A Bayesian Approach to Learning Bandit Structure in Markov Decision Processes

Kelly W. Zhang
Department of Computer Science
Harvard University
kellywzhang@seas.harvard.edu

Omer Gottesman
Department of Computer Science
Brown University
omer_gottesman@brown.edu

Finale Doshi-Velez
Department of Computer Science
Harvard University
finale@seas.harvard.edu

Abstract

In the reinforcement learning literature, there are many algorithms developed for (1) Contextual Bandit problems and (2) Markov Decision Process (MDP) problems respectively. However, often when deploying reinforcement learning algorithms in the real world, even with domain expertise, it is often difficult to know whether it is appropriate to assume Contextual Bandit or MDP problem. Moreover, assuming the wrong problem setting can lead to inefficient learning, or worse, the algorithm could never learn the optimal policy, even with infinite data. In this work, we develop a reinforcement learning algorithm is able to have low regret in both Contextual Bandit and MDP environments by using Bayesian hypothesis testing approach to learn whether or not the MDP is that of a contextual bandit. In particular, we allow practitioners to choose a prior probability of the environment being that of a Contextual Bandit. We find that in simulations that our algorithm is able to perform similarly to MDP-based algorithms in non-bandit MDP settings, but also performs better than MDP algorithms in contextual bandit environments.

1 Introduction

Sequential decision making problems are traditionally analyzed in three different frameworks: (1) Multi-Arm and Contextual Bandits (CB), (2) Markov Decision Processes (MDP), and (3) Partially Observable MDPs (POMDPs). For contextual bandits action selections do not impact state transition probabilities, while for MDPs state transition probabilities are determined by the combination of the chosen state and action, and for POMDPs only part of the state is observable, so the entire history of states and actions can affect state transition probabilities. Reinforcement learning algorithms are typically developed assuming the environment falls strictly within one of these three frameworks. However, when deploying reinforcement learning algorithms in the real world, it is often unknown which one of these frameworks one should assume.

For example, consider a mobile health application aimed to help users increase their step count [9]. A few times a day, the sequential decision making algorithm uses the user's state information (e.g. user's recent app engagement, local weather, time of day, location, ect.) to decide whether or not to send the user a message encouraging them to take a walk. It's not immediately clear whether to use a contextual bandit or an MDP algorithm in this problem setting. It may be that whether the algorithm sends a notification strongly affects what future states the user will visit—for example, notifications that annoy the user could lead the user to enter states with low reward; this setting warrants the use of

The Challenges of Real World Reinforcement Learning Workshop at NeurIPS 2020, Vancouver, Canada.

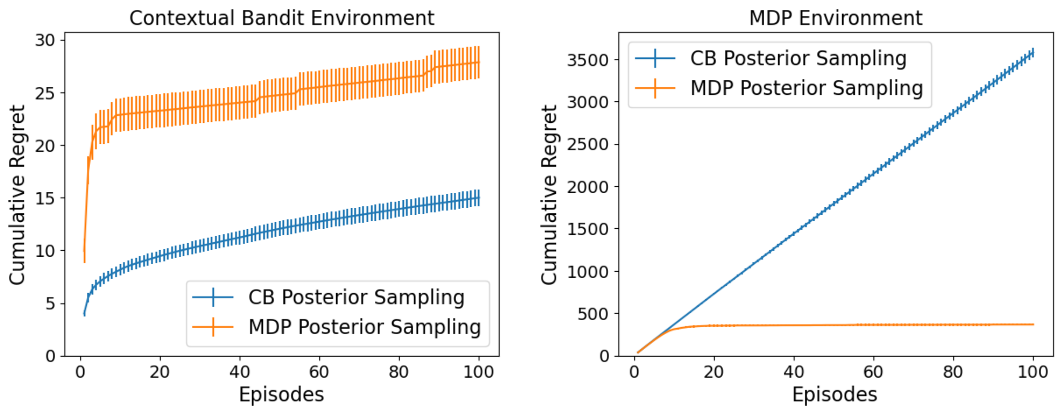


Figure 1: Above we plot the cumulative regret of a contextual bandit posterior sampling algorithm and a MDP posterior sampling algorithm in two environments (finite horizon 100, 6 states). The MDP environment is the river swim environment (see Figure 2). **The contextual bandit environment that is identical to the MDP environment, except the state transition probabilities are uniform over all states.** The error bars are standard errors for the estimates.

an MDP algorithm. However, it may be that what states the user enters does not depend much on past notifications—for example, whether the user enters high reward states depends on the user's work schedule. In this setting, a bandit algorithm could perform well on the problem, and using a full MDP algorithm would require more data to learn the optimal policy and incur greater regret. Assuming the wrong framework can lead to inefficient learning, or worse, the algorithm could never learn the optimal policy, even with infinite data. Below in Figure 1, we demonstrate the loss in total reward of using an MDP posterior sampling algorithm in a contextual bandit environment and the consequences of using a contextual bandit posterior sampling algorithm in an MDP environment.

In this work, we consider the problem in which the practitioner is unsure whether the environment is that (1) a Contextual Bandit or (2) an MDP. Note that a "contextual bandit" environment is a special case of MDPs in which for each state the probabilities of transitioning to any other given state doesn't change depending on the choice of action. Specifically we develop the *Bayesian Hypothesis Testing Reinforcement Learning (BHT-RL)* algorithm that allows practitioners to incorporate their uncertainty regarding whether the environment is that of a contextual bandit or MDP as a prior used by the algorithm. We find that empirically BHT-RL has (1) lower regret than MDP-based algorithms when the environment is that of a contextual bandit and (2) regret comparable to that of MDP-based algorithms in MDP environments.

Our algorithm is based on posterior sampling and utilizes Bayesian Hypothesis Testing to learn whether the environment is that of a contextual bandit a classical MDP. We allow practitioners to choose a prior probability between zero and one that the environment is that of a contextual bandit. Additionally, practitioners choose two reinforcement learning algorithms: one contextual bandit algorithm and one MDP algorithm. The choice of prior probability allows the practitioner to "interpolate" between a contextual bandit algorithm and a full MDP algorithm. Specifically, when the prior probability of a contextual bandit environment is one, then our algorithm is equivalent to just using the contextual bandit algorithm; when the prior probability is zero, then our algorithm is equivalent to just using the MDP algorithm of choice. Additionally, when posterior sampling algorithms are chosen for the contextual bandit and MDP algorithms, then the BHT-RL algorithm can be interpreted as posterior sampling with an additional prior that can up-weight the prior probability that environment is that of a bandit; thus, the Bayesian regret bounds for standard posterior sampling for MDPs can apply in this setting [11].

We demonstrate that our Bayesian Hypothesis Testing approach performs well empirically in both contextual bandit and MDP environments compared to contextual bandit and MDP algorithms in their respective misspecified environment. Moreover, our BHT-RL approach results in significantly better regret minimization empirically in both contextual bandit and MDP environments compared prior work on learning bandit structure in MDPs [15, 16].

2 Related Work

Reinforcement Learning Algorithms for both Contextual Bandits and MDPs There have been several works developing algorithms that have low regret in both contextual bandit and MDP environments—or more generally on all contextual decision processes. Jiang et al. 2017 introduced the term “contextual decision processes", which encompass Contextual Bandits, MDPs, and POMDPs [7]. They also develop the OLIVE algorithm and prove bounds for it in a variety of contextual decision processes when the the Bellman rank is known. However, since the Bellman rank of problems is generally unknown in real world problems, OLIVE is not a practically usable algorithm.

Zanette and Brunskill 2018 develop the UBEV-S reinforcement learning algorithm, which they prove has near optimal regret in the MDP setting and has regret that scales better than that of OLIVE in the contextual bandit setting [15]. Later, Zanette and Brunskill 2019 developed the EULER algorithm, which improves upon UBEV-S, and has optimal regret bounds (excluding log factors) in both the contextual bandit and MDP settings [16]. Both UBEV-S and EULER are upper confidence bound based methods that construct confidence bounds for the next timestep reward and future value, and then execute the most optimistic policy within those bounds.

Even though UBEV-S and EULER provably have regret that scales near optimally, we find that in simulation environments that BHT-PSRL outperforms both UBEV-S and EULER. It’s been shown in previous work that posterior sampling reinforcement learning algorithms generally outperform confidence bound based algorithms empirically [11, 12, 13]. As discussed in Osband and Van Roy 2017, one reason for the poor empirical performance of confidence bound based algorithms is that the confidence bounds used are often loose and solving for the most optimistic policy within these bounds often leads to choosing policies that are optimal for relatively unlikely MDPs [12].

Regularizing RL Algorithms by Using a Shorter Planning Horizon An open problem in reinforcement learning theory is understanding how sample complexity depends on the planning horizon (in infinite horizon problems this is the discount factor) [6]. Jiang et al. 2015 showed that longer planning horizons (larger discount factors) increases the size of the set of policies one searches over [8]. As a result, often it is better to use a smaller planning horizon than the evaluation horizon as a method of regularization to prevent overfitting to the data. We find empirically that in MDP environments that BHT-PSRL often slightly outperforms PSRL, which makes sense because BHT-PSRL can be considered a regularized version of typical PSRL in which we encourage the learning algorithm to use a shorter planning horizon early on in learning.

Posterior Sampling Reinforcement Learning There is a long history of using Bayesian methods in reinforcement learning [5]. Among one of the first papers to propose posterior sampling in reinforcement learning problems is Strens 2000 [14]; however, they do not prove any regret bounds for their algorithm. Russo and Van Roy 2014, proved Bayesian regret bounds for posterior sampling in contextual bandit environments [13] and Osband et al. 2013 was the first to prove Bayesian regret bounds for posterior sampling in MDP environments [11]. Note that while there are proofs for frequentist regret bounds for posterior sampling under certain priors [1], there are currently no frequentist regret bounds for posterior sampling on MDPs.

For our BHT-RL algorithm, we will pool the state transition for different actions in the same state together. Asmuth et al. 2009, [2] call this approach the tied Dirichlet model. However, they also assume that the experimenter has apriori knowledge and chooses before the study is run whether to assume the tied or regular Dirichlet model on the transition probabilities. In contrast, we will aim to *learn* whether in each state it is better to use the tied Dirichlet model or the standard one. To do this we will use Bayesian hypothesis testing [3]. Bayesian hypothesis testing is related to Bayesian model selection because the posterior probabilities of the null versus the alternative models are a function of the Bayes factor, which is used in model selection to compare the relative plausibilities of two different models or hypotheses.

3 Bayesian Hypothesis Testing Reinforcement Learning

3.1 Problem Setting

We define random variables for the states $S_t \in \mathcal{S}$, random variables for the action selections $A_t \in \mathcal{A}$, and rewards $R_t \in \mathbb{R}$. We also define ν, which is a random variable representing the environment; for example, given ν we know the expected rewards $R(s, a, \nu) = \mathbb{E}[R_t | A_t = a, S_t = s]$ and the transition probabilities $P(s'|s, a, \nu) = \mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a)$. We assume that we are in the finite-horizon episodic setting, so the data collected is made up of episodes each of length H. For example, for the k^{th} episode, we have the data $(A_{t_k+h}, S_{t_k+h}, R_{t_k+h})_{h=1}^{H}$, where $t_k := kH$. We define T to be the total number of data tuples total seen in total so far. We also define $m := \lceil T/H \rceil$ to be the total number of episodes seen so far. We define $\mathcal{H}_{t_k} = \{(S_{t_k+h}, A_{t_k+h}, R_{t_k+h})_{h=1}^{H}\}_{k'=0}^{k-1} = \{(A_t, S_t, R_t\}_{t=1}^{t_k}$ to be the random variable for the history following policy π. Note that we define our policies to be π_k to be $\sigma(\mathcal{H}_{t_k})$-measurable functions from $\mathcal{S} \times [1\colon H]$ to $|\mathcal{A}|$-dimensional simplex. So, our actions $A_{t_k+h} \sim \pi_k(S_{t_k+h}, h)$ are chosen according to the policy. Note that the policy takes the time-step in the episode, h, as an input because in the finite horizon setting the optimal policy can change depending on the timestep in the episode. For example, in the beginning it might make more sense to sacrifice immediate reward in order too get to a high reward state later in the episode; however, at the end of the episode the optimal policy will optimize the immediate reward.

3.2 Algorithm Definition

For our Bayesian Hypothesis Testing method we define the following null and alternative hypotheses:

Null hypothesis H_0**:** *Action selections do not affect transition probabilities, i.e.* $\mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a) = \mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a')$ *for all* $a, a' \in \mathcal{A}$, $s, s' \in \mathcal{S}$.

Under the null hypothesis we model our data as generated by the following process:

- For each $s \in \mathcal{S}$ we draw $\boldsymbol{\varphi}_s \sim \text{Dirichlet}(\boldsymbol{\alpha})$
- For all $t \in [1\colon T]$ such that $S_t = s$, we have that $S_{t+1} \sim \text{Categorical}(\boldsymbol{\varphi}_s)$

Alternative hypothesis H_0**:** *Action selections do affect transition probabilities, i.e.* $\mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a) \neq \mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a')$ *for some* $a, a' \in \mathcal{A}$, $s, s' \in \mathcal{S}$.

Under the alternative hypothesis we model our data as generated by the following process:

- For each $s \in \mathcal{S}$ and each $a \in \mathcal{A}$ we draw $\boldsymbol{\varphi}_{s,a} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
- For all $t \in [1\colon T]$ such that $S_t = s$ and $A_t = a$, we have that $S_{t+1} \sim \text{Categorical}(\boldsymbol{\varphi}_{s,a})$

We choose prior probabilities over the hypotheses $P(H_0)$ and $P(H_1) = 1 - P(H_0)$. Practically, for someone utilizing the algorithm, the choice of $P(H_0)$ would be how likely they think that the environment is that of a bandit, based on domain knowledge. Then given we've run k episodes already we can compute the posterior probabilities $P(H_0|\{\bar{S}_{t_{k'}}\}_{k'=0}^{k})$ and $P(H_1|\{\bar{S}_{t_{k'}}\}_{k'=0}^{k})$, where $\bar{S}_{t_k} := \{S_{t_k+h}\}_{h=1}^{H}$.

$$P(H_0|\mathcal{H}_T) = \frac{P(H_0, \mathcal{H}_T)}{P(\mathcal{H}_T)} = \frac{P(\mathcal{H}_T|H_0)P(H_0)}{P(\mathcal{H}_T|H_0)P(H_0) + P(\mathcal{H}_T|H_1)P(H_1)} = \frac{1}{1 + \frac{P(H_1)}{P(H_0)}K}$$

where $K = \frac{P(\mathcal{H}_T|H_1)}{P(\mathcal{H}_T|H_0)}$ is the Bayes factor.

Algorithm 1: Bayesian Hypothesis Testing Reinforcement Learning (BHT-RL)

Input: Prior distribution on MDPs Q; prior probability of nullness $P(H_0)$; generative models under H_0 and H_1 respectively; contextual bandit algorithm π^{CB} and MDP algorithm π^{MDP}

for *episodes* $k = 0, 1, 2, ...$ **do**
  Sample indicator of generative model $B_k \sim \text{Bernoulli}\big(P(H_0|\{\bar{S}_{t_{k'}}\}_{k'=0}^{k})\big)$
  if $B_k = 1$ **then**
    // Follow bandit algorithm of choice
    Let $\pi_k = \pi_k^{\text{CB}}$ the contextual bandit algorithm
  else
    // Follow MDP algorithm of choice
    Let $\pi_k = \pi_k^{\text{MDP}}(\mathcal{H}_{t_k})$ the MDP algorithm
  for *timesteps* $h = 1, 2, ..., H$ **do**
    Sample and apply $a_t = \pi_k(s_{t_k+h}, h)$
    Observe r_{t_k+h} and s_{t_k+h+1}
  end
  Update both π_k^{CB} and π_k^{MDP} with data $\{s_{t_k+h}, a_{t_k+h}, r_{t_k+h}\}_{h=1}^{H}$ observed in the episode.
end

Note that the Bayesian Hypothesis testing approach can be used with any choice of (1) a contextual bandit algorithm and (2) an MDP based algorithm. Note that if we set prior probability of the null $P(H_0)$ to 1 the BHT-RL algorithm is equivalent to policy π_k^{CB} and when setting $P(H_0)$ to 0 the BHT-RL algorithm is equivalent to policy π_k^{MDP}.

If one chooses posterior sampling methods for the contextual bandit and MDP algorithms, then BHT-RL can be interpreted as posterior sampling with a hierarchical prior. Under posterior sampling, a prior is put on the parameters of the environment, i.e., we assume that for some prior distribution Q, that $\nu \sim Q$. The policy for that episode is selected by first sampling $\nu_k \sim Q(\cdot|\mathcal{H}_{t_k})$, where $Q(\cdot|\mathcal{H}_{t_k})$ is the posterior distribution over ν. Then the policy for the episode π_k is chosen to be the optimal policy for environment ν_k. When using BHT-RL with posterior sampling CB and MDP algorithms, we have that π_k^{CB} is the optimal policy for $\nu_k \sim Q(\cdot|\mathcal{H}_{t_k}, H_0)$, the posterior distribution of Q given that the null hypothesis H_0 is true. Similarly, π_k^{MDP} is the optimal policy for $\nu_k \sim Q(\cdot|\mathcal{H}_{t_k}, H_1)$.

3.3 Regret Guarantees

We now define regret in the episodic setting. We first define the value function, which is the expected value of following some policy π during an episode:

$$V_{\pi,h}^{\nu}(s) = \mathbb{E}_{\pi,\mathcal{H}}\left[\sum_{h'=h}^{H} R(S_{h'}, A_{h'})\middle|\nu\right]$$

Above, the expectation is taken over randomness in the policy π and the randomness in the history $\mathcal{H}$. We define $\pi^*(\nu)$ to be the optimal policy for some MDP (or contextual bandit) environment ν; barring computational issues, the optimal policy for a given MDP environment can be solved for using dynamic programming.

The frequentist regret is defined as the difference in total expected reward for the optimal policy versus the actual policy used:

$$\mathcal{R}_m(\pi, \nu) = \sum_{k=0}^{m}\sum_{s\in\mathcal{S}} \rho(s)\big(V_{\pi^*,1}^{\nu}(s) - V_{\pi_k,1}^{\nu}(s)\big)$$

Above $\rho(s)$ represents the probability of starting the episode in state s, so $\sum_{s\in\mathcal{S}} \rho(s)$. For the Bayesian regret, we assume that the MDP environment ν is drawn from prior distribution Q. The Bayesian regret is defined as follows:

$$\mathcal{BR}_m(\pi, \nu) = \mathbb{E}_{\nu\sim Q}[\mathcal{R}_m(\pi, \nu)]$$

Note that frequentist regret bounds are automatically Bayesian regret bounds, as they must hold for the worst case environment ν. Bayesian regret bounds generally assume that the algorithm knows the prior on the environment Q.

Theorem 1 (Bayesian Regret Bound for MDP Posterior Sampling)**.** *Let Q be the prior distribution over ν used by the MDP posterior sampling algorithm. Let rewards $R_t \in [0, C]$, for some constant $0 < C < \infty$. Then,*

$$\mathcal{BR}_m(\pi, \nu) = \mathbb{E}_{\nu \sim Q}[\mathcal{R}_m(\pi, \nu)] = O(HS\sqrt{AT \log(SAT)})$$

Osband et al. 2013 prove that posterior sampling on MDPs has Bayesian regret $\tilde{O}(HS\sqrt{AT})$, as stated in Theorem 1 [11]. Since BHT-RL with posterior sampling contextual bandit and MDP algorithms is simply posterior sampling with a hierarchical prior, we can apply the regret bound of Theorem 1. Thus, BHT-RL with posterior sampling CB and MDP algorithms has Bayesian regret $\tilde{O}(HS\sqrt{AT})$. See [13, 11] for a discussion about how when the prior is misspecified, Bayesian regret bounds often translate to asymptotic regret bounds.

Corollary 1 (Bayesian Regret Bound for BHT-RL with Posterior Sampling)**.** *Suppose we use BHT-RL with posterior sampling contextual bandit and MDP algorithms. Let $P(H_0) \in [0, 1]$ be the prior probability of null hypothesis. Let $Q(\cdot|H_0)$ and $Q(\cdot|H_1)$ be the prior distribution over ν conditional on the null and alternative hypotheses respectively. When rewards $R_t \in [0, C]$, for some constant $0 < C < \infty$,*

$$\mathcal{BR}_m(\pi, \nu) = \mathbb{E}_{\nu \sim Q}[\mathcal{R}_m(\pi, \nu)] = O(HS\sqrt{AT \log(SAT)})$$

where distribution Q over ν is defined as $Q := P(H_0)Q(\cdot|H_0) + P(H_1)Q(\cdot|H_1)$.

Corollary 1 follows directly from Theorem 1 because BHT-RL with contextual bandit and MDP algorithms is equivalent to posterior sampling with prior distribution $Q := P(H_0)Q(\cdot|H_0) + P(H_1)Q(\cdot|H_1)$.

4 Simulation Results

We perform simulations in different finite state, finite action contextual bandit and MDP environments. We use the river swim environment from Figure 2 of [11], which is a particularly difficult MDP environment to learn in. The optimal policy (unless near the end of the episode) is to always choose action right, for which there will be a small probability of moving right; once reaching s_6, where there will be a large reward upon reaching there. Choosing the left action will move the agent left with probability 1; in state s_1 there is a small reward. Note that the rewards are deterministic and have no noise.

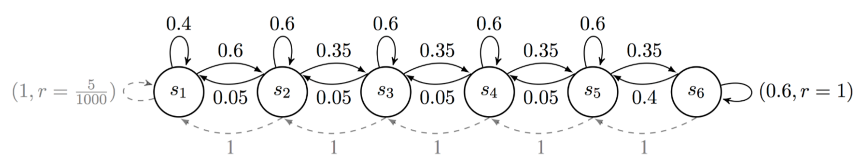


Figure 2: River Swim Environment (MDP); figure from Osband et al. 2013 [11].

We modify the River Swim environment to construct a "contextual bandit River Swim environment" in which the the probability of transitioning to any other given state is always $\frac{1}{6}$ regardless of the starting state or action selection; the rewards are the same as in the original River Swim MDP environment. Finally, we also construct the "interpolated River Swim environment", in which we modify the transition probabilities in the River Swim environment to with probability $\frac{1}{2}$ be according to the River Swim MDP environment and with probability $\frac{1}{2}$ be according to the River Swim contextual bandit environment. Note that the "interpolated" environment is still a classical MDP.

Simulation Hyper-Parameters

- For Bandit and MDP posterior sampling we have independent $\mathcal{N}(1, 1)$ priors on the rewards.
- For MDP posterior sampling we have Dirichlet ($\boldsymbol{\alpha} = [1, 1, ..., 1] \in \mathbb{R}^S$) priors on the transition probabilities.

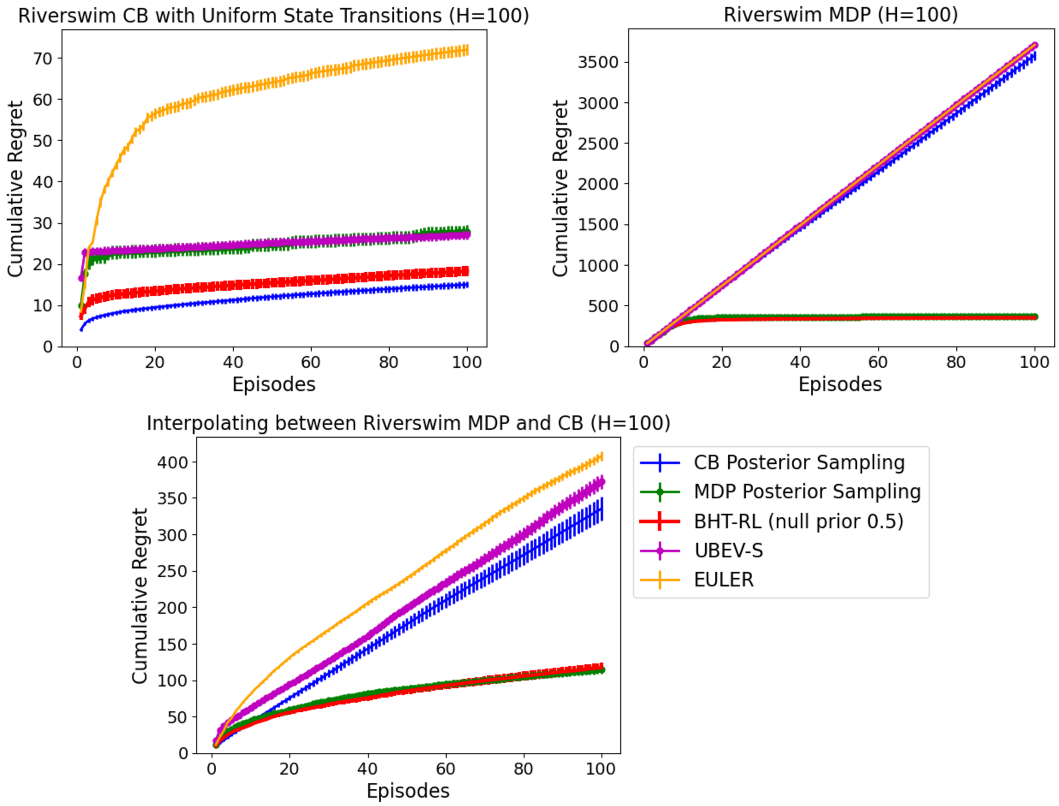
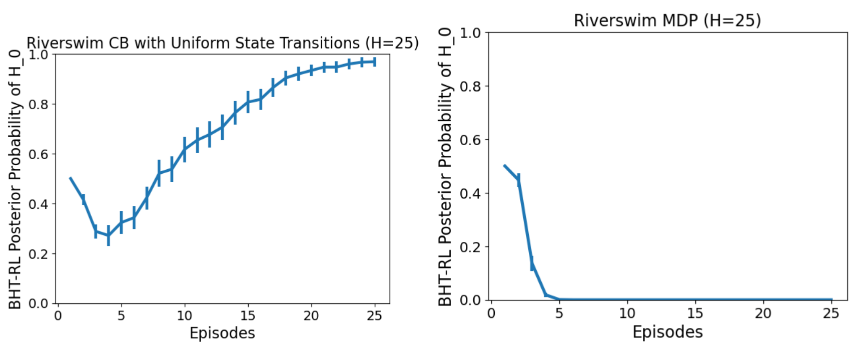


Figure 3: These simulations in three different environment with horizon $H = 100$. (1) **Top left**: modified River Swim contextual bandit environment; the probability of transitioning to any other state is always $\frac{1}{6}$. (2) **Top right**: original River Swim MDP environment of Figure 2. (3) **Bottom**: interpolated environment; with probability $\frac{1}{2}$ has the transition probabilities of the original River Swim MDP and with probability $\frac{1}{2}$ has the transitions probabilities of the River Swim contextual bandit environment.



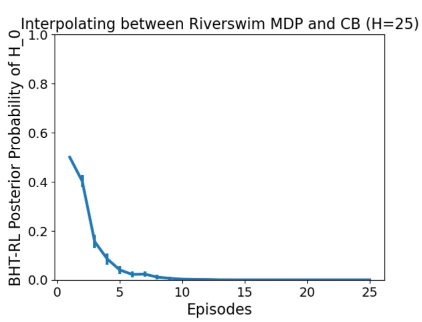


Figure 4: Here we plot the posterior probability of the null hypothesis $\mathbb{P}(H_0|\mathcal{H}_T)$ for BHT-RL in three different simulation environments with horizon $H = 25$. We see that in the CB environment that the posterior probability converges to 1. In the MDP and interpolated MDP and CB settings, we see that the posterior probability converges to 0.

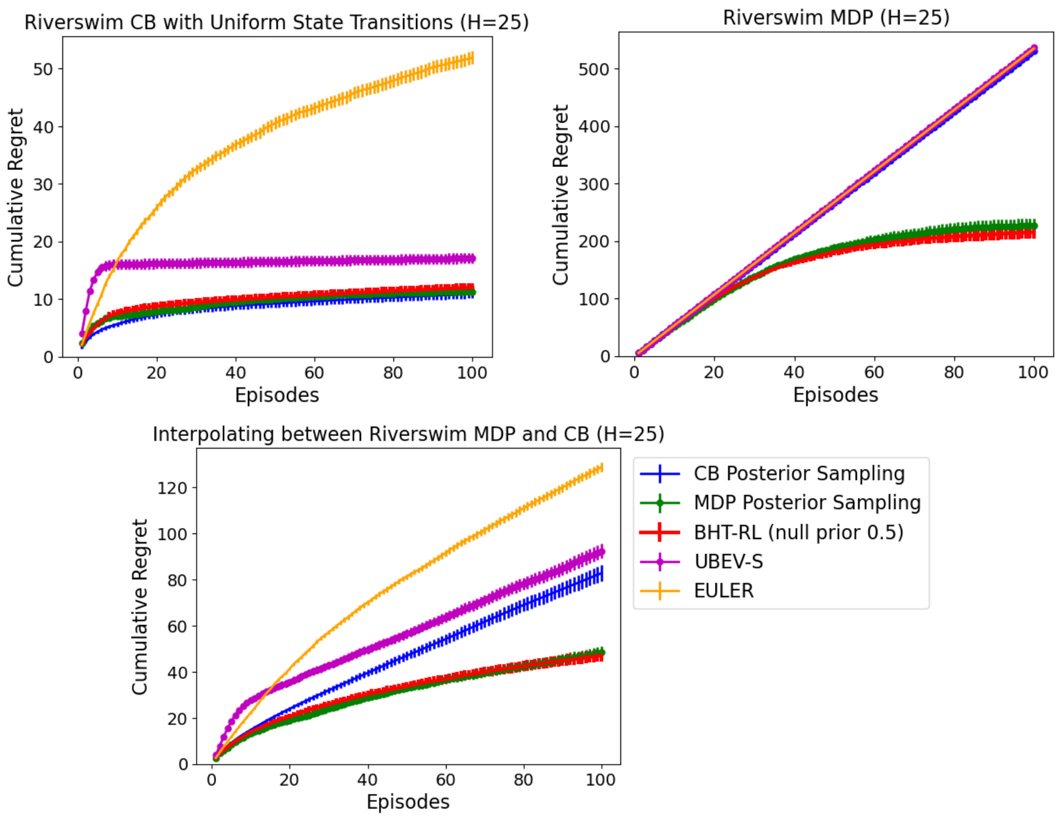


Figure 5: These simulations in three different environment with horizon $H = 25$.

- For BHT-PSRL we set the probability of the null hypothesis to $P(H_0) = 0.9$.
- UBEV-S and EULER we choose failure probability $\delta = 0.1$.
- We add $\mathcal{N}(0, 1)$ noise to all rewards.

5 Discussion

Our simulation results show that at least in finite state MDP and contextual bandit environments, the BHT-RL algorithm can perform well even when the environment is misspecified. Additionally, the BHT-RL approach allows practitioners to easily incorporate prior knowledge about the environment dynamics into their algorithm. Finally, BHT-RL can also be used as a regularization method for the full MDP based algorithm.

Some limitations of our work are that we only examine a relatively simplistic test bed. Additionally, there may be other theoretical guarantees we'd like to show about the BHT-RL algorithm, like a frequentist regret bound or a regret bound when the prior is misspecified. Finally, the BHT-RL algorithm relies heavily on the stationarity of the environment dynamics; thus, our method is not particularly robust to non-stationarity, which is often encountered in real world sequential decision making problems.

Beyond just learning whether the environment is that of a contextual bandit or an MDP, we conjecture that bayesian hypothesis testing could also be used to address other aspects of reinforcement learning problems. One example is learning better state representations [10], which is a major open problem in the reinforcement learning field [4].

Acknowledgments and Disclosure of Funding

Research reported in this paper was supported by National Institute on Alcohol Abuse and Alcoholism (NIAAA) of the National Institutes of Health under award number R01AA23187, National Institute on Drug Abuse (NIDA) of the National Institutes of Health under award number P50DA039838, and National Cancer Institute (NCI) of the National Institutes of Health under award number U01CA229437. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745303. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5):1–24, 2017.

[2] John Asmuth, Lihong Li, Michael L Littman, Ali Nouri, and David Wingate. A bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 19–26. AUAI Press, 2009.

[3] Jim Berger. Bayesian hypothesis testing. 2012.

[4] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[5] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *arXiv preprint arXiv:1609.04436*, 2016.

[6] Nan Jiang and Alekh Agarwal. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Conference On Learning Theory*, pages 3395–3398, 2018.

[7] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1704–1713, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[8] Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1181–1189. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[9] Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22, 2020.

[10] Ronald Ortner, Matteo Pirotta, Alessandro Lazaric, Ronan Fruit, and Odalric-Ambrym Maillard. Regret bounds for learning state representations in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 12717–12727, 2019.

[11] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3003–3011. Curran Associates, Inc., 2013.

[12] Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2701–2710. JMLR. org, 2017.

[13] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[14] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[15] Andrea Zanette and Emma Brunskill. Problem dependent reinforcement learning bounds which can identify bandit structure in MDPs. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5747–5755, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[16] Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7304–7312, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

A Bayesian Hypothesis Testing for Dirichlet Priors on Transition Probabilities

We define the set of states as $\mathcal{S}$ and the set of actions as $\mathcal{A}$. Suppose we have data $\mathcal{H}_T = \{S_t, A_t, R_t\}_{t=1}^T$.

- **Null hypothesis** H_0: We model our data as follows / Our data was generated as follows
  - For each $s \in \mathcal{S}$ we draw $\boldsymbol{\varphi}_s \sim \text{Dirichlet}(\boldsymbol{\alpha})$
  - For all $t \in [1\colon T]$ such that $S_t = s$, we have that $S_{t+1} \sim \text{Categorical}(\boldsymbol{\varphi}_s)$
- **Alternative hypothesis** H_1: We model our data as follows / Our data was generated as follows
  - For each $s \in \mathcal{S}$ and each $a \in \mathcal{A}$ we draw $\boldsymbol{\varphi}_{s,a} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
  - For all $t \in [1\colon T]$ such that $S_t = s$ and $A_t = a$, we have that $S_{t+1} \sim \text{Categorical}(\boldsymbol{\varphi}_{s,a})$

We choose prior probabilities over the hypotheses $P(H_0)$ and $P(H_1) = 1 - P(H_0)$. Then we can calculate the posterior probabilities $P(H_0|\mathcal{D})$ and $P(H_1|\mathcal{D})$

$$P(H_0|\mathcal{H}_T) = \frac{P(H_0, \mathcal{H}_T)}{P(\mathcal{H}_T)} = \frac{P(\mathcal{H}_T|H_0)P(H_0)}{P(\mathcal{H}_T|H_0)P(H_0) + P(\mathcal{H}_T|H_1)P(H_1)} = \frac{1}{1+K}$$

where $K = \frac{P(\mathcal{H}_T|H_1)P(H_1)}{P(\mathcal{H}_T|H_0)P(H_0)}$ is the Bayes factor.

Let us now derive the posterior distributions. Let $\Theta := \{\boldsymbol{\varphi}_s\}_{s\in\mathcal{S}} \cup \{\boldsymbol{\varphi}_{s,a}\}_{s\in\mathcal{S},a\in\mathcal{A}}$.

$$P(\Theta|\mathcal{H}_T) = \frac{P(\Theta, \mathcal{H}_T)}{P(\mathcal{H}_T)} = \frac{P(\mathcal{H}_T|\Theta)P(\Theta)}{\int P(\mathcal{H}_T|\Theta)P(\Theta)d\Theta} =: \frac{X}{Y}$$

First examining the numerator term X,

$$X = P(\mathcal{H}_T|\Theta)\big[P(\Theta|H_0)P(H_0) + P(\Theta|H_1)P(H_1)\big]$$

$$= P(H_0)\prod_{s\in\mathcal{S}}\left[\text{Dirichlet}(\boldsymbol{\varphi}_s;\boldsymbol{\alpha})\prod_{t=1}^{T}\text{Categorical}(S_{t+1};\boldsymbol{\varphi}_s)^{\mathbb{I}_{S_t=s}}\right] + P(H_1)\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\left[\text{Dirichlet}(\boldsymbol{\varphi}_{s,a};\boldsymbol{\alpha})\prod_{t=1}^{T}\text{Categorical}(S_{t+1};\boldsymbol{\varphi}_{s,a})^{\mathbb{I}_{S_t=s,A_t=a}}\right]$$

$$= \frac{P(H_0)}{B(\boldsymbol{\alpha})^{|\mathcal{S}|}}\prod_{s\in\mathcal{S}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_s(s')^{\alpha(s')-1}\prod_{t=1}^{T}\varphi_s(s')^{\mathbb{I}_{S_t=s,S_{t+1}=s'}}\right] + \frac{P(H_1)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_{s,a}(s')^{\alpha(s')-1}\prod_{t=1}^{T}\varphi_{s,a}(s')^{\mathbb{I}_{S_t=s,A_t=a,S_{t+1}=s'}}\right]$$

$$= \frac{P(H_0)}{B(\boldsymbol{\alpha})^{|\mathcal{S}|}}\prod_{s\in\mathcal{S}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_s(s')^{\alpha(s')-1+\sum_{t=1}^{T}\mathbb{I}_{S_t=s,S_{t+1}=s'}}\right] + \frac{P(H_1)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_{s,a}(s')^{\alpha(s')-1+\sum_{t=1}^{T}\mathbb{I}_{S_t=s,A_t=a,S_{t+1}=s'}}\right]$$

$$= \frac{P(H_0)}{B(\boldsymbol{\alpha})^{|\mathcal{S}|}}\prod_{s\in\mathcal{S}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_s(s')^{\alpha(s')-1+\sum_{t=1}^{T}\mathbb{I}_{S_t=s,S_{t+1}=s'}}\right] + \frac{P(H_1)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\left[\prod_{s'=1}^{\mathcal{S}}\varphi_{s,a}(s')^{\alpha(s')-1+\sum_{t=1}^{T}\mathbb{I}_{S_t=s,A_t=a,S_{t+1}=s'}}\right]$$

We define $\mathbf{N}_s = [\sum_{t=1}^T \mathbb{I}_{S_t=s,S_{t+1}=1}, \sum_{t=1}^T \mathbb{I}_{S_t=s,S_{t+1}=2}, ..., \sum_{t=1}^T \mathbb{I}_{S_t=s,S_{t+1}=|\mathcal{S}|}]$ and $\mathbf{N}_{s,a} = [\sum_{t=1}^T \mathbb{I}_{S_t=s,A_t=a,S_{t+1}=1}, \sum_{t=1}^T \mathbb{I}_{S_t=s,A_t=a,S_{t+1}=2}, ..., \sum_{t=1}^T \mathbb{I}_{S_t=s,A_t=a,S_{t+1}=|\mathcal{S}|}]$.

$$= \frac{P(H_0)}{B(\boldsymbol{\alpha})^{|\mathcal{S}|}} \prod_{s\in\mathcal{S}} B\big(\boldsymbol{\alpha}+\mathbf{N}_s\big)\text{Dirichlet}\big(\boldsymbol{\varphi}_s;\boldsymbol{\alpha}+\mathbf{N}_s\big)$$
$$+ \frac{P(H_1)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}} \prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}} B\big(\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)\text{Dirichlet}\big(\boldsymbol{\varphi}_{s,a};\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)$$

Thus,

$$X = \frac{P(H_0)B(\boldsymbol{\alpha})^{|\mathcal{S}|(|\mathcal{A}|-1)}\prod_{s\in\mathcal{S}} B\big(\boldsymbol{\alpha}+\mathbf{N}_s\big)\text{Dirichlet}\big(\boldsymbol{\varphi}_s;\boldsymbol{\alpha}+\mathbf{N}_s\big)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}$$
$$+ \frac{P(H_1)\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}} B\big(\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)\text{Dirichlet}\big(\boldsymbol{\varphi}_{s,a};\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}$$

Since $X = P(\mathcal{H}_T|\Theta)P(\Theta)$ and $Y = \int P(\mathcal{H}_T|\Theta)P(\Theta)d\Theta$, we have that

$$Y = \frac{P(H_0)}{B(\boldsymbol{\alpha})^{|\mathcal{S}|}} \prod_{s\in\mathcal{S}} B\big(\boldsymbol{\alpha}+\mathbf{N}_s\big) + \frac{P(H_1)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}} \prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}} B\big(\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)$$
$$= \frac{P(H_0)B(\boldsymbol{\alpha})^{|\mathcal{S}|(|\mathcal{A}|-1)}\prod_{s\in\mathcal{S}} B\big(\boldsymbol{\alpha}+\mathbf{N}_s\big) + P(H_1)\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}} B\big(\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}} =: \frac{W_0+W_1}{B(\boldsymbol{\alpha})^{|\mathcal{S}||\mathcal{A}|}}$$

Thus,

$$P(\Theta|\mathcal{H}_T) = \frac{X}{Y} = \frac{W_0}{W_0+W_1}\prod_{s\in\mathcal{S}}\text{Dirichlet}\big(\boldsymbol{\varphi}_s;\boldsymbol{\alpha}+\mathbf{N}_s\big) + \frac{W_0}{W_0+W_1}\prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\text{Dirichlet}\big(\boldsymbol{\varphi}_{s,a};\boldsymbol{\alpha}+\mathbf{N}_{s,a}\big)$$

Note that

$$P(H_0|\mathcal{H}_T) = \frac{P(H_0|\mathcal{H}_T)P(\mathcal{H}_T)}{P(\mathcal{H}_T)} = \frac{P(\mathcal{H}_T|H_0)P(H_0)}{P(\mathcal{H}_T)} = \frac{W_0}{W_0+W_1}$$
$$P(H_1|\mathcal{H}_T) = \frac{W_1}{W_0+W_1}$$