Pure Exploration in Episodic Fixed-Horizon Markov Decision Processes

Sudeep Raja Putta
Xerox Research Centre India
Bangalore, India
Putta.Raja@xerox.com

Theja Tulabandhula
Xerox Research Centre India
Bangalore, India
theja.tulabandhula@xerox.com

ABSTRACT

We consider a Two Phase Exploration problem where an agent interacts with an unknown environment in two separate phases. The first is the exploration phase and the second is the evaluation phase. The agent is initially allowed to explore the environment for a limited number of interactions after which the agent's performance is evaluated. Since the agent is not evaluated on the rewards collected in the exploration phase, we propose using a Pure Exploration strategy in this phase and switching to an Explore-Exploit strategy in the evaluation phase.

We extend the fixed confidence Pure Exploration problem of Multi Armed Bandits to episodic fixed-horizon Markov Decision Processes (MDP). Here, the goal of an agent interacting with the MDP is to reach a high confidence in as few episodes as possible. We propose Posterior Sampling for Pure Exploration (PSPE), a Bayesian algorithm for pure exploration in MDPs. We empirically show that PSPE achieves deep exploration and the number of episodes required by PSPE for reaching a fixed confidence value is exponentially lower than random exploration and lower than regret minimizing algorithms such as Posterior Sampling for Reinforcement Learning (PSRL). For the two phase exploration problem, we propose using PSPE in the exploration phase and PSRL in the evaluation phase. We empirically show that PSPE achieves a good posterior and the regret incurred in the evaluation phase by using PSPE in the exploration phase is lower than the regret incurred by using random exploration or PSRL.

CCS Concepts

•**Computing methodologies → Sequential decision making;**

Keywords

Reinforcement Learning, Pure Exploration, Bayesian Algorithms

1. INTRODUCTION

Consider the situation where an agent interacts with an environment in two phases. The first phase is for training and fine tuning, where the agent is allowed to explore its environment for a limited amount of time without being evaluated on the rewards it receives. In the second phase, the goal of the agent is to maximise the sum of the rewards collected. Such situations are commonly seen in competitions, where there is a limited training phase before the actual competition begins. Both the training phase and the final competition will be on the same environment. The score of the agent in the training phase does not matter and it can use this phase to learn the best strategy to maximise its score in the actual competition. We call the initial training phase as the exploration phase and the final competition as the evaluation phase.

We consider environments which are modelled as episodic fixed-horizon Markov Decision Processes (MDP) with a finite number of states and actions. The agent interacts with the environment in episodes, each consisting of H time steps. In the "tabula rasa" setting, the agent has no prior knowledge about the MDP except the number of states S, actions A and the episode length H. In each phase, the agent has a different objective and faces a different kind of interactive learning problem.

In the evaluation phase, the agent's goal is to maximize the sum of rewards received while learning the MDP's parameters. This is the classical Reinforcement Learning (RL) [29] problem. This is an instance of an online learning problem, as the learning happens while interacting with the MDP and the goal is to optimize the online performance of the agent. Since the agent learns about the MDP by interacting with it, it faces the exploration vs. exploitation trade-off. At each step, the agent may choose to exploit its current experience by executing the action that currently seems best or explore a different actions which could result in gaining information that would lead to higher rewards in the future. The agent should balance exploration and exploitation so that it converges to an optimal policy and also receives near optimal rewards.

In the exploration phase, the agent's goal is to maximize the probability of following an optimal policy of the MDP. We call this probability as the confidence of the agent. This is the Pure Exploration (PE) problem. This is an instance of an active learning problem, as the agent has the ability to choose the sequence of policies to try, to maximize the confidence. Since the rewards received during this phase do not matter, the agent can execute actions solely based on the objective of reaching a high confidence as fast as possible.

We are interested in model based Bayesian algorithms. In these algorithms, the agent maintains a prior distribution on

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

the parameters of the MDP and computes posteriors based on the rewards and transitions observed. The algorithm uses these posteriors to pick actions according to the goal of the agent. The posterior distributions can also be used for sampling an instance of a MDP and for calculating the confidence of the agent.

Both the RL and PE problems have been studied in the case of stochastic Multi Armed Bandits (MAB). These are degenerate episodic fixed horizon MDPs, with a single state, A actions (also called arms of the bandit) and single step episodes. The RL problem for bandits is the classical problem of reward maximization while learning [16]. The Thompson Sampling (TS) [31] algorithm is a Bayesian algorithm for maximizing the cumulative reward while learning in bandits. The idea is to sample an instance of a bandit from the posterior at each step and pull its optimal arm. The PE problem for bandits is known as the best-arm identification problem. The TS algorithm is not suitable for PE as it pulls the estimated best arm almost all the time, and it takes a very long time to ascertain that none of the other arms offer better rewards. The Pure exploration Thompson Sampling(PTS)[25] algorithm modifies TS by adding a resampling step that prevents pulling the estimated best arm too often and helps it in achieving higher confidence in lesser number of arm pulls.

TS serves as a general sampling technique for Bayesian learning and can be easily extended to the complete RL problem on episodic fixed-horizon MDPs. The Posterior Sampling for Reinforcement Learning (PSRL) algorithm [18] maintains a prior distribution over MDPs. At the beginning of each episode, it samples a MDP instance from the current posterior and finds an optimal policy for the sampled instance using dynamic programming. It then acts according to this policy for the duration of the episode. It updates the posterior according to the rewards and transitions witnessed during the episode. Convergence to the optimal policy is guaranteed as samples are drawn from the full posterior and the mean of the posterior approaches the true MDP as the number of episodes increase. In the bandit case, this method is equivalent to TS.

In this paper we propose a model based Bayesian algorithm for the PE problem in stochastic episodic fixed-horizon MDPs called PSPE. PSPE modifies PSRL by adding a resampling step. We provide a simple procedure for finding the confidence using the posterior distributions. We empirically show that PSPE achieves deep exploration and reaches a high confidence faster than PSRL and exponentially faster than random exploration. We claim that the posterior distribution obtained after running PSPE for a fixed duration can be used by PSRL to obtain better rewards. For the two phase exploration problem, we empirically show that using PSPE in the exploration phase produces posteriors which can be used by PSRL in the evaluation phase to get higher rewards than using random exploration or PSRL in the exploration phase.

2. EPISODIC FIXED HORIZON MARKOV DECISION PROCESSES

An episodic fixed horizon Markov Decision Process M is given by the tuple $\langle \mathcal{S}, \mathcal{A}, R, P, H, \rho \rangle$. Here $\mathcal{S} = \{1, ..., S\}$ and $\mathcal{A} = \{1, ..., A\}$ are finite sets of states and actions respectively. The agent interacts with the MDP in episodes of length H steps. The initial state distribution is given by ρ. In each step $h = 1, ..., H$ of an episode, the agent observes a state $s_h \in \mathcal{S}$ and performs an action $a_h \in \mathcal{A}$. It receives a reward r_h sampled from the reward distribution $R(s_h, a_h)$ and transitions to a new state s_{h+1} sampled from the transition probability distribution $P(s_h, a_h)$. Let the average reward received for a particular state-action be $\bar{R}(s, a) = \mathbb{E}[r | r \sim R(s, a)]$.

For fixed horizon MDPs, a policy π is a mapping from $s \in \mathcal{S}$ and time step $h = 1, ..., H$ to action $a \in A$. In Appendix A, we give an example MDP to show that optimal actions may depend on both s and h. The number of deterministic policies for a MDP is A^{SH}, which is quite large even for small values of S,A and H. The value of a state s and action a under a policy π is defined as the expected sum of rewards attained starting at state s at step h, performing action a and acting according to π until the end of the episode.

$$Q_\pi(s, a, h) = \mathbb{E}\left[\bar{R}(s_h, a_h) + \sum_{i=h+1}^{H} \bar{R}(s_i, \pi(s_i, i))\right]$$

Let $V_\pi(s, h) = Q_\pi(s, \pi(s, h), h)$. A policy π^* is an optimal policy for the MDP if $\pi^* \in \arg\max_\pi V_\pi(s, h)$ for all $s \in \mathcal{S}$ and $h = 1, ..., H$. When the rewards and transition probabilities are known, an optimal policy for the MDP can be found using the Backward Induction[23] algorithm, described in Appendix B. a MDP may have multiple optimal policies. For a MDP M, let Π_M be the set of optimal policies.

The mean episodic reward of a policy π is given by $\mu(\pi)$.

$$\mu(\pi) = \sum_{s \in \mathcal{S}} \rho(s) V_\pi(s, 1)$$

Let $\mu^* = \max_\pi \mu(\pi)$. The gap of a policy is defined as $\Delta(\pi) = \mu^* - \mu(\pi)$.

2.1 Posterior Sampling for RL

PSRL is a natural extension of TS to episodic fixed-horizon MDPs. Consider a MDP with S states, A actions and horizon length H. PSRL maintains a prior distribution on the set of MDPs $\mathcal{M}$, i.e on the reward distribution R (on SA variables) and the transition probability distribution P (on S^2A variables). At the beginning of each episode t, a MDP M_t is sampled from the current posterior. Let P_t and R_t be the transition and reward distributions of M_t. The set of optimal policies Π_{M_t} for this MDP can be found using Dynamic Programming (see Appendix B) as P_t and R_t are known. The agent samples a policy π_t from Π_{M_t} and follows it for H steps. The rewards and transitions witnessed during this episode are used to update the posteriors. Let f be the prior density over the MDPs and $\mathcal{H}_t$ be the history of episodes seen until $t-1$. Let $s_{h,t}$ be the state observed, $a_{h,t}$ be the action performed and $r_{h,t}$ be the reward received at time h in episode t. Algorithm 1 describes PSRL.

Like TS, PSRL maintains a prior distribution over the model, in this case a MDP. At each episode, it samples a model from the posterior and acts greedily according to the sample. TS selects arms according to their posterior probability of being optimal and PSRL selects policies according to the posterior probability they are optimal. It is possible to compute the posterior efficiently and sample from it by a

Algorithm 1 PSRL

```
$\mathcal{H}_1 = \{\}$
for $t = 1, 2, ...$ do
    Sample $M_t \sim f(\cdot|\mathcal{H}_t)$
    Choose a policy $\pi_t$ at random from $\Pi_{M_t}$
    Observe initial state $s_{1,t}$
    for $h = 1, ..., H$ do
        Perform action $a_{h,t} = \pi_t(s_{h,t}, h)$
        Observe reward $r_{h,t}$ and next state $s_{h+1,t}$
    end for
    $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{(s_{h,t}, a_{h,t}, r_{h,t}, s_{h+1,t})|h = 1..H\}$
end for
```

proper choice of conjugate prior distributions or by the use of Markov Chain Monte Carlo methods.

3. POSTERIOR SAMPLING FOR PE

PSRL is not suitable for pure exploration since it is designed for maximizing cumulative rewards and hence follows an optimal policy very often. PSPE modifies PSRL by adding a re-sampling step. This is an extension of the Top-Two sampling idea of PTS to PSRL. This prevents it from following an estimated optimal policy too frequently. The algorithm depends on a parameter β, where $0 \leq \beta \leq 1$, which controls how often an optimal policy of a sampled MDP is followed. At each episode t, PSPE samples a MDP M_t and finds its set of optimal policies Π_{M_t}. With probability β it follows a policy from this set. With probability $1-\beta$ it re-samples MDPs until a different set of policies $\Pi_{\widetilde{M}_t}$ is obtained. It then follows a policy from the set $\Pi_{\widetilde{M}_t} - \Pi_{M_t}$ for H steps. Algorithm 2 describes PSPE. In the case of bandits, PSPE is equivalent to PTS.

Algorithm 2 PSPE

```
$\mathcal{H}_1 = \{\}, t = 1$
for $t = 1, 2, ...$ do
    Sample $M_t \sim f(\cdot|\mathcal{H}_t)$
    Sample $B \sim Bernoulli(\beta)$
    if $B = 1$ then
        Choose a policy $\pi_t$ at random from $\Pi_{M_t}$
    else
        repeat
            Re-sample $\widetilde{M}_t \sim f(\cdot|\mathcal{H}_t)$
        until $\Pi_{\widetilde{M}_t} - \Pi_{M_t} \neq \emptyset$
        Choose a policy $\pi_t$ at random from $\Pi_{\widetilde{M}_t} - \Pi_{M_t}$
    end if
    Observe initial state $s_{1,t}$
    for $h = 1, ..., H$ do
        Perform action $a_{h,t} = \pi_t(s_{h,t}, h)$
        Observe reward $r_{h,t}$ and next state $s_{h+1,t}$
    end for
    $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{(s_{h,t}, a_{h,t}, r_{h,t}, s_{h+1,t})|h = 1..H\}$
end for
```

3.1 Computing the Confidence

Let M^* be the true underlying MDP and let Π^* be its set of optimal policies. The confidence of the agent α_t at episode t is the probability of sampling a MDP M_t and following one of its optimal policies π_t such that $\pi_t \in \Pi^*$.

We define the confidence of a set of policies Π as the probability of sampling a MDP M and following a policy from Π_M such that it is also in Π. Let $x_\Pi(M)$ denote the probability of picking a policy from Π_M which is also in Π.

$$x_\Pi(M) = \frac{|\Pi_M \cap \Pi|}{|\Pi_M|}$$

The confidence of Π, denoted by α_Π can be expressed as the expectation of $x_\Pi(M)$ computed over the current posterior distribution of MDPs.

$$\alpha_\Pi = \mathbb{E}_M[x_\Pi(M)] = \int\limits_{M \in \mathcal{M}} x_\Pi(M) f(M|\mathcal{H}) dM$$

Due to the Law of Large Numbers, this expectation is the same as this summation in the limit.

$$\alpha_\Pi = \lim_{n \to \infty} \frac{\sum_{j=1}^{n} x_\Pi(M_j)}{n}$$

Where M_j are drawn independently from $f(\cdot|\mathcal{H})$ for all $j = 1, ..., n$. This gives an easy way of approximating α_Π, by drawing a large number of sample MDPs from the posterior and finding the average value of $x_\Pi(M)$ for these samples.

At episode t, the confidence of the agent is the same as the confidence over the set of policies Π^*, i.e, $\alpha_t = \alpha_{\Pi^*}$. We monitor the value of α_t and count the number of episodes required to reach a certain high confidence value to evaluate the performance of our algorithm. Note that our algorithm itself does not require the confidence value for its operation.

3.2 Empirical Evaluation

We compare the performance of PSPE with PSRL and random exploration. We measure the number of episodes required by each of these algorithms to reach a high confidence value. To ease the procedure of computing posterior distributions and sampling MDPs from the posterior, we use suitable conjugate-prior distributions. For the transition probabilities, we assume a uniform Dirichlet prior and a categorical likelihood, and for reward distribution, we assume a Gaussian prior ($\mathcal{N}(0, 1)$) and a Gaussian likelihood with unit variance for each state-action pair. We calculate α_{Π^*} by sampling 10000 independent MDPs from the posterior. The experiment tracks the first time when the confidence value exceeds a fixed confidence of 0.90, 0.95 and 0.99. All the results are averaged across 50 trials. We choose $\beta = 1/2$ in PSPE.

We generate a random episodic fixed-horizon MDP having 3 states, 3 actions and horizon length 3. The total number of deterministic policies for this MDP are $3^{3\times3}$. The transitions are stochastic and Gaussian noise is added to the rewards produced by this MDP. Since it is computationally intensive to calculate the confidence after each episode, we only calculate it after every 10,000 episodes. Figure 1 displays the average number of episodes required by each algorithm to reach each fixed confidence. PSPE reaches a high confidence in lesser number of episodes than both PSRL and random exploration.

Stochastic Chains (Figure 2) proposed by Osband & Van Roy [22, 21], is a family of MDPs which consist of a long chain of N states. At each step, the agent can choose to go left or right. The left actions (indicated by thick lines)

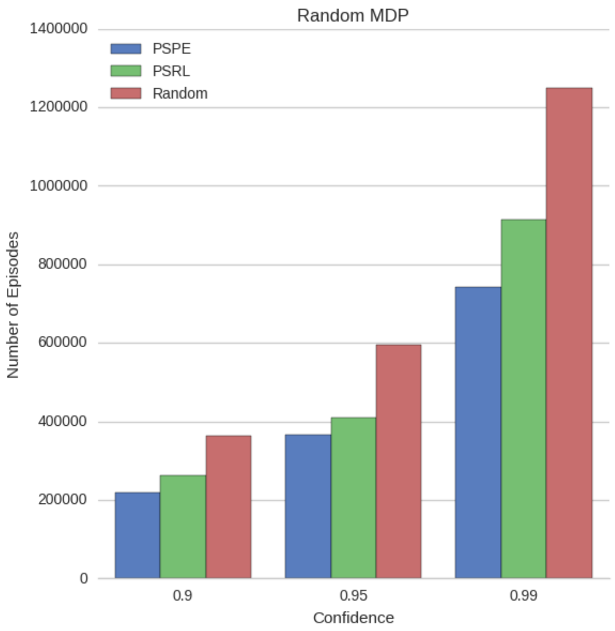

Figure 1: Number of Episodes required to reach a given confidence in Random MDP

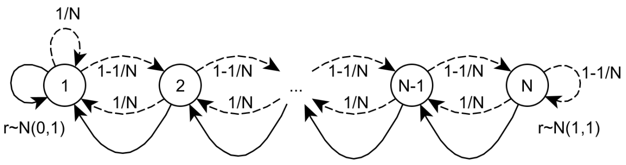

Figure 2: Stochastic Chain MDP

are deterministic, but the right actions (indicated by dotted lines) result in going right with probability $1 - 1/N$ or going left with probability $1/N$. The only two rewards in this MDP are obtained by choosing left in state 1 and choosing right in state N. These rewards are drawn from a normal distribution with unit variance. Each episode is of length $H = N$. The agent begins each episode at state 1. The optimal policy is to go right at every step to receive an expected reward of $(1 - \frac{1}{N})^{N-1}$. For the RL problem on these MDPs, dithering strategies like ϵ-greedy or Boltzmann exploration are highly inefficient and could lead to regret that grows exponentially in chain length.

We consider a stochastic chain of length 10. The total number of deterministic policies for this MDP are $2^{10\times 10}$. We calculate the confidence after every 100 episodes. Figure 3 displays the average number of episodes required by each algorithm to reach each fixed confidence of 0.90, 0.95 and 0.99. Both PSRL and PSPE reach the desired confidence fairly quickly, but random exploration requires a very large number of episodes.

On this family of MDPs, the number of episodes required to reach a fixed confidence grows exponentially when using random exploration. In the case of PSRL and PSPE, this number grows linearly. We consider stochastic chains of length 2 to 10. We measure the number of episodes required to reach a confidence of 0.95 for each of these MDPs using random exploration, PSRL and PSPE. Figure 4 displays the results. The number of episodes required by PSPE and PSRL is practically the same and grows very slowly for this family of MDPs. Random exploration however, is highly inefficient and the number of episodes it requires grows exponentially. This is because both PSRL and PSPE are able to achieve "Deep Exploration" [17, 19] whereas random exploration does not. Deep Exploration means that the algorithm selects actions which are oriented towards positioning the agent to gain useful information further down in the episode.

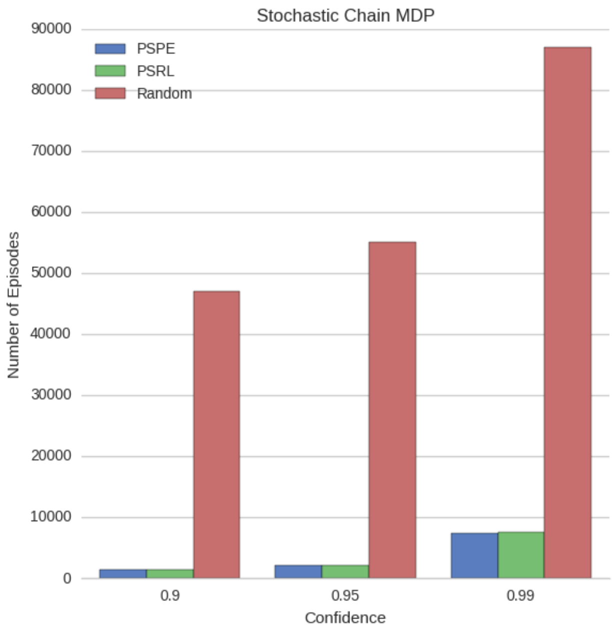

Figure 3: Number of Episodes required to reach a given confidence in Stochastic Chain of length 10

3.3 Discussion

PSPE reaches a high confidence level in lesser number of episodes than both PSRL and random exploration. The random exploration strategy would choose to follow a policy uniformly at random. It takes the most number of episodes as it treats each policy equally, without considering the rewards received. It does not leverage the fact that some policies can be quickly ruled out as they are clearly suboptimal. On the other hand, PSRL often follows the policy with the highest confidence and does not spend much effort in refining its knowledge of other policies. The re-sampling step in PSPE ensures that the algorithm adaptively chooses policies such that nearly optimal policies are chosen more often than policies which are clearly suboptimal.

The random exploration policy is highly inefficient as the number of episodes it requires to reach a fixed confidence level grows exponentially with the size of the MDP. PSPE is able to achieve deep exploration as demonstrated on the stochastic chain MDP. The number of episodes required by PSPE is significantly less as it is able to direct its effort towards gathering information about rewards and transitions further down the chain.

The confidence of the agent $\alpha_t \to 1$ as $t \to \infty$. Using

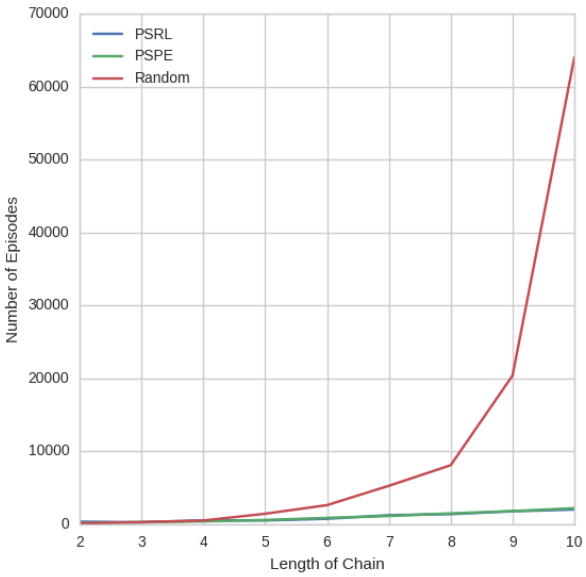

Figure 4: Number of episodes required to reach a confidence of 0.95 for various chain lengths

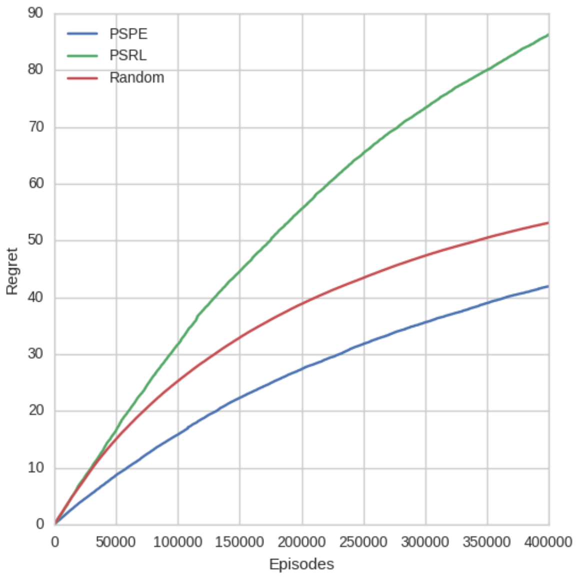

Figure 5: Evaluation phase regret in Random MDP

Theorem 1 from Russo [25], we claim that $(1-\alpha_t) \rightarrow 0$ at a rate $\exp\{-t\Gamma^*_{1/2}\}$ under PSPE with $\beta = 1/2$. Here

$$\Gamma^*_{1/2} = O((\sum_{\pi \notin \Pi^*} \Delta(\pi)^{-2})^{-1})$$

The rate of convergence for random exploration is of the order of $O(\min_{\pi \notin \Pi^*} \Delta(\pi)^2)/A^{SH})$. The convergence of PSPE is faster as it depends on the individual policy gaps whereas for random exploration, it depends only on the smallest gap.

4. TWO PHASE EXPLORATION

Consider a competition which involves an autonomous agent exploring an unknown environment. Typically in such competitions, there are two phases of interaction. In the first phase, the agent is allowed to interact with the environment without being evaluated on the rewards it receives. This is the exploration phase. In the second phase, it interacts with the same environment, but the rewards received are used to evaluate the agent's performance. This is the evaluation phase. The purpose of the first phase is for the agent to fine tune its performance on the environment so that it gets the best possible rewards in the second phase. The first phase has a limited number of interactions after which the second phase begins.

It is not entirely apparent which strategy the agent should use in the first phase. The agent could ignore the fact that the rewards accumulated in the first phase do not matter and use a reward maximizing strategy in both phases. It could explore randomly during the first phase, and then switch to a reward maximizing strategy in the second phase. It could just as well use a pure exploration strategy and then a reward maximizing strategy in the second phase. The strategy to use could also depend on the budget of the first phase.

Consider the case where the environment is an episodic fixed horizon MDP. Ideally, having good posterior distributions after the exploration phase should result in lower regret in the evaluation phase. If the posterior distributions are concentrated around the true parameter values of the MDP, then PSRL will follow an optimal policy with high probability resulting in lower regret. As we only have a limited number of episodes in the exploration phase, the agent should try to obtain good posteriors in as few episodes as possible. Since PSPE reaches a high confidence value faster than PSRL and random exploration, an optimal policy of a sampled MDP will be an optimal policy of the true MDP with high probability. Hence, we argue that lower regret is incurred by PSRL in the evaluation when PSPE is used in the exploration phase.

4.1 Empirical Evaluation

We consider three different strategies as discussed above. These either use PSRL, PSPE or random exploration in the exploration phase and switch to PSRL in the evaluation phase. All the results are averaged across 50 trials. We consider the Random MDP having 3 states, 3 actions and horizon length 3. The exploration phase lasts for the first 400,000 episodes and then the regret of the agents is monitored for the next 400,000 episodes. Figure 5 displays the regret incurred. PSPE incurs the least amount of regret in the exploration phase. PSRL on the other hand incurs the most regret.

Next we consider the stochastic chain of length 10. The exploration phase lasts for the first 10,000 episodes and then the regret of the agents is monitored for the next 10,000 episodes (Figure 6). In this MDP, random exploration incurs the most amount of regret. This is because random exploration does not gain good posteriors on the rewards and transitions further down the chain as it is unable to achieve deep exploration. PSPE and PSRL however incur very low regret.

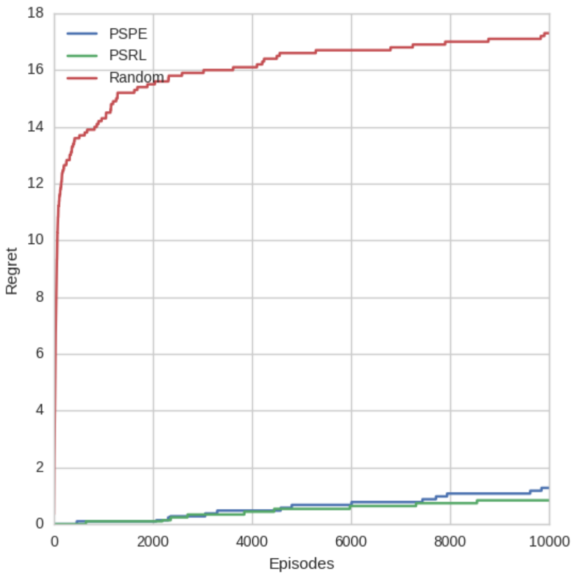

Figure 6: Evaluation phase regret Stochastic Chain

We also investigate the dependence of the regret in the evaluation phase on the budget of the exploration phase when using PSPE, PSRL or random exploration. On the Random MDP, we let the budget of the exploration phase vary from 1000 to 100,000 in steps of 1000. After the first phase, the evaluation phase lasts for 10,000 episodes. The results are plotted in Figure 7. Random exploration incurs lower regret when the exploration phase budget is less than 50,000 episodes. After 50,000 episodes however, PSPE incurs lower regret. This result agrees with Figure 5 when we set the exploration budget as 400,000 episodes.

On the stochastic chain MDP of length 10, we let the budget of the exploration phase vary from 100 to 10,000 in steps of 100. After the first phase, the evaluation phase lasts for 10,000 episodes. The results are plotted in Figure 8 Random exploration incurs the highest regret of all the three. This is because it incurs a very high regret during the initial episodes of the evaluation phase (Figure 6), as it does not achieve deep exploration as seen in Figure 4. Both PSPE and PSRL achieve very low regrets.

4.2 Discussion

For the two phase exploration problem, using PSPE instead of random exploration in the exploration phase seems to give lower regret in the evaluation phase. In the case of the Random MDP, random exploration performs well when the exploration phase's budget is low. We argue that this is because the MDP was generated randomly, it will not be sparse, i.e., there will be some non-zero probability of transitioning from one state to any other state and every action will result in some non-zero reward. The difference in performance can be clearly seen when we consider stochastic chain MDPs, where random exploration does not perform well at all. On the stochastic chain, PSPE and PSRL incur almost the same amount of regret in the evaluation phase. On the Random MDP however, PSPE has lesser regret than

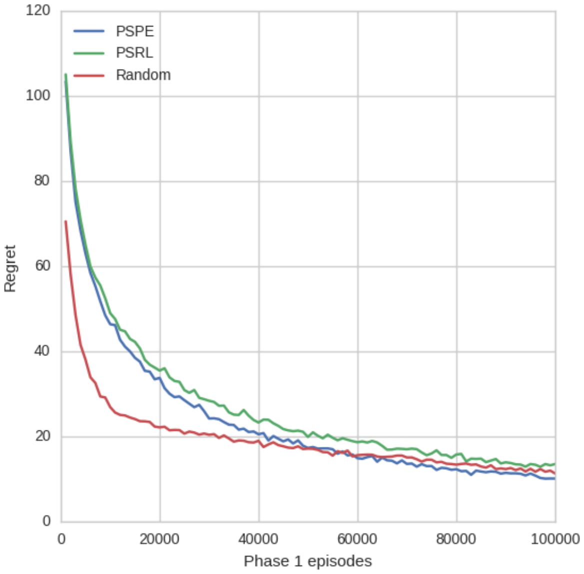

Figure 7: Evaluation phase regret vs Exploration phase budget in Random MDP

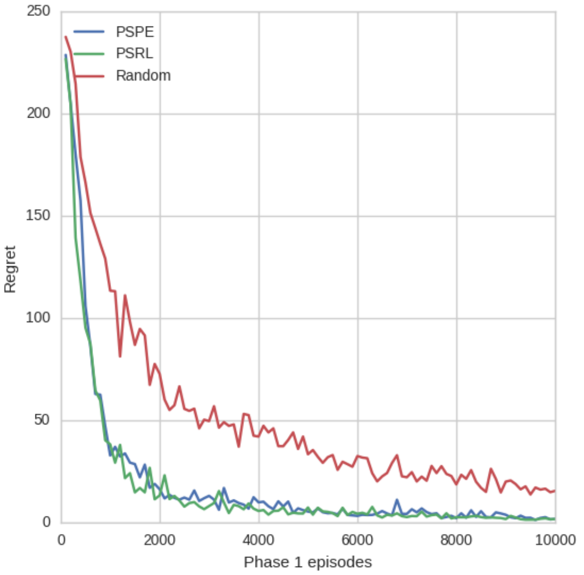

Figure 8: Evaluation phase regret vs Exploration phase budget in Stochastic Chain

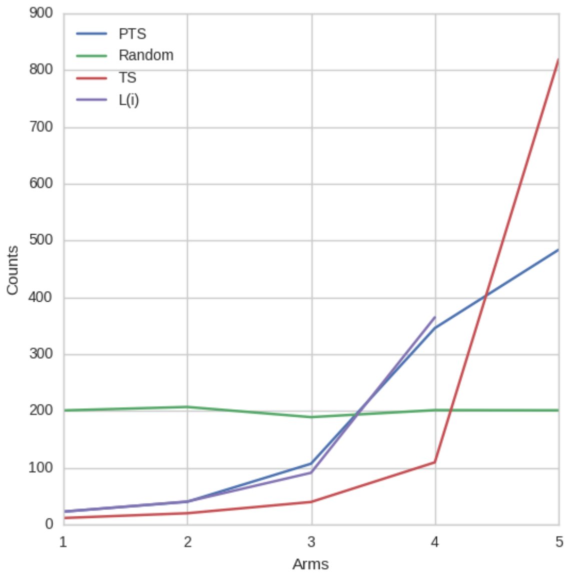

Figure 9: Number of times each arm is pulled

PSRL. By using PSPE in the exploration phase, we are able to achieve a higher confidence as well as a better posterior than PSRL or random exploration. This posterior can be used by PSRL in the evaluation phase to get better rewards.

In the analysis of TS by Agarwal & Goyal [1], they state that after T steps, arm i's posterior distribution is tightly concentrated around its true mean reward with high probability if it has been pulled a sufficient number ($O((\ln T)/\Delta_i^2)$, where Δ_i is its arm gap) of times. This implies that arms with a low arm gap (high arm mean) must be pulled more often than arms with a high arm gap (low arm mean). Consider a 5 armed Bernoulli bandit with arm means (0.1, 0.2, 0.3, 0.4, 0.5). We run TS, PTS and random exploration for $T = 1000$ steps and plot the number of times each arm has been pulled in Figure 9. We compare the plot with $L(i) = C(\ln T)/\Delta_i^2$ for the first 4 arms with a suitable value of C.

Random exploration is equivalent to pulling arms uniformly, as each arm has the same probability of being pulled. This causes random exploration to pull clearly suboptimal arms more often than necessary. TS pulls the optimal arm too often, which prevents the posteriors of other arms from tightly concentrating around their respective arm means.The number of times PTS pulls the arm i is very close to to the value of $L(i)$ for all the suboptimal arms. The posterior distribution of the arms will be tightly concentrated around their respective arm means. PTS is able to adaptively decide which arm to pull so that it achieves the best possible posterior within the given budget. Thus in the two phase exploration problem, the posterior obtained by using PTS in the exploration phase can be used by TS in the evaluation phase to achieve very low regret. Since PSPE uses the same top two sampling procedure of PTS, we argue that PSPE also achieves the best possible posterior within the exploration budget.

5. RELATED LITERATURE

Both PSPE and PSRL are based on algorithms which were originally developed for Multi Armed Bandits. MAB problems have been widely studied in a variety of areas since as early as the 1930s. One of the earliest works on this problem was by Thompson [31] who was studying bandit problems in the context of clinical trials. Robbins [24] derived strategies that asymptotically attain an average reward that converges in the limit to the reward of the best arm. Lai and Robbins [16] provide asymptotic lower bounds on the expected regret of any algorithm for the stochastic MAB problem. The Upper Confidence Bound family of algorithms for bandits, which are based on the principle of Optimism in the Face of Uncertainty (OFU) have been shown to have good theoretical guarantees. The Upper Confidence Bound (UCB) [2] algorithm is a popular approach based on the OFU principle. Thompson Sampling (TS) [31] is a natural Bayesian algorithm for the MAB problem that uses randomized probability matching. Chappel and Li [6] demonstrate that empirically TS achieves regret comparable to the lower bounds of Lai and Robbins[16]. A frequentist regret analysis of TS was first given by Agarwal & Goyal [1] for Bernoulli MABs with uniform prior. Russo & Van Roy [26] provide Bayesian regret bounds for TS using information theoretic tools. Refer to [10] and [5] for a comprehensive discussion of these algorithms.

Pure exploration in MABs is the problem of identifying the best arm within a fixed budget of arm pulls or up to a fixed confidence in as few arm pulls as possible. Bubeck et al.[5] provide a brief survey of this problem. Jamieson & Nowak [12] review many of the algorithms proposed for this problem in the fixed confidence setting. Russo [25] proposed PTS along with two other Bayesian algorithms for this problem along with a frequentist analysis of their performance. In the case of MDPs, PTS generalized to PSPE.

Several efficient algorithms with theoretical guarantees have been proposed for the Reinforcement Learning problem. The UCRL2 [11] algorithm is based on the OFU principle and achieves logarithmic regret. PSRL was first proposed by Strens[28] under the name of Bayesian dynamic programming and a Bayesian regret bound was provided by Osband et al.[18]. Algorithms which are PAC-MDP [13] have a high probability bound on the number of times the algorithm acts sub-optimally. Algorithms which are PAC-MDP include R-Max [4], E^3 [14], Delayed Q-Learning[27], BEB[15]. Refer to [10] and [30] for a comprehensive discussion of these algorithms. Dann & Burnskill [7] propose UCFH, which is an algorithm suitable for episodic fixed horizon MDPs and is based on the OFU principle. UCFH has a PAC guarantee for the number of episodes required to be close to the optimal episodic reward.

There exists only a few theoretical works on the Pure Exploration problem in MDPs. Thrun [32] considered the problem of active learning in deterministic environments and provided bounds on the number of actions required for finding an optimal policy. This bound can be significantly improved as shown by Szepesvari [30]. Evan-Dar et al. [9] consider the problem in finite stochastic MDPs under the assumption that the agent can reset the state of the MDP to an arbitrary state. Our algorithm PSPE is suitable for stochastic MDPs and does not require resets to arbitrary states as the MDP is episodic.

6. EXTENSIONS AND FUTURE WORK

Osband & Van Roy [20] suggest a few approaches to adapt PSRL for infinite horizon discounted MDPs. These methods could also be used for PSPE. A simple way is to impose an artificial episode length $H = O((1-\gamma)^{-1})$ when γ is the discount factor. Algorithm such as UCRL [11] and REGAL[3] start a new episode when the total number of visits to any state and action has doubled. We can also apply the same technique for PSRL and PSPE.

PSRL and PSPE require solving MDPs through dynamic programming at each step. An alternative approach which avoids solving sampled MDPs could be to use value function sampling [8]. Osband et al. propose RLSVI [22] which samples from the distribution over value functions and is designed for efficient exploration in large MDPs and generalizes through linearly parameterized value functions. Bootstrap DQN [17], which uses Bootstrap for posterior sampling and a deep neural network for representing value functions has been shown to generalize efficiently and achieve deep exploration in environments with extremely large state spaces. These approaches are similar to PSRL as they act greedily according to the sampled instance and are designed to maximise the cumulative reward. However, extending the value function sampling approach to the idea of Top-Two sampling is not straight forward. Using value function sampling approaches to achieve pure exploration remains an open research direction.

We only provide a loose convergence rate for the confidence of PSPE. This bound can be improved with a thorough analysis. We empirically show that PTS adaptively pulls arms in order to obtain a good posterior distribution. Analysis of the number of times each arm is pulled by PTS could offer further insights.

7. CONCLUSION

In this paper, we pose the Two Phase Exploration problem which consists of separate exploration and evaluation phases. We present Posterior Sampling for Pure Explorations as a Bayesian algorithm for the problem of Pure exploration under a fixed confidence setting in episodic fixed-horizon MDPs. We demonstrate that PSPE can achieve a high confidence in lesser number of episodes than PSRL or random exploration. We also show that PSPE is able to achieve deep exploration like PSRL. For the two phase exploration problem, we show that by using PSPE in the exploration phase, we get higher rewards in the evaluation phase. In the bandit setting, we show that PTS achieves the best possible posteriors in a limited budget. Since PTS is a special case of PSPE, we claim that using PSPE in the exploration phase gives posterior distributions which enable PSRL to obtain higher rewards in the evaluation phase.

APPENDIX

A. NON STATIONARY POLICIES

Unlike infinite horizon MDPs the policies we consider here are non-stationary, i.e. the action depends on the state s and current step in the episode h. We present an example to show that non stationary policies are necessary for fixed horizon episodic MDPs. Consider the MDP in Figure 10. It has 4 states, two actions and horizon length 3 starting from s_0. Choosing to go right from s_0 will end in going to s_2 or remaining in s_0 with equal probability. The only rewards in the system have value 1 and 5 which are obtained by going left at s_1 and right at s_3 respectively.

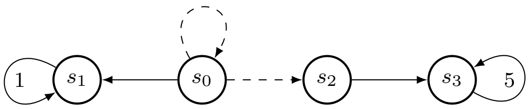

Figure 10: Episodic fixed-horizon MDP

In this MDP, the best stationary policy is to always go right. This policy's expected total reward per episode is 2.5. The best non-stationary policy however is to go right from s_0 at $h = 1$. If it succeeds to go to s_2, then it can get a reward of 5 by continuing right until the episode ends. However, if it remains in s_0, it can get a reward of 1 by going left until the episode ends. Hence the expected total reward for the best non-stationary policy is 3.

B. FIXED HORIZON DYNAMIC PROGRAMMING

Here we describe the Backward Induction algorithm for computing the optimal policy of a fixed-horizon MDP when its average rewards and transition probabilities are known. The algorithm iterates backwards from $h = H$ to $h = 1$. At $h = H$, the Q values for each state-action are equal to their immediate rewards as it is the end of the episode. At each step, it computes the Q values by applying the Bellman Operator for each state and action. The time complexity of this algorithm is $O(S^2AH)$ as applying the Bellman Operator requires $O(S)$ steps and we do this SAH times.

Algorithm 3 Backward Induction

```
$Q(s,a,H) = \bar{R}(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$
$V(s,H) = \max_a\{Q(s,a,H)\}$ for all $s \in \mathcal{S}$
for $h = H-1,..,1$ do
    for $s \in \mathcal{S}$ do
        for $a \in \mathcal{A}$ do
            $Q(s,a,h) = \bar{R}(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s')V(s',h+1)$
        end for
        $V(s,h) = \max_a\{Q(s,a,h)\}$
        $\pi(s,h) = \operatorname{argmax}_a\{Q(s,a,h)\}$
    end for
end for
return $\pi$
```

REFERENCES

[1] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–47, 2012.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[3] P. L. Bartlett and A. Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In

Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 35–42. AUAI Press, 2009.

[4] R. I. Brafman and M. Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

[5] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[6] O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

[7] C. Dann and E. Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

[8] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-learning. In *AAAI Conference on Artificial Intelligence*, pages 761–768, 1998.

[9] E. Even-Dar, S. Mannor, and Y. Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[10] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. *Bayesian reinforcement learning: a survey*. World Scientific, 2015.

[11] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

[12] K. Jamieson and R. Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Forty eighth Annual Conference on Information Sciences and Systems*, pages 1–6. IEEE, 2014.

[13] S. M. Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London, 2003.

[14] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[15] J. Z. Kolter and A. Y. Ng. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 513–520. ACM, 2009.

[16] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[17] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. *arXiv preprint arXiv:1602.04621*, 2016.

[18] I. Osband, D. Russo, and B. Van Roy. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

[19] I. Osband and B. Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

[20] I. Osband and B. Van Roy. Posterior sampling for reinforcement learning without episodes. *arXiv preprint arXiv:1608.02731*, 2016.

[21] I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning. *arXiv preprint arXiv:1607.00215*, 2016.

[22] I. Osband, B. Van Roy, and Z. Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

[23] M. L. Puterman. *Markov Decision Processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[24] H. Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[25] D. Russo. Simple bayesian algorithms for best arm identification. *Twenty ninth Annual Conference on Learning Theory*, pages 1417–1418, 2016.

[26] D. Russo and B. Van Roy. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 2014.

[27] A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *Twenty third International Conference on Machine Learning*, pages 881–888. ACM, 2006.

[28] M. Strens. A Bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, pages 943–950, 2000.

[29] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*, volume 1. MIT Press Cambridge, 1998.

[30] C. Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.

[31] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[32] S. B. Thrun. Effcient exploration in reinforcement learning. Technical report, CMU-CS-92-102, School of Computer Science, Carnegie Mellon University, 1992.