Efficient Exploration Using Expert Knowledge

CS748: Midterm Presentation - T6

Ishank Juneja

25^{th} February 2020

Introduction - Problem Formulation

- Effective and efficient exploration is key to learning complex behaviours
- An agents experience with prior tasks should make it easier to adapt to new related tasks

Can we inject Expert-Knowledge in the form of an initialisation policy π^e to achieve better sample efficiency for PAC-RL algorithms?
That is obtain an algorithm $\mathcal{A}$ such that,

$$T^{\mathcal{A}}(\{S, A, T, R, \gamma, \epsilon, \delta, \pi^e\}) < T^{PAC-RL}(\{S, A, T, R, \gamma, \epsilon, \delta\})$$

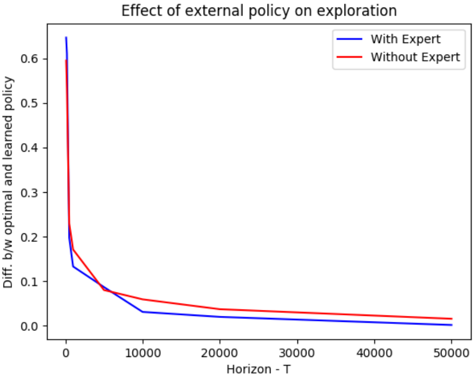
For certain good policies π^e

Experimental Validation

- Started off by validating idea of using a good policy to reduce sample complexity required in learning an $\epsilon - \delta$ optimal policy
- Idea: Trust external policy with probability α
- With probability $1 - \alpha$, uniformly sample all actions $a \in A$ on landing on a certain state $s \in S$ until learned model becomes valid
- Perform ϵ-greedy exploration thereafter - continuing to trust external policy with probability α
- Have implemented PAC baseline Model Based Action Elimination (Even-Dar et al.) as well but not reported - implementation issues

Validation on an MDP

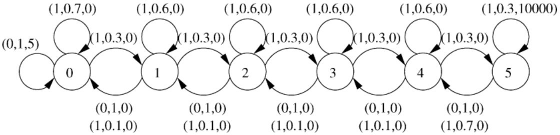
- First the sample efficiency of incorporating expert knowledge through an external policy was compared to the vanilla exploration approach
- Incorporating expert knowledge speeds up the model learning process significantly. MDP-10 states, 5 actions (From CS747 PA2)

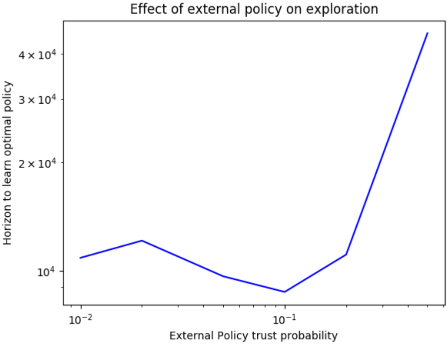


$\pi^* =$
$[0, 1, 0, 1, 4, 3, 1, 0, 1, 1]$
$\pi^e =$
$[0, 0, 0, 1, 4, 3, 1, 0, 0, 0]$

Validation River-Swim Task

- Next the idea was validated on the RiverSwim MDP





$\pi^* = [1, 1, 1, 1, 1, 1]$
$\pi^e = [0, 1, 1, 1, 0, 0]$
In plot trust probability α on the horizontal axis and the horizon T required to learn the optimal policy on the vertical.
Benefit from trusting the external policy just the right amount.

Systematic Policy weighting

- Systematically adapting trust - A Bandit Problem?
- Regret minimization under the *Adversarial Bandit* setup
- Stochastic MAB - Assumes rewards generated from stationary distributions
- No such assumptions under in Adversarial MAB setting
- Bandit instance corresponds to all reward realizations for all arms at all time steps (In Hindsight)

Exp3 Algorithm - Adversarial Bandit Algorithm

- Exp3 - **Exp**onential-weight algorithm for **Exp**loration and **Exp**loitation
- At every time step Exp3 does the following
  1. Sample an arm based on a previously computed distribution
  2. Estimate rewards for all actions based on the observed reward
  3. Use the estimated rewards to update selection distribution
- Reward estimation:

$$\hat{X}_{ti} = \frac{\mathbb{1}\{A_t = i\}}{P_{ti}} X_t$$

$$P_{ti} \doteq \frac{\exp(\eta \hat{S}_{t-1,i})}{\sum_j \exp(\eta \hat{S}_{t-1,j})} .$$

Application to MDP Exploration

- Pit policies produced by PAC-RL exploration algorithm and external expert against each other
- Let weights adapt over time in online fashion in accordance with Exp3
- Estimate Rewards associated with both *arms* by performing fixed length roll-outs