 \( (M, \Pi, n, \epsilon, \phi) \)
\[ M_{k+1} \leftarrow M_k \]
\[ R \leftarrow \emptyset \]
for \( t = 1, 2, \ldots \) do
\[ \pi_{\text{exploit}} = \text{argmax}_{\pi_{\text{exploit}}} v_{\text{exploit}}(\pi_{\text{exploit}}, M_k) \]
if \( v_{\text{exploit}}(\pi_{\text{exploit}}, M_k) > \frac{\epsilon}{2} \) then
Add \( n \) trajectories following \( \pi_{\text{exploit}} \) to \( R \)
\[ M_{k+1} \leftarrow \text{UpdateModelSet}(M_k, R, \phi) \]
else
Choose any \( M \in M_k \)
\[ \pi_{\text{exploit}} = \text{argmax}_{\pi} v_{\text{exploit}}(\pi, M) \]
Halt and return \( \pi_{\text{exploit}} \)
end if
end for

Analysis

Use rank of model misfit matrices as complexity measure:
\[ d = \max \text{rank}(A_h) \]

Theorem

Assuming that \( M^* \in M_i \) for any \( \epsilon, \delta \in (0, 1] \) and appropriate \( n, \phi \), with probability at least \( 1 - \delta \), Algorithm 1 outputs a policy \( \pi_{\text{exploit}} \) such that \( v_{\text{exploit}}(\pi_{\text{exploit}}, M) \geq v^\star - \epsilon \). The number of trajectories collected is at most \( O\left( \frac{n^2 d^2}{\epsilon^2} \log \left( \frac{|M|}{d} \right) \right) \), where \( d \) is the rank of the misfit matrix.

Proof sketch (simplified, errors are \( 0/1 \)):
- High disagreement between \( M, M' \implies \) at least one must have high error
- At any iteration, there exists a model with high error or all models give a good exploitation policy
- At iteration \( t \), row \( \pi_t \) of \( A_h \) linearly independent of rows of previous \( \pi_{t-1} \implies \) at most rank \( (A_{t-1}) \leq d \) iterations

Experiments

Environments tested (require exploration):
- Stochastic combination lock
- Mazes
- Classic Control

References


Links

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