Blind Decision Making: Reinforcement Learning with Observation Delays

Mridul Agarwal, and Vaneet Aggarwal,
Purdue University

Background and Motivation

- Consider a robot with multiple sensors and actuators. The states acquired from the sensors reach the brain with some communication delays. The brain adds additional delays while computing the optimal action which is then finally executed by the actuator.
- We ask, how to tackle the issue where the agent is required to make decisions with old state information
- Consider an MDP M with state space S, and action space A. The agent is maximizing sum of discounted rewards with discount factor γ.
- The agent , at time t observes state s_{t-d} which is the state d steps in the past

Existing Algorithms and Learnings

- Extended MDP Approach [1]: Consider an MDP with prior actions stacked to last known state as augmented state,

$$\tilde{s}_t = (s_{t-d}, a_{t-d}, \cdots, a_{t-1})$$

- Model Based Simulation Approach [2]: Play the action optimal for the most likely state $a^* = \arg_a \max Q(s'_t, a)$

$$s'_t = \arg_s \max P(s|s_{t-d}, a_{t-d}, \cdots, a_{t-1})$$

- Memory Less Approaches [3]: Train the agent using effective state and actions as $s_t = s_{t-d}, a_t = a_t$

Key Difficulties

- Allow the model to handle stochastic delays
- Maintaining performance without expanding the state space exponentially
- Deal with low individual state probabilities

Key Ideas

- Optimize for the expected value of the unknown state s_t, $\mathbb{E}[V(s_t)]$.
- Select action a_t as $a_t = \arg_a \max \mathbb{E}[Q(s_t, a)]$
- To bound the performance gap, observe that the expected reward received is comes from the

EQM Algorithm

Algorithm 1 *Get_EQM_Action*

Input: $\mathcal{S}$, $\mathcal{A}$, $\hat{p}$, Q, $(s_{t-d}, a_{t-d}, \cdots, a_{t-1})$
Output: Estimated greedy action a_t
$\bar{\mathbf{p}} = [0, \cdots, 0]$, vector of length $|\mathcal{S}|$
$\bar{\mathbf{p}}[s_{t-d}] = 1$
for $0 \leq k < d$ **do**
  $\bar{\mathbf{p}} = \left(\hat{p}(:, a_{t-(d-k)}, :)\right)^T \bar{\mathbf{p}}$
end for
$\bar{\mathbf{Q}} = \bar{\mathbf{p}}^T Q$
Return $\arg\max_a \bar{\mathbf{Q}}$

- The complexity of Algorithm 1 becomes $O(d|S|^2 + |A|)$ for matrix multiplications inside the for loop and finding the best action for the arg max

Performance Guarantee

- [Lemma 2 ,4]: The value function of any policy of the extended MDP is the expected value of the current state, or

$$V(\tilde{s}_t) = \mathbb{E}_{s_t}[V(s_t)|\tilde{s}_t]$$

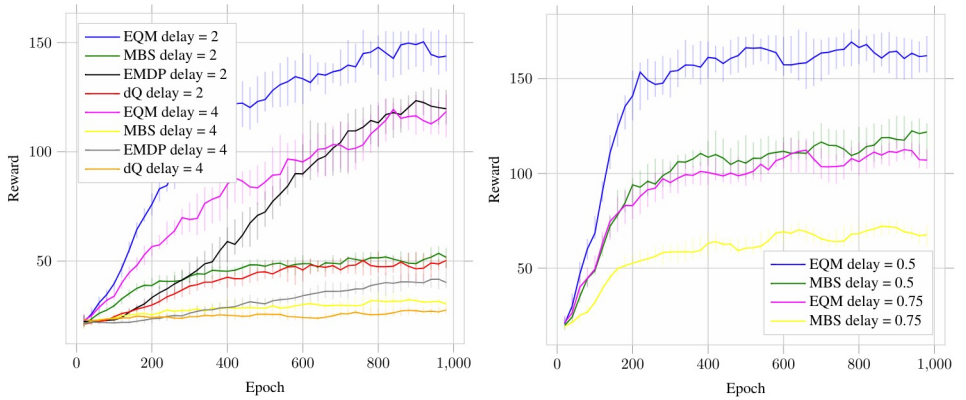
- [Theorem 1, 4]: The value of EQM policy $\tilde{\pi}$ satisfies:

$$V^{\tilde{\pi}}(\tilde{s}_t) = \mathbb{E}_{s_t}\left[V^{\pi^*}(s_t)\middle|\tilde{s}_t\right] - \frac{1}{(1-\gamma)^2}\left(1 - \frac{1}{|A|}\right)$$

where π^* is the policy which an oracle uses to determine the optimal action.

Simulation Results

- To empirically evaluate the EQM algorithm, we use Cart Pole environment. We discretize the continuous states of the environment to 10,000 states. The number of actions become 2.
- We run the algorithm for both, constant and stochastic delays. We compare against the Extended MDP approach [1], MBS algorithm [2], and dQ algorithm [3].
- We observe that the performance of EQM algorithm does not decays much even on doubling the delays.



(b) Constant delays (a) Stochastic delays

References

[1] Altman, E.; and Nain, P. 1992.Closed-loop control with delayed information, volume 20. ACM
[2] Walsh, T. J.; Nouri, A.; Li, L.; and Littman, M. L. 2009.Learning and planning in environments with delayed feed-back. AAMAS 18(1):83.
[3] Schuitema, Erik, et al. "Control delay in reinforcement learning for real-time dynamic systems: a memoryless approach." *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010.
[4] Agarwal, M., and V. Aggarwal. "Blind Decision Making: Reinforcement Learning With Delayed Observations". *Proceedings of the ICAPS*, vol. 31, no. 1, May 2021, pp. 2-6,