EMPIRICAL STUDY FOR THOMPSON SAMPLING IN REINFORCEMENT LEARNING

Baizhi (Daniel) Song
Department of IEOR
Columbia University
New York, NY 10027
bs3228@columbia.edu

September 2, 2020

ABSTRACT

We conduct an empirical regret analysis under simulation setting to compare the performance of different model-based reinforcement learning algorithms. Under all test scenarios, the Thompson Sampling type algorithm PSRL outperforms the benchmark algorithms UCRL2, with significantly lower total regret and a faster converge speed. The empirical results also show the necessity of multi-sampling within the PSRL. When the state space is small, the multi-sampling method has slight improvement on the PSRL's performance. As the number of states increases, such method could efficiently avoid the non-convergence problem and accordingly improve the PSRL algorithm's performance.

1 Introduction

In this project we consider the regret analysis of Reinforcement Learning (RL) problem where the agent sequentially interacts with an unknown environment and repeatedly updates its policy to minimized the difference between its total rewards and the optimal policy's rewards. The model-based reinforcement learning algorithm model the unknown environment as a Markov decision process (MDP) with unknown parameters and accordingly divides the problem into two parts: exploration of the MDP's parameters and optimal planning based on the estimated MDP. The second part has been well discussed in the field of dynamic programming and control problem where the MDP is given and fixed. As for the first part, the main difficulty for an RL algorithm to perform well is handling the inherent trade-off between exploration and exploitation: the agent could temporally get more rewards by executing the optimal strategy on its current information, but might miss the larger rewards states in the long run if it does not collect enough information on those less known states.

Current research on the exploration mainly focus on two types of method. The classic Upper Confidence Bound (UCB) algorithm is based on the frequentist view where agents estimated the confidence bound of the environment parameters and take the upper bound as the estimation to consider both mean and variance of the estimators. Recently there are increasingly more research on the Posterior Sampling (PS, or Thompson Sampling) algorithm. The PS type algorithm is based on the Bayesian view and update the posterior distribution of the environment parameters. Such algorithm samples directly from the posterior distribution as the estimation where the distribution itself includes the variance of the current knowledge. In this project we compare two versions of our PS type algorithm PSRL with the classic UCB type benchmark algorithm UCRL2. We conduct multiple numerical experiments under simulation setting and analyze the empirical performance of those algorithm under various scenarios.

2 Algorithm

This project compares the Thompson Sampling type RL with the benchmark UCB type algorithm for the regret analysis. As both described as model-RL algorithm, those two types of algorithms share many similar features. In both case the algorithms collect the historical transition information: $N_k(s,a) := \#\{\tau < t_k : s_\tau = s, a_\tau = a\}$, $N_k(s,a,s') := \#\{\tau < t_k : s_\tau = s, a_\tau = a, s_{\tau+1} = s'\}$ and $R_k(s,a) := \sum_{\tau=1}^{t_k-1} r_\tau \mathbb{1}_{s_\tau=s, a_\tau=a}$. Meanwhile, both of the algorithms evaluate the policy on the epoch base and update the explorable policy at the beginning of each epoch k.

2.1 UCRL2 Algorithm

Described in (Jaksch et al., 2010), the UCRL2 estimate the upper confidence bound for both reward function $\hat{r}_k(s,a) := \frac{R_k(s,a)}{max\{1,N_k(s,a)\}}$ and transition matrix $\hat{p}_k(s'|s,a) := \frac{N_k(s,a,s')}{max\{1,N_k(s,a)\}}$. With the confidence set of MDP $\mathcal{M}_k$, the algorithm solve the optimal policy through an extended version of value iteration where the maximum transition probability is calculated directly by the convexity. The algorithm in summarized as following Algorithm 1:

$$|\tilde{r}(s,a) - \hat{r}(s,a)| \leq \sqrt{\frac{7log(2SAt_k/\delta)}{2max\{1, N_k(s,a)\}}} \tag{1}$$

$$|\tilde{p}(\cdot|s,a) - \hat{p}(\cdot|s,a)| \leq \sqrt{\frac{14log(2SAt_k/\delta)}{2max\{1, N_k(s,a)\}}} \tag{2}$$

Initialization: t:=1, and reset $N_k(s,a)$, $N_k(s,a,s')$, $R_k(s,a)$;
for *epoch k = 1, 2, ...* **do**
    Set $t_k := t$, $v_k(s,a) = 0$;
    Calculate upper confidence bound for $\hat{r}_k(s,a)$ and $\hat{p}_k(s'|s,a)$ via equation (1) and (2);
    Compute optimal policy π^k through extended value iteration;
    while $v_k(s,a) < max\{1, N_k(s_t, \pi^k(s_t)\}$ **do**
        Execute π^k and update $N_k(s,a)$, $N_k(s,a,s')$, $R_k(s,a)$;
        $a_t = \pi^t(s_t)$;
        Observe next state $s_{t+1} = env(a_t)$;
        $t = t + 1$;
    end
end

Algorithm 1: UCRL2 Algorithm

2.2 PSRL Algorithm

Described in (Agrawal et al., 2017), the PSRL updates the posterior distribution of the transition probability and the reward function. Given the history information, the posterior distribution is updated with:

$$Q_k^j(s,a) \sim Dirichlet(\frac{1}{\kappa}(N_k(s,a,s') + \omega)) \tag{3}$$

Where $Q_k^j(s,a)$ is the j^{th} sampled probability from total ψ pairs of sampled parameters. In this project we discussed two version of the PSRL with the single-sampling version to have $\psi = 1$ and multi-sampling version with $\psi = S$. In the following empirical part we analyze the difference of the versions. The reward is sampled in a similar way with Normal-Gamma distribution.

Initialization: t:=1, and reset $N_k(s,a)$, $N_k(s,a,s')$, $R_k(s,a)$;
for *epoch k = 1, 2, ...* **do**
    Set $t_k := t$, $v_k(s,a) = 0$;
    Sampled ψ pairs of parameters $\hat{r}_k^j(s,a)$ and $Q_k^j(s,a)$ via equation (3);
    Compute optimal policy π^k through extended value iteration;
    while $v_k(s,a) < max\{1, N_k(s_t, \pi^k(s_t)\}$ **do**
        Execute π^k and update $N_k(s,a)$, $N_k(s,a,s')$, $R_k(s,a)$;
        $a_t = \pi^t(s_t)$;
        Observe next state $s_{t+1} = env(a_t)$;
        $t = t + 1$;
    end
end

Algorithm 2: PSRL Algorithm

3 MDP Setting

In this project we test and compare the algorithms in two different MDP problem. Both of the two MDPs are designed in a sparse way such that the high reward state is extremely hard to reach. Under such setting the algorithm has to conduct its exploration sufficiently otherwise would end up with a high total regret. Therefore, the empirical results from those setting is suitable to compare the algorithms' exploration efficiency.

3.1 RiverSwim MDP

RiverSwim is a classic MDP setting as shown in the Figure. 1. In this setting the states are aligned in a single chain like a river. The agent starts at the left end of the chain (bottom of the river) and can choose to swim left (downward) and right (upward). The left end state has a tiny reward while the right end states has a large reward.

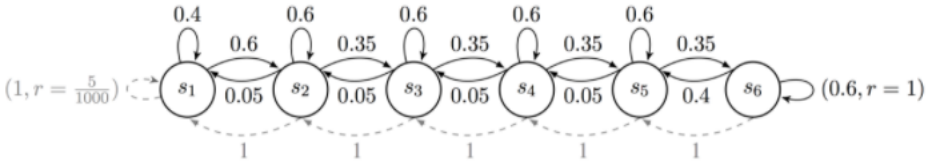


Figure 1: RiverSwim MDP

The optimal policy here is intuitively swim right. However, since there is no reward on any of the middle states, without sufficient exploration, the agent might end up with staying at the left end for the consistent tiny reward.

3.2 Tree MDP

Apart from the chain-like MDP, we also consider a tree-like MDP that discussed as the lower bound scenario for UCRL2's regret analysis. As shown in Figure. 2, there are two types of the state where all type-1 states are connected in a tree structure and type-2 states only connect to one type-1 state. Each type-1 state has A' leaf states, one root state and one type-2 state. None of the type-1 states has reward while all of the type-2 states have reward 1.

As for the transition matrix, the agent can transfer between type-1 states with probability 1 by taking according actions (A' actions to leaf states and 1 action to root state). For each type-1 and type-2 state pair, there are another A' actions from type-1 to type 2, the probability of moving to opposite states by taking any of the action are all δ except for one action in one pair. In such "good" state pair agent has probability of $\delta + \epsilon$ to move to type-2 states by playing such good action. In this MDP the total actions are $2A' + 1$. We add a constraint that $0 < \delta < \epsilon < \frac{1}{3}$.

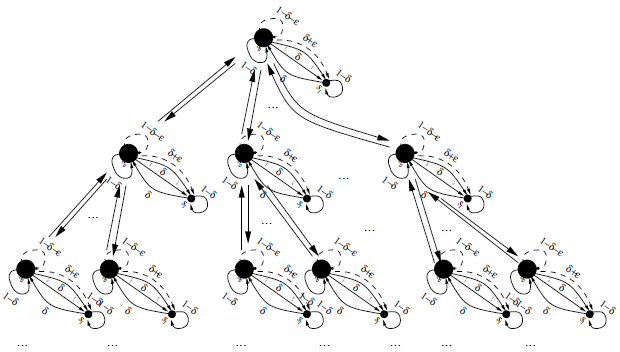


Figure 2: Tree MDP

In such setting the optimal policy is first transfer through the type-1 states to the "good" type-1 state, and then keep taking the "good" action with extra ϵ probability to the type-2 state. Due to the complex transition structure, it is difficult to traverse all the states and action pair and built well knowledge on all pairs. Meanwhile, as there is only one good state-action with an extra ϵ expected reward, the agent might also end up with a common branch and keep getting the average rewards.

4 Empirical Results

We conduct multiple numerical experiments under various simulation settings. In each experiment we simulate 32 instances and record the algorithms' total regret in all 32 paths. Based on the simulated regret paths, we compare the algorithms' performance with respect to the total regret and convergence speed.

4.1 Non-Bayesian Environment

With the non-Bayesian setting, we assume that the transition probability of the environment is a fix matrix. Therefore, the 32 simulated environments have the identical transition matrix and reward function. Here for RiverSwim MDP we take the default parameters shown in the Figure. 1 and simply replicate the middle states when increase the total state space. For Tree MDP we assume $\epsilon = 0.3$ and $\delta = 0.1$. For each experiment we calculate the average total regret among all 32 paths and summarized them in the table. 1.

Table 1: Average total regret for non-Bayesian Setting

Algorithm	*RiverSwim* $S = 6$ $T = 5 \times 10^5$	*RiverSwim* $S = 12$ $T = 10^6$	*Tree* $S = 6$ $T = 5 \times 10^6$	*Tree* $S = 14$ $T = 5 \times 10^7$
UCRL2	12966.5	173052.7	93898.5	888570.6
PSRL with single-sampling	1756.1	21001.2	2812.4	9277.8
PSRL with multi-sampling	1696.3	19059.1	2880.2	6565.6

The result shows that in all four test scenarios, both of the two versions of the PSRL achieve significantly lower total regret compared with UCRL2 algorithm. The average total regrets for the two versions of PSRL are close and different of two version are not statistically significant. For a large state space, the PSRL outperforms UCRL2 especially in the complex structure case (Tree MDP with $S = 14$)m which may related with the specific reward structure in Tree MDP's setting.

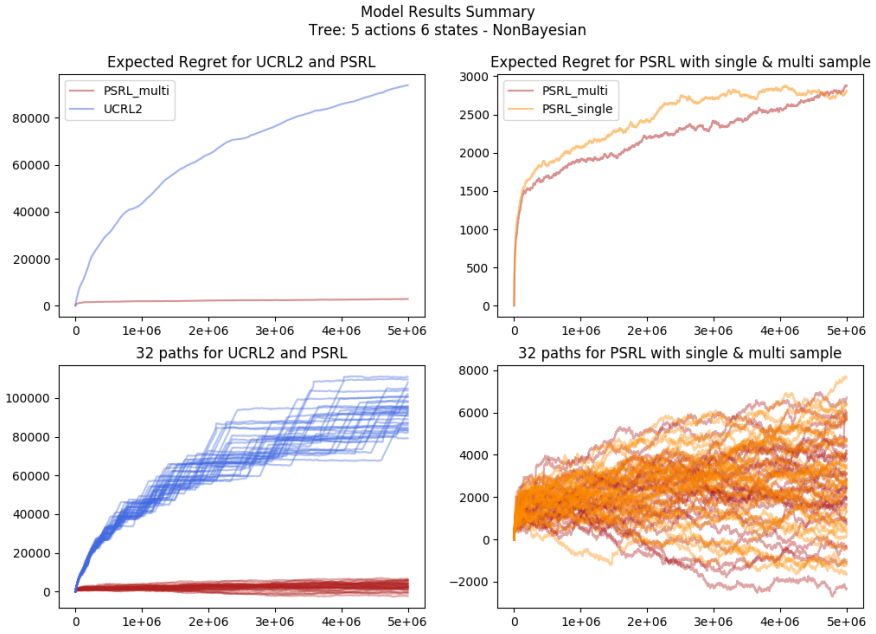


Figure 3: Tree MDP with $S = 6$

We visualized the regret path by compared both UCRL2 vs. PSRL-multi-sampling and PSRL-single-sampling vs. PSRL-multi-sampling with respect to both average total regret and all 32 single regret path. The whole results is summarized in Appendix-A and the Figure. 3 shows the example for Tree MDP with $S = 6$. From the regret plot it can be seen that the PSRL converges much faster than UCRL2. The piecewise regret curves of UCRL2 shows that it conducts the exploration in a less efficient way.

4.2 Bayesian Environment

With the Bayesian setting we assume the transition probability of the environments follows a certain prior distribution. Each time when we simulate an environments, we sample the probability from the prior distribution. Therefore, the 32 simulated environments have different transition matrix from same distribution. Here Dirichlet distribution is the natural candidate and we sample from it in both RiverSwim and Tree MDP.

We assume the Dirichlet distribution mean is same as the probability vectors of non-Bayesian setting. We further assume that the probability structure is the same as non-Bayesian setting, ie. the middle states in RiverSwim MDP keep sharing the same transition probability and the same case for Tree

MDP's). Take Tree for example, we only sample once from the $Dirichlet(\delta, \epsilon, 1-\delta-\epsilon)$ as δ and the ϵ are the only two parameters of probability and we keep it to be same along all state pairs. Finally, we add a feasible constraint. For RiverSwim MDP, $P(s_{t+1} = s_t + 1|s_t) > 2P(s_{t+1} = s_t - 1|s_t)$ for middle states $0 < s_t < S - 1$ and $P(s_{t+1} = min(s_t + 1, S - 1)|s) > P(s_{t+1} = max(s_t - 1, 0)|s_t)$ for two ends states $s_t \in \{0, S - 1\}$. For Tree MDP, $\delta < \epsilon$ and $0 < \delta + \epsilon < \frac{2}{3}$.

Table 2: Average total regret for Bayesian Setting

Algorithm	*RiverSwim* $S = 6$ $T = 5 \times 10^5$	*RiverSwim* $S = 12$ $T = 10^6$	*Tree* $S = 6$ $T = 5 \times 10^6$	*Tree* $S = 14$ $T = 5 \times 10^7$
UCRL2	12855.5	195276.6	452801.8	3428788.9
PSRL with single-sampling	2114.5	29896.0	87529.4	226280.2
PSRL with multi-sampling	2085.0	31174.4	95989.6	221700.7

We summarized the empirical results of average total regret in table. 2.It can be seen that the result is quite similar to the previous table. The PSRL algorithm significantly out perform the UCRL2 algorithm, while the PSRL with multi-sampling does not show an obvious improvement compared with single sampling version.

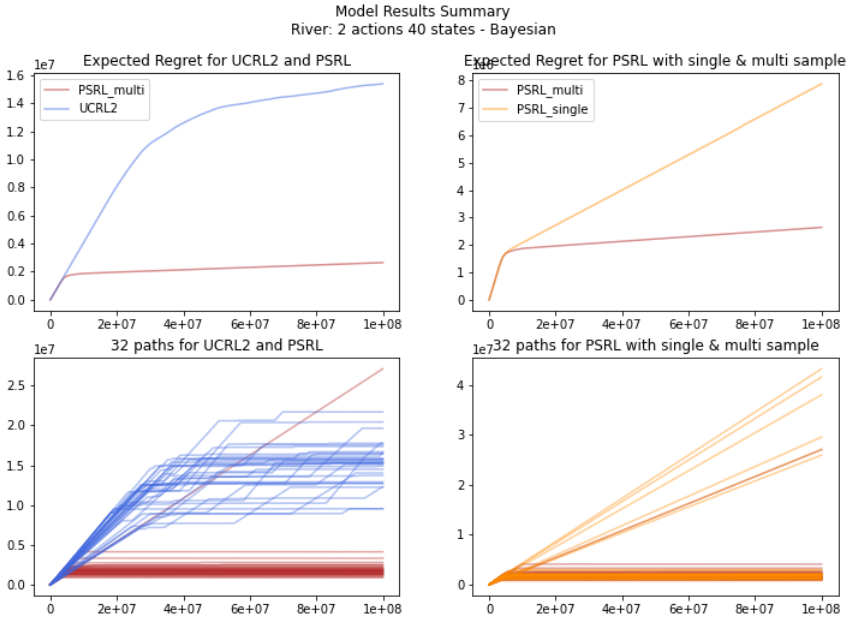


Figure 4: RiverSwim MDP with $S = 40$

To further distinguish the performance of the two version of PSRL, we test the algorithms in a much larger states: RiverSwim MDP with 40 states, with only the last state being fruitful. The regret plot for this scenario is shown in Figure. 4, where the PSRL with only single sampling achieves significantly larger regret compared to its multi-sampling version.

With a close look into the single regret path on bottom right, it can be seen that the main reason of the PSRL's regret difference is the proportion of convergence. For PSRL with single-sampling, 6 out 32 paths does not converge while the proportion is 1 out of 32 for multi-sampling version. The multi-sampling method leads to a lower variance on exploration and increase the algorithm's convergence probability.

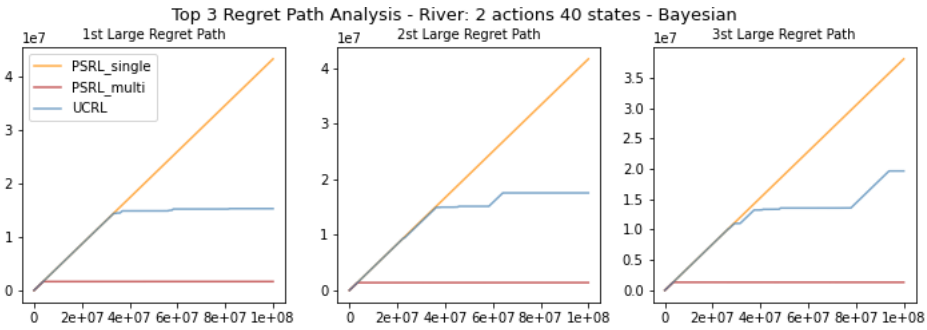


Figure 5: Top 3 regret simulated path for RiverSwim with S= 40

We visualize 3 regret paths where the PSRL does not converge. It can be seen that in those cases UCRL2 all converges with a lower total regret. Therefore, the reason of non-convergence is related to the algorithm instead of the environment. The detailed plot is summarized in Appendix-b.

5 Conclusion

In this project we conduct numerical experiments for model-based RL algorithms under both non-Bayesian and Bayesian setting. The results show that the Thompson Sampling method PSRL achieves a significantly lower total regret and faster converge speed compared with UCRL2 algorithm in both small and large states space cases. When the state space is small, PSRL with only single sampling achieves performs similarly with respect to the total regret compared with PSRL with multi-sampling. However, when the state space becomes large, single sampling method has higher probability to not converge and multi-sampling method can solve this problem efficiently.

References

[1] Auer, Peter, Thomas Jaksch, and Ronald Ortner. "Near-optimal regret bounds for reinforcement learning." Advances in neural information processing systems. 2009.

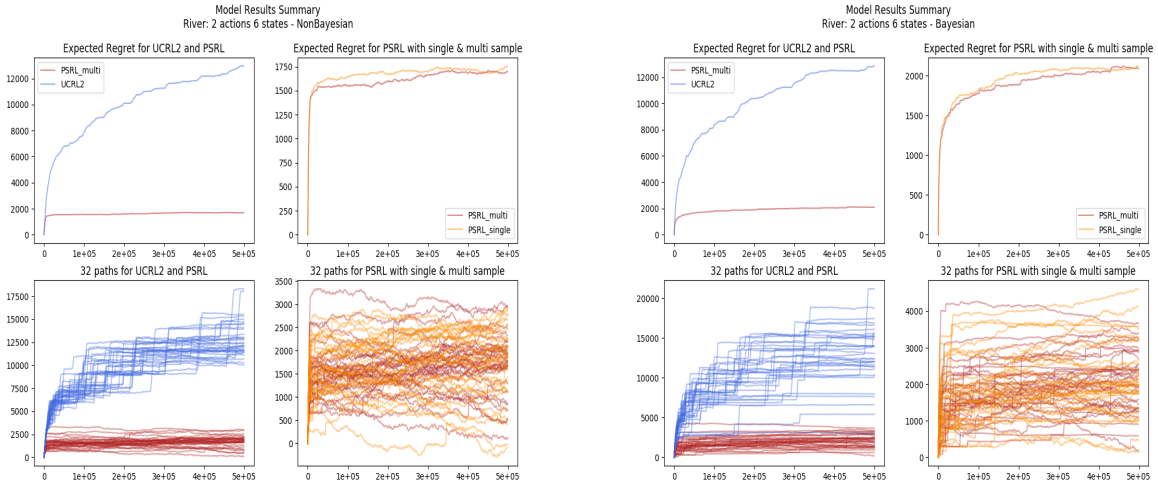
[2] Agrawal, Shipra, and Randy Jia. "Optimistic posterior sampling for reinforcement learning: worst-case regret bounds." Advances in Neural Information Processing Systems. 2017.

[3] Osband, Ian, Daniel Russo, and Benjamin Van Roy. "(More) efficient reinforcement learning via posterior sampling." Advances in Neural Information Processing Systems. 2013.

[4] Russo, Daniel, et al. "A tutorial on thompson sampling." arXiv preprint arXiv:1707.02038 (2017).

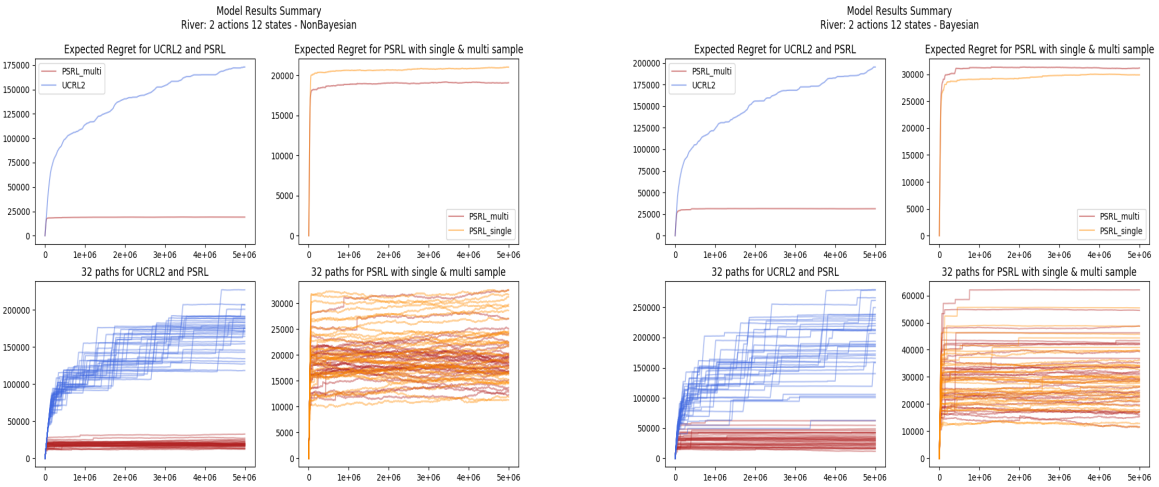
A Regret Path Plot

Here we summarized the path plots for all numerical experiments. Each plot includes four subplots, where the first row is the average total regret along all 32 paths and the second row present each of the single simulated paths. The first column compares UCRL2 with the PSRL with multi-sampling, while the second columns compares the two version of PSRL.



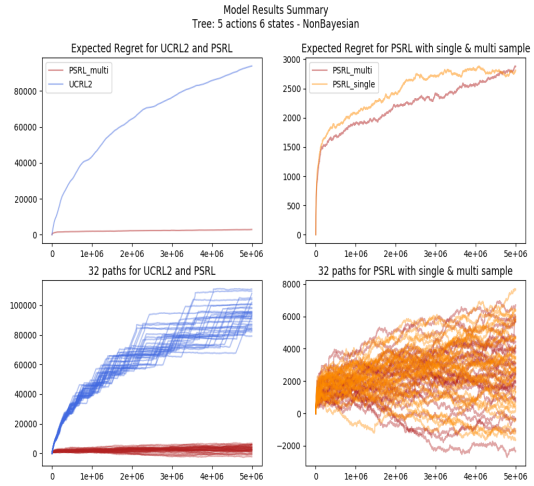
(a) Non-Bayesian Environment (b) Bayesian Environment

Figure 6: RiverSwim MDP with $S = 6$

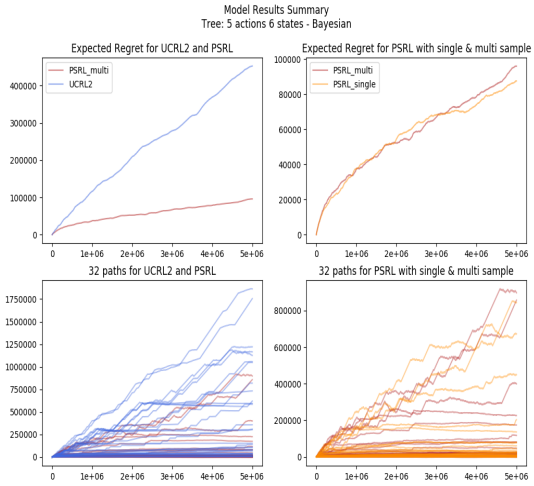


(a) Non-Bayesian Environment (b) Bayesian Environment

Figure 7: RiverSwim MDP with $S = 12$

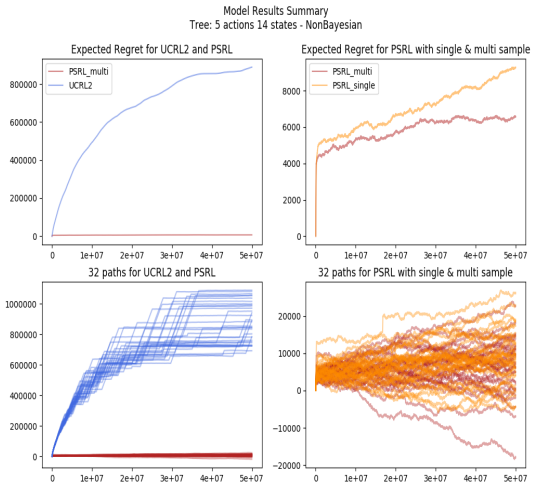


(a) Non-Bayesian Environment

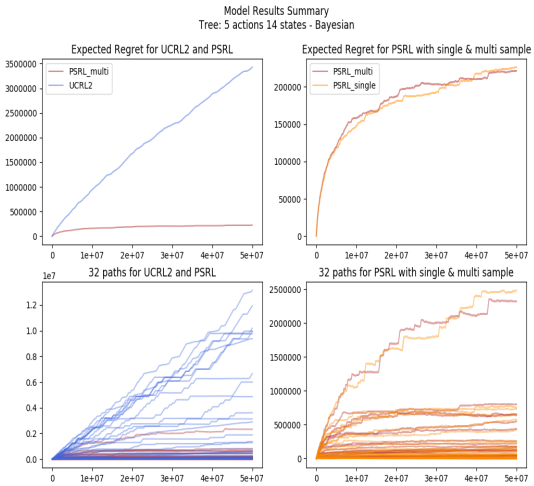


(b) Bayesian Environment

Figure 8: Tree MDP with $S = 6$



(a) Non-Bayesian Environment



(b) Bayesian Environment

Figure 9: Tree MDP with $S = 14$

B Non-Convergence Path Plot

Here we summarize the top 8 regret paths for single-sampling-PSRL for RiverSwim MDP with $S = 40$. It can be seen that the first 6 cases the single-sampling-PSRL all fail to converge while multi-sampling-PSRL only fail in one case.

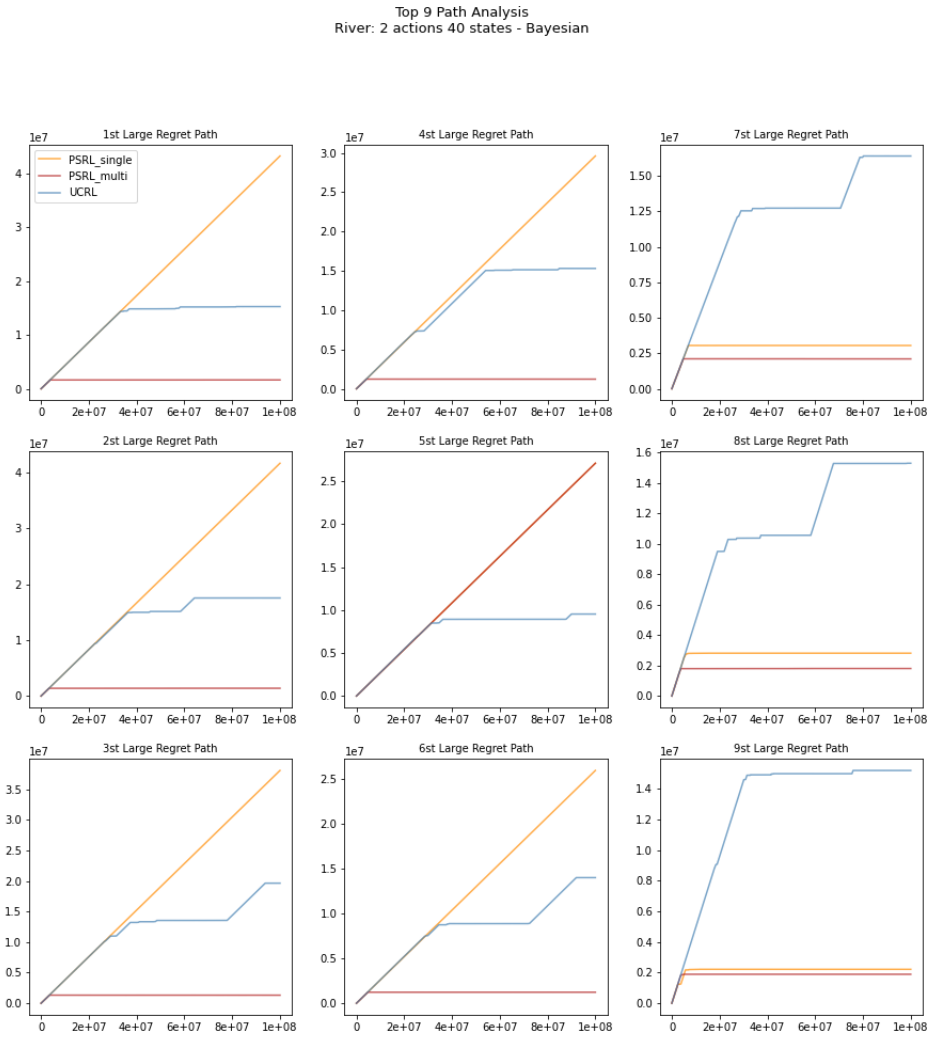


Figure 10: Top 8 regret simulated path for RiverSwim with S= 40