Bayesian Reinforcement Learning under Uncertain Transition Dynamics

Haochen Wu
Mechanical Engineering
University of Michigan, Ann Arbor

***Abstract*—Bayesian reinforcement learning maintain a hidden belief state over the uncertainty of system transition dynamics. It provides the opportunity for the agent to act optimally under the dilemma of exploration and exploitation. A formulation of Bayesian reinforcement learning in discrete space is presented. An online algorithm training the policy while learning the transition dynamics is proposed by combining Thompson Sampling approximation and Experience Replay mechanism. The algorithm is implemented on the classic Chain problem and a simple practical fire fighting problem. The regret analysis is conducted for both cases to show the performance of the algorithm.**
***Keywords*: Bayesian Inference, Markov Decision Process, Bayesian Reinforcement Learning, Planning under Uncertainties**

I. Introduction

While planning and making decisions under uncertainties, one always faces the dilemma that whether to explore to gain more information or to exploit to benefit from the current best decision. Reinforcement learning is often modeled as a Markov Decision Process (MDP) [1] that finds the optimal policy or sequence of decisions that maximizes the expected reward of achieving the goal. However, since the system is dynamic and sometimes unpredictable, the state transition probability is often uncertain. The advantage of Bayesian approach in reinforcement learning is that the belief of the state transition probability could be modeled and maintained throughout the learning process. Then the trade-off between exploration and exploitation is naturally optimized.

Bayesian reinforcement learning (BRL) [2] framework parameterize the transition model and the agent is maintaining the beliefs over the transition probabilities. With this formulation, the agent would never take random actions. Instead, the agent would take the best action based on the current beliefs on the transition model.

Solving the Bayesian reinforcement learning problem to get the offline policy in the belief space is computationally intractable. The contributions of this project are the following:

1) Formulate the Bayesian reinforcement learning in continuous belief space and provide the equivalent representation in discrete space.
2) Provide an online solution for the agent to balance exploration and exploitation while inferring transition parameters using Thompson Sampling [3] approximation.
3) Treat the online planning as a trial and embed the Experience Replay [4] mechanism to train an experienced intelligent agent.

The rest of the report is organized as: Section II provides the background on Markov Decision Process and Bayesian Reinforcement Learning. Section III discusses the formulation of BRL and proposes an online training algorithm. Section IV analyzes the performance of the algorithm in two scenarios.

II. Preliminaries

A. Markov Decision Process

In literature, the reinforcement learning (RL) problem is usually modeled as the Markov Decision Process (MDP) which is described in depth in [1][5]. MDP is a popular method for addressing action uncertainty. It is a decision-making framework in which the action uncertainty is modeled using a stochastic state transition function and the state of problem is perceived perfectly by the agent.

1) MDP Definition: An MDP problem $\mathcal{M}$ is defined as a 4-tuple $\mathcal{M} =< S, A, T, R >$, where $S := s$ is a finite set of discrete states of the world, $A := a$ is a finite set of actions available for the agent. The stochastic transition dynamics or the transition model $T : S \times A \times S \to [0, 1]$ is given by:

$$T(s, a, s') = Pr(s'|s, a)$$

which is the probability of transitioning to the next state s' conditioned on given current state s and action taken a. The reward function $R : S \times A \to \mathbb{R}$ is the reward for taking action a at state s.

Notice that the next state and the expected reward depend only on the previous state and the action taken, which provides the Markov property — the state and reward at time $t + 1$ is dependent only on the state and action at time t.

2) Value and Policy: The MDP problem is seeking an optimal policy $\pi^* : S \to A$ that maximizes the expected total rewards $V(\pi)$ in a planning horizon t_h,

$$\pi^* = argmax_\pi V(\pi)$$

where

$$V(\pi) = \mathbb{E}[\sum_{t=0}^{t=t_h} \gamma^t R(s_t, a_t)|\pi]$$

and $\gamma \in (0, 1)$ is a discount factor that indicates earlier reward has more value and ensures the sum is finite. Employing Bellman equation, the Q-Value for a state-action pair is defined as the one-step value of state s if an action is taken:

$$Q(s, a) = R(s, a) + \sum_{s' \in S} T(s, a, s')V(s')$$

where $V(s')$ is the accumulated reward for state s'. Under optimal policy π^*, the optimal Value $V_{\pi^*}(s)$ can be determined by

$$V_{\pi^*}(s) = V^*(s) = max_{a \in A} Q(s, a), \forall s \in S \quad (1)$$

and the optimal policy could be extracted as

$$\pi^*(s) = argmax_{a \in A} R(s, a) + \gamma \sum_{s' \in S} T(s, a, s')V^*(s') \quad (2)$$

To solve the MDP problem, value iteration and policy iteration [5] are the most commonly used strategies.

B. Bayesian Reinforcement Learning

In the ideal case as described in MDP, the transition dynamics is known for each state-action pair. However, for the cases where the agent is placed into an environment for the first time, the transition dynamics of taking an action is often unknown. Take playing a game as an example, the probability of being in the next state under an action is unknown. Then the agent would face a dilemma that whether to explore to gain more information about the environment of to exploit to benefit from the current best decision. Bayesian Reinforcement Learning (BRL) leverages methods from Bayesian Inference to incorporate information into the learning process by treating unknown parameters as the observable states and maintaining belief about them [2].

1) Partially Observable MDP: One way to planning an MDP with unknown parameters is to plan under uncertainties and formulate the problem as a Partially Observable MDP (POMDP) [6]. However, solving the POMDP is computationally intractable because it maps infinite belief states to actions. Online algorithms [7] under POMDP formulation have been proposed to compute efficiently and minimize the amount of exploration.

2) Bayesian-Adaptive Markov Decision Process: Under certain assumptions about the form of the unknown model parameters, the BRL problem could be modeled as a Bayesian-Adaptive MDP (BAMDP) [8] that is mathematically equivalent with POMDP formulation but with a set of discrete states. However, the size of the state grows quickly since it treats each model parameter as a part of the world state.

III. Methods

A. Problem Formulation

Bayesian reinforcement learning problem can be formulated by augmenting the physical world state and the belief state on the distribution of unknown parameters - transition dynamics in this case. The problem is defined as a tuple $\overline{\mathcal{M}} =< \overline{S}, A, \overline{T}, R >$, where

$\overline{S} := S \times B, (s, b) \in \overline{S}$

S : physical states of the world

B : belief states of the transition parameters θ, $b(\theta)$

S, A, R are the same as defined in Section II-A1

$\overline{T} := S \times B \times A \times S \times B \to [0, 1]$

$\overline{T}(s, b, a, s', b') = Pr(s', b'|s, b, a)$

The objective of this problem is to find the policy $\pi := S \times B \to A$. The optimal policy under Bellman's principle over the augmented state can be computed as:

$$\pi^*(s, b) = argmax_{a \in A} Q(s, b, a) \quad (3)$$

where

$$Q(s, b, a) = R(s, a) + \gamma \sum_{(s', b') \in \overline{S}} \overline{T}(s, b, a, s', b')V^*(s', b')$$

The transition probability $\overline{T}$ can be factored into two conditional probabilities by chain rule as:

$$Pr(s', b'|s, b, a) = Pr(s'|s, b, a)Pr(b'|s, b, a, s') \quad (4)$$

The first term could be easily determined under given belief over the transition model parameters. In the second term, b' is the posterior distribution over transition parameters given prior b and the data (s, a, s') received by interacting with the environment. $Pr(b'|s, b, a, s') = 1$ if $b'(\theta) = b(\theta|s, a, s')$, and 0 otherwise.

B. Dirichlet Distribution

The transition parameters θ has its special properties. $\theta := S \times A \times S \to [0, 1]$ determines the probability of transitioning to s' given s, a. Let $\theta_{a,s}^{s'}$ denote each transition parameter. It satisfies $\sum_{s' \in S} \theta_{a,s}^{s'} = 1, \forall s \in S, a \in A$, which is the standard $K - 1$ simplex ($K = |S|$). Therefore, for each state-action pair, the transition parameters could be modeled as a Dirichlet Distribution

denoted as $Dir(\alpha)$ and parameterized by α - a vector of K real numbers. $b(\theta_{s,a}) \sim Dir(\alpha)$

C. Belief Monitoring

Inferring the transition parameters requires the belief update once the transition data (s, a, s') is received. As Dirichlet Distribution belongs to the exponential family, it is a conjugate prior so that the resulting posterior is also a Dirichlet Distribution. With this property, the belief over the transition parameters can be updated by Bayes Theorem:

$$b'(\theta_{s,a}) \propto Pr(s'|s, a, \theta_{s,a}) b(\theta_{s,a}|s, a, s') \tag{5}$$

Since the belief over transition parameters can be parameterized by a vector α, let $\alpha_{s,a} \in \mathbb{N}^K$ represent the parameterized Dirichlet Distribution for each state-action pair. Equation 5 becomes:

$$\begin{aligned} Dir(\alpha'_{s,a}) &\propto \theta_{s,a}^{s'} Dir(\alpha_{s,a}) \\ &\propto \theta_{s,a}^{s'} \prod_{s'' \in S} \theta_{s,a}^{s''\ \alpha_{s,a}^{s''}-1} \end{aligned}$$

which is equivalent to:

$$\alpha_{s,a}^{s'} = \alpha_{s,a}^{s'} + 1 \tag{6}$$

With this special property of Dirichlet Distribution, the belief over transition parameters can be modeled in discrete space, and the transition model $Pr(s'|s, a)$ can be computed as $\frac{\alpha_{s,a}^{s'}}{\sum_{s''} \alpha_{s,a}^{s''}}$.

D. Proposed Algorithms

Instead of solving the dilemma of exploration and exploitation by two objectives, the Bayesian reinforcement learning problem has only one objective which is to maximize the total expected rewards in a planning horizon t_h:

$$V_\pi(s, b) = \mathbb{E}[\sum_{t=0}^{t=t_h} \gamma^t R(s_t, a_t)|\pi]$$

Since the transition dynamics of the problem is unknown, the agent has to actually interact with the environment to get transition data while selecting the best action. Computing the optimal policy for exploration and exploitation decision at each (s, b) state is intractable. Therefore, an online partial planning algorithm adapting Thompson Sampling is proposed below.

1) Thompson Sampling for Bayesian RL: The Thompson Sampling (TS) algorithm provides a natural Bayesian approach to the problem using randomized probability matching [3]. The algorithm chooses the action that maximizes the expected reward with respect to the samples from the posterior belief. The TS algorithm for Bayesian RL is summarized in Algorithm 1. For each iteration, transition parameter samples are taken from the Dirichlet Distribution for each state-action pair. Line 3-7 solves MDP for each sample and finds the best action to take under average Q-Value. Line 8 executes the action and interacts with the environment. Line 9 updates the belief as discussed in Section III-C.

Algorithm 1 Thompson Sampling for Bayesian RL

Inputs: s, $\{\alpha_{a,s}\}$

1: **while** max iteration is not reached **do**
2: Sample $\theta_{a,s}^k$ from $\alpha_{a,s}, \forall a \in A, s \in S$
3: **for** $i = 1 : k$ **do**
4: $Q_i(s, a) = solveMDP(\theta^i)$
5: **end for**
6: $\overline{Q}(s, a) = \frac{1}{k} \sum_i Q_i(s, a)$
7: $a^* = argmax_a \overline{Q}(s, a)$
8: Execute a^* and receive data (s, a, s', r)
9: $\alpha_{s,a}^{s'} = \alpha_{s,a}^{s'} + 1$
10: $s = s'$
11: **end while**

Agrawal and Goyal [9] presented frequentist regret bounds for TS. The expected total regret is the amount we lose because of not taking an optimal action in each step. It is more convenient to work with the equivalent measure of regret. The expected total regret is defined as:

$$\mathbb{E}[\mathcal{R}(T)] = \sum_a \Delta_a \mathbb{E}[k_a(T + 1)] \tag{7}$$

where Δ_a is the lose measure between the Value of taking action and the optimal Value of taking the same action under the current beliefs on transition probabilities and $k_a(t)$ denotes the number of times the action a is played up to the max iteration.

2) Bayesian RL with Experience Replay: Since Thompson Sampling algorithm is an online planning solution to Bayesian RL, it could be treated as one trial of a game and the agent is the game player gaining experience as more games are played. This process can be modeled as Experience Replay [4]. The Experience Replay mechanism keeps a memory of past behaviors and randomly samples previous transitions, which smooths the learning over many past behaviors. Especially when the transition dynamics is time-varying, using Experience Replay would speed up the credit propagation and adapt to recent experiences.

Experience Replay has been applied to learn the weights of Q-Network [10] where the output of the neural network is the approximate Q-Value for each state-action pair. However, the Experience Replay proposed by them follows a ϵ-greedy strategy, meaning the agent is facing the exploration-exploitation dilemma and taking a random action with probability ϵ.

Algorithm 2 Bayesian RL with Experience Replay

Initialize replay memory $\mathcal{D}$ with capacity N
Initialize prior $\{\alpha_{a,s}\}$ for transition parameters
while max episode is not reached **do**
  $s = s_0$
  while max iteration is not reached **do**
    Sample $\theta^k_{a,s}$ from $\alpha_{a,s}, \forall a \in A, s \in S$
    for $i = 1 : k$ **do**
      $Q_i(s,a) = solveMDP(\theta^i)$
    end for
    $\overline{Q}(s,a) = \frac{1}{k}\sum_i Q_i(s,a)$
    $a^* = argmax_a \overline{Q}(s,a)$
    Execute a^* and receive data (s,a,s',r)
    Store (s,a,s',r) into $\mathcal{D}$
    Randomly select a minibatch (s,a,s',r) from $\mathcal{D}$
    $\alpha^{s'}_{s,a} = \alpha^{s'}_{s,a} + 1$
    $s = s'$
  end while
end while

Algorithm 2 proposed above uses Thompson Sampling to estimate the Q-Value instead of training the Q-Network. The focus of this algorithm is to eliminate the exploration-exploitation dilemma by updating the transition parameters and acting optimally under uncertain transition dynamics.

IV. Results

A. Chain Problem

Figure 1 shows the chain problem where the agent is selecting between two actions a, b and receiving rewards at the corresponding states. Action a takes the agent to the right with probability 0.8 and slips back to state 1 with probability 0.2. Action b takes the agent to the right with probability 0.2 and resets to state 1 with probability 0.8. The optimal policy is obviously to always choose action a, leading the agent to receive +10 reward until slipping back to the initial state. When the transition probabilities are unknown, the agent has to trade-off exploration and exploitation to act optimally while learning the transition dynamics.

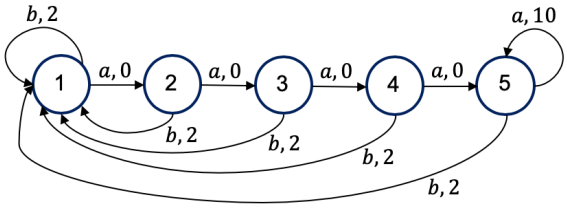


Fig. 1: The chain problem showing states, actions and rewards

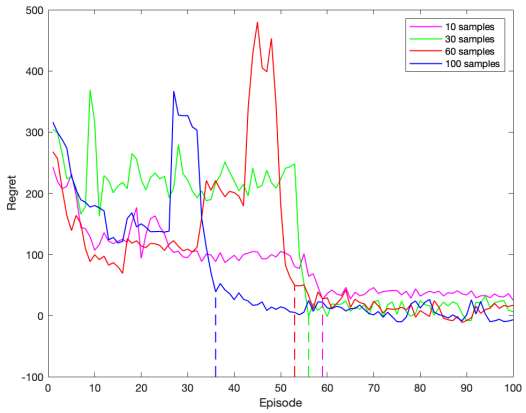


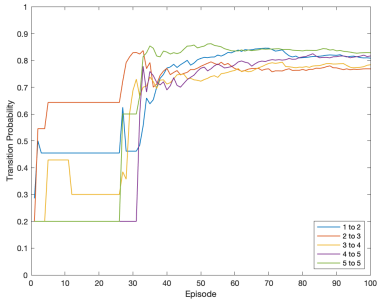
Fig. 2: Convergence of the expected total regret with different sample sizes in Chain problem. The dashed realization lines indicate the biggest drop of the regret.

Besides using the expected total regret as defined in Equation 7, we can also consider each episode as a trial of game in which the score is determined by the number of times the agent has been to the final state 5. The chain problem is hard to find the desired policy because taking action b is constantly gaining rewards and the agent has to risk to take action a. The agent would be too comfortable with taking action b.

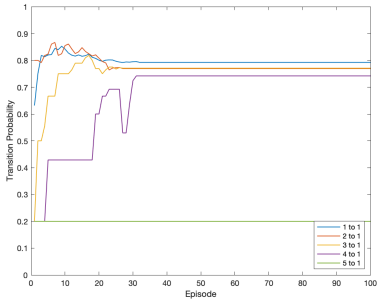
1) Game Performance: By implementing the proposed algorithm Bayesian RL with Experience Replay with 100 episodes, 50 iterations, and 10 memory capacity, the trends of the expected regret with samples sizes from 10 to 100 are shown in Figure 2. The corresponding dashed lines indicate the biggest drop in the regret, meaning the agent starts to realize the "true" dynamics of the environment and step out of its "comfort zone". We also compare the average score obtained before and and after such realization in Table I. As more samples are taken, the situation realization occurs at the earlier episode. Low sample size like 10 does not improve the score within 100 episodes. Also, the average score after the realization significantly improves.

n_s	Ep_r	$Score_b$	$Score_a$
10	59	0.1356	0.1463
30	56	3.9107	15.2500
60	53	2.0000	18.5319
100	36	3.0278	18.1250

TABLE I: n_s: sample size for Thompson Sampling; Ep_r: the realization episode; $Score_b, Score_b$: the average game score before and after realization.



(a) Transition probability for taking action a



(b) Transition probability for taking action b

Fig. 3: Posteriors on transition probability on certain state-action pairs for each episode with sample size 100. (a) Action a is expected to move the agent to the right with probability 0.8, (b) and action b is expected to reset the agent to the initial state with probability 0.8.

2) Transition Probability Inference: Figure 3 shows how the posterior of each transition probability evolves through the Experience Replay with sample size of 100. It is noticeable that around the realization episode (36), the agent switches the decision to take action a more frequently, which is the time the agent steps out of its comfort zone and receive higher accumulated rewards. Moreover, all of the transition probabilities converge to the true values except for the state 5 action b. This state-action pair is never updated because the agent is able to realize the best action at state 5 is always action a. This result is expected as Experience Replay because the agent would not waste time on exploring the action that does not bring credits.

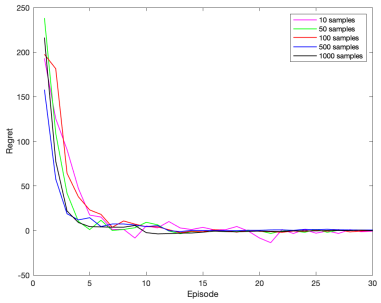
B. Fire Fighting Problem

Now we consider a simpler but more practical fire fighting problem. The state of the world is the fire level {None, Low, High}. The agent is commanding the the fire fighter to either stay at the same location to extinguish fire or move to another location to extinguish fire {Stay, Move}. This problem can be also formulated by Bayesian RL because the fire dynamics is uncertain and the fire fighter is facing the dilemma to either exploit to slowly put out the fire in front of him/her or explore in order to find the source of the fire and extinguish it completely. The actual transition probabilities are shown in Table II. Reward 10 will be received if the fire level is None and 0 otherwise.

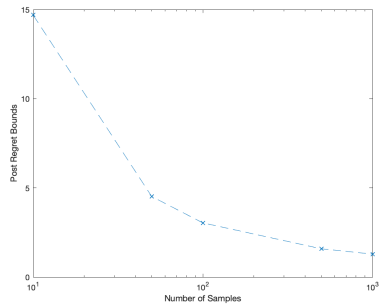
$T(s, Stay, s')$	None	Low	High
None	1	0	0
Low	0.6	0.4	0
High	0	0.6	0.4
$T(s, Move, s')$	None	Low	High
None	1	0	0
Low	0.9	0.1	0
High	0.5	0.4	0.1

TABLE II: True transition probabilities in fire fighting problem

The algorithm is run with 30 episodes, 10 max iteration, and 5 memory capacity. The convergence of the regret and the regret bounds for different samples sizes in the fire fighting problem are shown in Figure 4a and Figure 4b respectively. The convergence happens around



(a) Regret convergence



(b) Regret bounds

Fig. 4: (a) The regret convergence as more episodes trained; (b) The regret bounds for the fire fighting problem with different sample sizes.

the similar episode in this simple fire fighting case. However, the large regret bound with small samples size indicates the performance of the agent is not guaranteed, but is guaranteed within a bound. For more theoretical proof regarding the regret bounds, please refer to [9].

V. Conclusion

This project studies how to make decisions under unknown transition dynamics by balancing exploration and exploitation, so that there will never be a random action taken. Such decision making problem under uncertain transition probabilities can be modeled as a Bayesian Reinforcement Learning (BRL) framework. There, a formulation of BRL that converts the continuous belief states into the discrete space is presented by parameterizing the transition distributions. Also, an online learning algorithm - Experience Replay with Thompson Sampling for Bayesian RL is propsed to train an intelligent agent to adapt the system dynamics and make trade-offs between exploration and exploitation. We have shown that the algorithm could help the agent to step out of its "comfort zone" to explore more possibilities and eventually gain better accumulated rewards. Also, the belief monitoring on the transition probabilities has shown that the algorithm decides not to exploit when the expected reward is not better than exploration, which naturally solves the exploration-exploitation dilemma.

The proposed algorithm considers exploration strategies in Reinforcement Learning. However, estimate Q-Value using Thompson Sampling is computationally intensive in higher dimensions. The algorithm could be extended to representing the Q-Value with Q-Network where Q-Value in MDP could be replaces by a neural network and the network could be trained in the Experience Replay. In this case, the Q-Network would be trained more robustly to provide better performance especially facing exploration-exploitation dilemma.

References

[1] M. L. Puterman, *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York, NY: Wiley, 1994.

[2] J. P. M. Ghavamzadeh, S. Mannor and A. Tamar, "Bayesian reinforcement learning: A survey," *Foundations and Trends in Machine Learning*, vol. 8, p. 359–492, 2015.

[3] W. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, p. 285–294, 1933.

[4] L.-J. Lin, "Reinforcement learning for robots using neural networks," *School of Computer Science, Carnegie Mellon University*, 1993.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.

[6] A. R. C. L. P. Kaelbling, M. L. Littman, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, p. 99–134, 1998.

[7] J. H. P. Poupart, N. Vlassis and K. Regan, "An analytic solution to discrete bayesian reinforcement learning," *International Conference on Machine learning*, pp. 697–704, 2006.

[8] M. Duff, "Optimal learning: Computational procedures for bayes-adaptive markov decision processes," *University of Massachusetts Amherst, Amherst, MA*, 2002.

[9] S. Agrawal and N. Goyal, "Further optimal regret bounds for thompson sampling," *International Conference on Artificial Intelligence and Statistics*, pp. 99–107, 2013.

[10] D. S. A. G. I. A. D. W. V.Mnih, K. Kavukcuoglu and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *Clinical Orthopaedics and Related Research*, 2013.