Why is Posterior Sampling Better than Optimism for Reinforcement Learning?

Ian Osband
Stanford University, Google DeepMind
iosband@stanford.edu

Benjamin Van Roy
Stanford University
bvr@stanford.edu

Abstract

Computational results demonstrate that posterior sampling for reinforcement learning (PSRL) dramatically outperforms algorithms driven by optimism, such as UCRL2. We provide insight into the extent of this performance boost and the phenomenon that drives it. We leverage this insight to establish an $\tilde{O}(H\sqrt{SAT})$ expected regret bound for PSRL in finite-horizon episodic Markov decision processes, where $H$ is the horizon, $S$ is the number of states, $A$ is the number of actions and $T$ is the time elapsed. This improves upon the best previous bound of $\tilde{O}(HS\sqrt{AT})$ for any reinforcement learning algorithm.

1 Introduction

We consider a well-studied reinforcement learning problem in which an agent interacts with a Markov decision process with the aim of maximizing expected cumulative reward [25, 6]. Our focus is on the tabula rasa case, in which the agent has virtually no prior information about the MDP. As such, the agent is unable to generalize across state-action pairs and may have to gather data at each in order to learn an effective decision policy. Key to performance is how the agent balances between exploration to acquire information of long-term benefit and exploitation to maximize expected near-term rewards.

In principle, dynamic programming can be applied to compute the so-called Bayes-optimal solution to this problem [12]. However, this is computationally intractable for anything beyond the simplest of toy problems. As such, researchers have proposed and analyzed a number of heuristic reinforcement learning algorithms.

The literature on efficient reinforcement learning offers statistical efficiency guarantees for computationally tractable algorithms. These provably efficient algorithms (see, e.g., [10, 11, 13, 23, 20, 14, 8]) predominantly address the exploration-exploitation trade-off via optimism in the face of uncertainty (OFU): when at a state, the agent assigns to each action an optimistically biased while statistically plausible estimate of future value and selects the action with the greatest estimate. If a selected action is not near-optimal, the estimate must be overly optimistic, in which case the agent learns from the experience. Efficiency relative to less sophisticated exploration strategies stems from the fact that the agent avoids actions that neither yield high value nor informative data.

An alternative approach, based on Thompson sampling [27], involves sampling a statistically plausibly set of action values and selecting the maximizing action. These values can be generated, for example, by sampling from the posterior distribution over MDPs and computing the state-action value function of the sampled MDP. This approach, originally proposed in [24], is called posterior sampling for reinforcement learning (PSRL). Computational results in [20] demonstrate that PSRL dramatically outperforms algorithms based on OFU. The primary aim of this paper is to provide insight into the extent of this performance boost and the phenomenon that drives it.
We consider the problem of learning to optimize a random finite-horizon MDP $M = (S, A, R, P, H, \rho)$ over episodes of interaction, where $S = \{1,...,S\}$ is the state space, $A = \{1,...,A\}$ is the action space, $H$ is the horizon, and $\rho$ is the initial state distribution. In each time period $h = 1,...,H$ within an episode, the agent observes state $s_h \in S$, selects action $a_h \in A$, receives a reward $r_h \sim R^* (s_h, a_h)$, and transitions to a new state $s_{h+1} \sim P^* (s_h, a_h)$.

A policy $\mu$ is a mapping from state $s \in S$ and period $h = 1,...,H$ to action $a \in A$. For each MDP $M = (S, A, R, P, H, \rho)$ and policy $\mu$ we define the state-action value function for each period $h$:

$$Q_{\mu, h}^M (s, a) := E_{M, \mu} \left[ \sum_{j=h}^{H} r_j \mid s_j = s, a_j = a \right],$$  \hspace{1cm} (1)

where $r_j \sim R^* (s_j, a_j)$. The subscript $\mu$ indicates that actions over periods $h + 1, ..., H$ are selected according to the policy $\mu$. Let $V_{\mu, h}^M (s) := Q_{\mu, h}^M (s, \mu (s, h))$. A policy $\mu^M$ is optimal for the MDP $M$ if $\mu^M \in \text{arg max}_\mu V_{\mu, h}^M (s)$ for all $s \in S$ and $h = 1, ..., H$. We will use $\mu^M$ to denote such an optimal policy.

Let $\mathcal{H}_t = (s_1, a_1, r_1, ..., s_{t-1}, a_{t-1}, r_{t-1})$ denote the history of observations made prior to time $t$. To highlight this time evolution within episodes, with some abuse of notation, we let $s_{kh} = s_t$ for $t = (k-1)H + h$, so that $s_{kh}$ is the state in period $h$ of episode $k$. We define $\mathcal{H}_{kh}$ analogously. An RL algorithm is a deterministic sequence $\{\pi_k \mid k = 1, 2, ...\}$ of functions, each mapping $\mathcal{H}_{k1}$ to a probability distribution $\pi_k (\mathcal{H}_{k1})$ over policies, from which the agent samples a policy $\mu_k$ for use over the $k$th episode. We define the regret incurred by a reinforcement learning algorithm $\pi$ up to time $T$ to be

$$\text{Regret}(T, \pi, M^*) := \sum_{k=1}^{\lfloor T/H \rfloor} \Delta_k,$$  \hspace{1cm} (2)

where $\Delta_k$ denotes regret over the $k$th episode, defined with respect to true MDP $M^*$ by

$$\Delta_k := \sum_{s} \rho(s) (V_{\mu^*_1}^{M^*} (s) - V_{\mu_k, 1}^{M^*} (s))$$  \hspace{1cm} (3)

with $\mu^* = \mu^M$. Note that regret is random since it depends on the random MDP $M^*$, on the random sampling of policies, and, through the history $\mathcal{H}_{k1}$, on previous transitions and rewards. We will assess and compare algorithm performance in terms of expected regret.

In a longer paper, we would like to provide some more detailed discussion between the similarities and differences between this formulation and other common problems in the literature. This would include some discussion of finite vs infinite horizon problems [4, 3], regret vs sample complexity [15] and Bayesian vs frequentist guarantees [17]. However, this is outside the scope of this paper.

2 Problem formulation
3 Why are optimistic algorithms inefficient?

Algorithm 1 conveys the typical structure of an RL algorithm based on OFU. Before each episode, it constructs a confidence set to represent the range of MDPs that are statistically plausible given prior knowledge and observations. Then, a policy is selected by maximizing value simultaneously over policies and MDPs in this set. The agent then follows this policy over the episode. It is interesting to contrast this approach against PSRL, which is presented as Algorithm 2. Instead of maximizing over a confidence set, PSRL samples a single statistically plausible MDP and selects a policy that maximizes value for that MDP.

Algorithm 1 OFU RL

1: Input: confidence set constructor $\Phi$
2: for episode $k=1,2,...$ do
3: construct confidence set $M_k = \Phi(\mathcal{H}_k)$
4: compute $\mu_k \in \text{argmax}_{\mu} \max_{M} V_{\mu}^{M_k}$
5: for time $h=1,2,...,H$ do
6: take action $a_{kh} = \mu_k(s_{kh}, h)$
7: observe $r_{kh}$ and $s_{kh+1}$
8: update $H_{kk} = H_{kk} \cup (a_{kh}, r_{kh}, s_{kh+1})$
9: end for
10: end for

Algorithm 2 PSRL

1: Input: prior distribution $\phi$
2: for episode $k=1,2,...$ do
3: sample MDP $M_k \sim \phi(\cdot|\mathcal{H}_k)$
4: compute $\mu_k \in \text{argmax}_{\mu} V_{\mu}^{M_k}$
5: for time $h=1,2,...,H$ do
6: take action $a_{kh} = \mu_k(s_{kh}, h)$
7: observe $r_{kh}$ and $s_{kh+1}$
8: update $H_{kk} = H_{kk} \cup (a_{kh}, r_{kh}, s_{kh+1})$
9: end for
10: end for

The literature on efficient reinforcement learning offers statistical efficiency guarantees for a number of OFU-based algorithms [16 5 13 23 26 14 8]. It is worth noting that some of these algorithms do not explicitly construct an MDP confidence set but instead directly generate optimistic MDP value functions, which can be thought of as value functions for MDPs that maximizes values over an implicit confidence sets. In this section, we discuss how and why these algorithms forgo the level of statistical efficiency enjoyed by PSRL.

3.1 Confidence sets and statistical efficiency

To understand when and why optimistic algorithms sacrifice statistical efficiency, it is useful to review an example along the lines of one discussed in [22]. Our example involves a maximal-reward path problem and offers a simple context that facilitates illustration of a basic phenomenon. Just like the RL problem, this maximal-reward path problem involves learning to solve a random MDP $M^* = (S, A, R^*, P^*, H, \rho)$. Let $S$ be deterministic with $H = A = S$. The transitions $P^M(s,a)$ assign probability one to state $a$ so that the action dictates the state transition. The mean rewards are distributed $\pi^M(s,a) \sim \mathcal{N}(1{\{s \neq a\}},1)$ independently across state-action pairs, and $R^*(s,a)$ assign probability one to $\pi^M(s,a)$. The initial distribution $\rho$ assign probability one to $s = 1$. This environment can be thought of as a graph with $S$ densely-connected vertices and edges that offer normally distributed rewards. As with our reinforcement learning formulation, we consider learning to solve this problem over repeated episodes, but we introduce one difference. Instead of observing a sample from $R^*(s_{kh}, a_{kh})$ when action $a_{kh}$ is selected at state $s_{kh}$, the agent only observes a noise-corrupted version of the cumulative reward in an episode. In particular, at the end of the $k$th episode, the agent observes $\sum_{h=1}^{H} \tilde{\pi}^M(s_{kh}, a_{kh}) + w_k$, where $w_k \sim \mathcal{N}(0,1)$.

Proposition 3 of [22], translated to our current context, establishes that for our maximal-reward path problem, the expected regret of PSRL satisfies

$$E[\text{Regret}(T, \pi^{\text{PSRL}}, M^*)] = \tilde{O}(\sqrt{H SAT}).$$

One can design an OFU-based algorithm that obeys the same bound, as established in [4]. Such an algorithm would, before each episode, construct an ellipsoidal confidence set characterized by a level set of the posterior distribution over mean rewards, which is a normal distribution over $S \times A$ variables. Then, it would select the policy and combination of mean rewards from this set to maximize expected reward over the next episode. Such an algorithm would enjoy expected regret competitive with that of PSRL but, as observed in [7], it requires solving an NP-hard optimization problem at the start of each episode. PSRL, on the other hand, is computationally efficient since it only involves solving a single known MDP.
An alternative and computationally efficient OFU algorithm generates an independent confidence interval for each edge’s mean reward and combines these to give hyper-rectangular confidence sets. Using these confidence sets, you can establish a regret bound of the form
\[
E\left[\text{Regret}(T, \pi^{\text{OFUrect}}, M^*)\right] = \tilde{O}(HS\sqrt{AT}),
\] (5)
which sacrifices a factor of \(\sqrt{HS}\) relative to ellipsoidal bounds in (4). The reason for this loss in statistical efficiency is intuitive, as we now explain. A hyper-rectangular confidence set supplies a separate confidence interval for each edge’s mean reward. Each interval has to be wide enough to include all statistically plausible values. When these intervals are combined, the resulting hyper-rectangle will contain combinations of edge rewards that are not statistically plausible. In particular, though having one of the mean rewards at the boundary of its confidence interval may be statistically plausible, having mean rewards of all edges simultaneously at their respective boundaries should not be. As such, hyper-rectangular confidence sets force OFU-based algorithms to be overly conservative in ruling out combinations of mean rewards, which impedes learning.

A similar issue arises in building confidence sets of MDPs for general reinforcement learning. To facilitate computational tractability, OFU-based reinforcement learning algorithms maintain separate confidence sets for uncertainties associated with separate state-action pairs. These are sometimes confidence sets of transition probabilities and mean rewards and are sometimes confidence sets of state-action values. In either case, the independence of these confidence sets introduces statistical inefficiencies akin to those we discussed in the context of our maximal-reward path problem. A more statistically efficient OFU-based algorithm would construct confidence sets that correspond to level sets of a posterior distribution over MDPs. This would, however, likely give rise to computationally intractable optimization problems since the confidence bounds are coupled across each state and action through time. PSRL naturally samples from the properly calibrated posterior distribution which enables its relative statistical efficiency, but at a computational cost no greater than solving a single known MDP. By comparison even statistically loose hyper-rectangular approximations such as UCRL2 are \(S\)-times more computationally expensive [14].

3.2 Computational illustration

In this section, we present some computational illustration that the confidence sets used in common OFU-based algorithms lead to extremely inefficient estimates as \(S\) or \(H\) grow. We highlight these failures via some trivial MDPs with only one action which we describe in Figure 1. In each of these environments the true value from the initial state \(0\) is \(Q(0, 1) = \frac{1}{2}(H - 1)\). The simplicity of the examples means it is easy to see how to design more effective confidence sets in these contexts. Nonetheless, we will see that confidence sets suggested by OFU-RL can become incredibly mis-calibrated as \(H\) and \(S\) grow.

3.2 Computational illustration

In this section, we present some computational illustration that the confidence sets used in common OFU-based algorithms lead to extremely inefficient estimates as \(S\) or \(H\) grow. We highlight these failures via some trivial MDPs with only one action which we describe in Figure 1. In each of these environments the true value from the initial state \(0\) is \(Q(0, 1) = \frac{1}{2}(H - 1)\). The simplicity of the examples means it is easy to see how to design more effective confidence sets in these contexts. Nonetheless, we will see that confidence sets suggested by OFU-RL can become incredibly mis-calibrated as \(H\) and \(S\) grow.

![Figure 1: Simple environments designed to highlight potential problems with OFU-RL.](image)

In each experiment we sample \(K = 1000\) episodes of data from the MDP and then examine the optimistic/sampled Q-values for UCRL2 and PSRL. We implement a version of UCRL2 optimized for finite horizon MDPs and implement PSRL with a uniform Dirichlet prior over the initial dynamics \(P(0, 1) = (p_1, \ldots, p_{2N})\) and a \(N(0, 1)\) prior over rewards updating as if rewards had \(N(0, 1)\) noise. For both algorithms, if we say that \(R\) or \(P\) are known then we mean that we use the true \(R\) or \(P\) inside this part of UCRL2 or PSRL. We present three simple experiments. Figure 2a shows the effect of increasing \(N\) in Figure 1a on an agent with full reward knowledge but unknown transitions. Figure 2b shows the effect of increasing \(N\) in Figure 1a on an agent with full transition knowledge but unknown rewards.
Figure 2c shows the effect of increasing $H$ in Figure 1b on an agent with unknown rewards and transitions. We compare the estimates of PSRL to a version of UCRL2 modified for finite horizon settings. In each experiment, the estimates guided by OFU become extremely mis-calibrated, even for relatively small values of $S, H$. PSRL remains stable.

(a) $R$ known, $P$ unknown.  
  Vary $N$ in Figure 1a.  

(b) $P$ known, $R$ unknown.  
  Vary $N$ in Figure 1b.  

(c) $R$, $P$ unknown.  
  Vary $H$ in Figure 1b.  

Figure 2: UCRL2 leads to miscalibrated optimism, PSRL does not.

The results of Figure 2b are particularly revealing. They demonstrate the potential pitfalls of OFU-RL even when the underlying transition dynamics are entirely known. Several OFU algorithms have been proposed to remedy the loose UCRL-style L1 concentration from transitions [9, 1, 8] but none of these address the inefficiency from hyper-rectangular confidence sets. These loose confidence sets lead to extremely poor performance. We push these results to Appendix A along with comparison to several other OFU approaches.

4 Posterior sampling for reinforcement learning

We believe that PSRL offers comparable performance to the computationally intractable OFU-RL algorithm that uses appropriately designed confidence sets to overcome shortcomings discussed in the previous section. Importantly, PSRL has computational cost no greater than solving a single known MDP. In this section, we develop insight into how PSRL accomplishes this and the extent of the resulting performance improvement. This improvement can be formalized through regret bounds. For the context of a finite horizon episodic MDP, prior analyses of UCRL2 [14] and PSRL [20] lead to an expected regret bound:

$$E[\text{Regret}(T, \pi, M^*)] = \tilde{O}(HS \sqrt{AT}).$$

In fact, we believe that PSRL satisfies a tighter regret bound:

$$E[\text{Regret}(T, \pi, M^*)] = \tilde{O}(\sqrt{HSAT}).$$

The next subsections shed light on the reduction of $S$ to $\sqrt{S}$ and the reduction of $H$ to $\sqrt{H}$.

4.1 From $S$ to $\sqrt{S}$

The following result establishes that PSRL satisfies a stronger expected regret bound than previous analysis [20], reducing the scaling with $S$ to $\sqrt{S}$.

**Theorem 1.** For any prior over rewards with $\overline{r} \in [0, 1]$ and sub Gaussian noise and any independent Dirichlet prior over transitions, PSRL satisfies

$$E[\text{Regret}(T, \pi^{PSRL}, M^*)] = \tilde{O}\left(\frac{H \sqrt{SA}}{T}\right)$$

Our proof of Theorem 1 begins with the standard analysis of optimistic algorithms to add and subtract the *imagined* optimal reward $V^{M_k}_{\mu_k, 1}$.

$$\Delta_k = V^{M^*}_{\mu^*, 1} - V^{M^*}_{\mu_k, 1}$$

$$= \frac{V^{M^*}_{\mu_k, 1} - V_{\mu_k, 1}}{\Delta_k^{\text{true}}} + \frac{V_{\mu_k, 1} - V^{M^*}_{\mu_k, 1}}{\Delta_k^{\text{true}}}$$

(7)
Where $\Delta^\text{opt}_k$ is the regret from optimism and $\Delta^\text{conc}_k$ is the regret from concentration. We then use posterior sampling to assert that, conditional on any past data \[22, 20\]

$$E[\Delta^\text{opt}_k|H_{k1}] = E[V^*_1 - V^k_{1}|H_{k1}] = 0.$$ 

To condense our notation we write $x_{kh} := (s_{kh}, a_{kh})$ and $V^k_{k,h} := V^M_{\mu_{k,h}}$. We let $\hat{r}_k(x) := E[r^*(x)|H_{k1}]$, $\hat{P}_k(x) := E[P^*(x)|H_{k1}]$ and introduce $w^R(x) := r_k(x) - \hat{r}_k(x)$ and $w^P_h := (P_k(x_{k1}) - \hat{P}_k(x_{k1}))^T V^k_{k,h} + 1$.

We note that, conditional upon the data $H_{k1}$ the true reward and transitions are independent of the rewards and transitions sampled by PSRL, so that $E[r^*(x)|H_{k1}] = \hat{r}_k(x)$, $E[P^*(x)|H_{k1}] = \hat{P}_k(x)$ for any $x$. However, $E[w^R(x)|H_{k1}]$ and $E[w^P_h(x)|H_{k1}]$ are generally non-zero, since the agent chooses its policy to optimize its reward under $M_k$. We can rewrite the regret from concentration via the Bellman operator (section 5.2 of \[20\]),

$$E[V^k_{1}|H_{k1}] = E[\tau_k(x_{k1}) - r(x_{k1}) + P_k(x_{k1})^T V^k_{2} - P^*(x_{k1})^T V^k_{2} | H_{k1}]$$

$$= E[\tau_k(x_{k1}) - r(x_{k1}) + P_k(x_{k1})^T V^k_{2} + \{V^k_{2} - V^k_{2}(s')|s' \sim P^*(x_{k1})\} | H_{k1}]$$

$$= \ldots$$

$$= E\left[\sum^H_{h=1}\{\tau_k(x_{k1}) - r^*(x_{k1})\} + \sum^H_{h=1}\{P_k(x_{k1}) - \hat{P}_k(x_{k1})\}^T V^k_{h}\right] | H_{k1}$$

$$\leq E\left[\sum^H_{h=1}|w^R(x_{kh})| + \sum^H_{h=1}|w^P_h(x_{kh})|\right] | H_{k1}]. \quad (8)$$

Bounding the contribution from $|w^R|$ is relatively straightforward. The key step to our proof is a new result that bounds $|w^P_h|$ in terms of a Gaussian posterior, even though the underlying posterior distribution is not Gaussian. At the root of our argument is the notion of stochastic optimism \[21\], which introduces a partial ordering over random variables.

**Definition 1.** For any $X$ and $Y$ real-valued random variables we say that $X$ is stochastically optimistic for $Y$ if for any $u: \mathbb{R} \rightarrow \mathbb{R}$ convex and increasing $E[u(X)] \geq E[u(Y)]$. We will use the notation $X \succeq_{st} Y$ to express this relation.

At a high level, we establish that posterior sampling with a correct prior can be approximated by posterior sampling with an incorrect prior if the mis-specified prior and all possible posteriors are stochastically optimistic the corresponding true prior and posteriors. This allows us to approximate PSRL with a Dirichlet prior over transition probabilities via a Gaussian prior over probability-weighted next-state values, based on the following result \[21\]:

**Lemma 1.** For all $V \in [0, 1]^N$ and $\alpha \in [0, \infty)^N$ with $\alpha^T 1 \geq 2$, if $X \sim N(\alpha^T V/\alpha^T 1, 1/\alpha^T 1)$ and $Y = P^T V$ for $P \sim \text{Dirichlet}(\alpha)$ then $X \succeq_{st} Y$.

It turns out that analysis of a conceptual algorithm that makes use of this Gaussian approximation rather than Dirichlet distributions is simpler, and this allows us to establish the regret bound of Theorem 1. We call this algorithm Gaussian PSRL and describe the algorithm in Appendix \[C\]. This result allows us to prove that $|w^P_h(x)|$ concentrates at a rate $\tilde{O}(H/\sqrt{n(x)})$, rather than previous analysis $\tilde{O}(H\sqrt{S/n(x)})$ where $n(x)$ is the number of prior observations at $x$.

Prior work has explored other ways designing OFU-based algorithms that reduce how regret or learning time scales with $S$. MORMAX \[26\] and delayed Q-learning \[23\], in particular, come with sample complexity bounds that are linear in $S$, which may be the best dependence on $S$ that one can hope for. But even in terms of sample complexity, these algorithms are not necessarily an improvement over UCRL2 or its variants with similar bounds \[8\]. We can compare these bounds in terms of $T^\pi(\epsilon) := \text{"time until algorithm } \pi \text{ has expected average regret less than } \epsilon$:

$$T^{\text{DelayQ}}(\epsilon) = \tilde{O}\left(\frac{H^2 S A}{\epsilon^2}\right), \quad T^{\text{MORMAX}}(\epsilon) = \tilde{O}\left(\frac{H^2 S A}{\epsilon^2}\right), \quad T^{\text{UCRL2}}(\epsilon) = \tilde{O}\left(\frac{H^2 S^2 A}{\epsilon^2}\right). \quad (9)$$

Theorem 1 implies $T^{\text{PSRL}}(\epsilon) = \tilde{O}(H^2 S A)$. MORMAX and delayed Q-learning reduces the $S$-dependence of UCRL2, but this comes at the expense of worse dependence on $H$, and the resulting algorithms are not practical.
4.2 From $H$ to $\sqrt{H}$

The results of [4] adapted to finite horizon MDPs would suggest a lower bound $\Omega(H\sqrt{SAT})$ on the worst-case regret for any algorithm. However, the associated proof is incorrect. The strongest lower bound with a correct proof is $\Omega(\sqrt{HSAT})$ [14]. It remains an open question whether such a lower bound applies to expected regret, though we believe it does.

Recent analyses [18, 8] suggest that reducing the dependence of $H$ to $\sqrt{H}$ is possible. They note that “local value variance” satisfies a Bellman equation. Intuitively this captures that if we transition to a bad state $V \simeq 0$, then we cannot transition anywhere much worse during this episode. This relation means that $\sum_{h=1}^{H} w(R)(x_{kh})$ should behave more as if they were independent and grow $O(\sqrt{H})$, unlike our analysis which crudely upper bounds them each in turn $O(H)$. We present a sketch towards an analysis of Conjecture 1 in Appendix D.

**Conjecture 1.** For any prior over rewards with $r \in [0, 1]$ and sub Gaussian noise and any independent Dirichlet prior over transitions, we conjecture that

$$E[\text{Regret}(T, \pi^{\text{PSRL}}, M^*)] = \tilde{O}(\sqrt{HSAT}),$$

and that this matches the lower bounds for any algorithm up to logarithmic factors.

We note that, unlike our proof of Theorem 1 these arguments would not hold for Gaussian PSRL. Interestingly, we find that our experimental evaluation is consistent with $\tilde{O}(HS\sqrt{AT})$, $O(H\sqrt{SAT})$ and $O(\sqrt{HSAT})$ for UCRL2, Gaussian PSRL and PSRL respectively. These results help to suggest that it might be possible to formalize the arguments in our conjecture and maybe even in terms of the problem-specific frequentist regret [11].

4.2.1 An empirical investigation

We now discuss a computational study designed to illustrate how learning times of reinforcement learning algorithms scale with $H$, and in particular, to support the conjecture that PSRL reduces regret by a factor of $\sqrt{H}$ relative to UCRL2. The class of MDPs we consider involves a long chain of states with $S = H = N$ and with two actions: left and right. Each episode the agent begins in state 1. The optimal policy is to head right at every timestep, all other policies have zero expected reward. Inefficient strategies for exploration will take $\Omega(2^N)$ episodes to learn the optimal policy [21].

![Figure 3: A family of MDPs designed to highlight the need for efficient exploration.](image)

We evaluate several learning algorithms from five random seeds and $N = 2, \ldots, 100$ for one million episodes each. Our goal is to investigate their empirical performance and scaling. We highlight results for three algorithms with $O(\sqrt{T})$ regret bounds: UCRL2, Gaussian PSRL and PSRL. We include more details for these experiments including a link to their source code and experimental results for several other algorithms in Appendix E. We implement UCRL2 with confidence sets optimized for finite horizon MDPs. For the Bayesian algorithms we use a uniform Dirichlet prior for transitions and $N(0, 1)$ prior for rewards.

Figure 4 display the regret curves for these algorithms for $N \in \{5, 10, 30, 50\}$. As suggested by our analysis, PSRL outperforms Gaussian PSRL which itself outperforms UCRL2. These differences seems to scale with the length of the chain $N$ and that even for relatively small MDPs, PSRL is many orders of magnitude more efficient than UCRL2.
Figure 4: PSRL outperforms other methods by large margins.

We investigate the empirical scaling of these algorithms with respect to $N$ and compare these results to our analysis. Suppose for some $B \in \mathbb{R}$ we can bound $\text{Regret}(T) \leq \sqrt{BT}$. This means we that $\tilde{T}(\epsilon) := \# \text{episodes until the average regret per episode} \leq \epsilon = O(B^2 H/\epsilon^2)$. The results of Theorem 1 and Conjecture 1 would suggest that up to $o(\log(N))$:

$$\log(\tilde{T}_{UCRL2}) \simeq 5 \log(N), \quad \log(\tilde{T}_{GaussPSRL}) \simeq 4 \log(N), \quad \log(\tilde{T}_{PSRL}) \simeq 3 \log(N).$$

Figure 5 displays $\tilde{T}(0.1)$ for UCRL2, Gaussian PSRL and PSRL. Along side these results we also draw dotted lines with slope 5, 4 and 3 respectively, which are the scalings associated with bounds $O(HS \sqrt{AT})$, $O(H \sqrt{SAT})$ and $O(\sqrt{HSAT})$ respectively. We find that the empirical results match the results of Theorem 1 and provide support to Conjecture 1.

5 Conclusion

PSRL is orders of magnitude more statistically efficient than UCRL and $S$-times less computationally expensive. In the future, we believe that analysts will be able to formally specify an OFU approach to RL whose statistical efficiency matches PSRL. However, we argue that the resulting confidence sets which address both the coupling over $H$ and $S$ will result in a computationally intractable optimization problem. For this reason, computationally efficient approaches to OFU RL will sacrifice statistical efficiency; this is why posterior sampling is better than optimism for reinforcement learning.

References

[1] Mauricio Araya, Olivier Buffet, and Vincent Thomas. Near-optimal brl using optimistic local transitions. arXiv preprint arXiv:1206.4613, 2012.

[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] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 89–96. Curran Associates, Inc., 2009.

[4] Peter L. Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, pages 35–42, June 2009.

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

[6] Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for Markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997.

[7] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366, 2008.

[8] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, page TBA, 2015.

[9] Sarah Filippi, Olivier Cappé, and Aurélien Garivier. Optimism in reinforcement learning and kullback-leibler divergence. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 115–122. IEEE, 2010.

[10] Raphaël Fonteneau, Nathan Korda, and Rémi Munos. An optimistic posterior sampling strategy for Bayesian reinforcement learning. In *NIPS 2013 Workshop on Bayesian Optimization (BayesOpt2013)*, 2013.

[11] Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized Markov decision processes. *arXiv preprint arXiv:1406.7498*, 2014.

[12] Arthur Guez, David Silver, and Peter Dayan. Better optimism by bayes: Adaptive planning with rich models. *arXiv preprint arXiv:1402.1958*, 2014.

[13] Josef Hadar and William R Russell. Rules for ordering uncertain prospects. *The American Economic Review*, pages 25–34, 1969.

[14] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

[15] Sham Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

[16] Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

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

[18] Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In *Algorithmic learning theory*, pages 320–334. Springer, 2012.

[19] Haim Levy. Stochastic dominance and expected utility: survey and analysis. *Management Science*, 38(4):555–593, 1992.

[20] Ian Osband, Daniel Russo, and Benjamin Van Roy. (More) efficient reinforcement learning via posterior sampling. In *NIPS*, pages 3003–3011. Curran Associates, Inc., 2013.

[21] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

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

[23] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *ICML*, pages 881–888, 2006.

[24] Malcolm J. A. Strens. A Bayesian framework for reinforcement learning. In *ICML*, pages 943–950, 2000.

[25] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, March 1998.

[26] István Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1031–1038, 2010.

[27] 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.
APPENDICES

A Computational illustration

In this section we expand upon the simple examples given by Section 3.2 to a full decision problem with two actions. We define an MDP similar to Figure 1 but now with two actions. The first action is identical to Figure 1, but the second action modifies the transition probabilities to favour the rewarding states with probability $0.6/N$ and assigning only $0.4/N$ to the non-rewarding states.

We now investigate the regret of several learning algorithms which we adapt to this setting. These algorithms are based upon BEB [17], BOLT [1], $\epsilon$-greedy with $\epsilon = 0.1$, Gaussian PSRL (see Appendix C), Optimistic PSRL (which takes $K = 10$ samples and takes the maximum over sampled Q-values similar to BOSS [2]), PSRL [24], UCFH [8] and UCRL2 [14]. We link to the full code for implementation in Appendix E.

Figure 6: Known rewards $R$ and unknown transitions $P$, similar to Figure 2a.

Figure 7: Unknown rewards $R$ and known transitions $P$, similar to Figure 2b.

We see that the loose estimates in OFU algorithms from Figure 2 lead to bad performance in a decision problem. This poor scaling with the number of successor states $N$ occurs when either the rewards or the transition function is unknown. We note that in stochastic environments the PAC-Bayes algorithm BOLT, which relies upon optimistic fake prior data, can sometimes concentrate too quickly and so incur the maximum linear regret. In general, although BOLT is PAC-Bayes, it concentrates too fast to be PAC-MDP just like BEB [17].

In Figure 8 we see a similar effect as we increase the episode length $H$. We note the second order UCFH modification improves upon UCRL2’s miscalibration with $H$, as is reflected in their bounds [8]. We note that both BEB and BOLT scale poorly with the horizon $H$. 

10
B Proof of Theorem 1

The contributions from rewards \( w_R(x_{kh}) = \tilde{O}(\sqrt{1/n}) \) are relatively simple to bound for any posterior distribution \([20]\). The key piece of our analysis is modified form of Lemma 1 to bound the contribution from transitions \( w_P(x_{kh}) \) under any future value function \( V_{kh+1} \).

Before we can approach this result we need to look more deeply into the notion of stochastic optimism.

Stochastic optimism is closely related to second-order stochastic dominance:
\[ X \succeq_{so} Y \text{ if and only if } -Y \text{ second-order stochastically dominates } -X \,[13] \]

This characterization of optimism for random variables affords us several equivalences via integration by parts \([19]\).

**Lemma 2.** The following are equivalent to \( X \succeq_{so} Y \):

1. For any random variable \( Z \) independent of \( X \) and \( Y \),
   \[ E[\max(X, Z)] \geq E[\max(Y, Z)] \]
2. \( X = Y + A + W \) for \( A \geq 0 \) and \( E[W|Y+A] = 0 \) for all values \( y + a \).
3. For any \( \alpha \in \mathbb{R} \),
   \[ \int_{\alpha}^{\infty} \{P(X \geq s) - P(Y \geq s)\} \, ds \geq 0 \]

Lemma 1 implies that the maximum of optimistic variables is optimistic, 3.2 provides some more clear intuition for this condition and previous work \([21]\) uses 3.3 to establish Lemma 1. At first glance Lemma 1 may look obstructive and the full proof \([21]\) which we reproduce in Appendix. Nevertheless, the ideas behind this argument are simple and the result is quite deep. Lemma 1 compares the posterior distribution of the Q-values for a problem similar Figure 1 when updating \( x_{kh} \) under a Gaussian posterior for the rewards as if the \( N(\mu, 1) \) \( X \) and the PSRL update for Dirichlet transitions \( Y \). Lemma 1 says that, for any possible data, the resulting posteriors will be ordered \( X \succeq_{so} Y \).

Lemma 1 will allow us to bound the concentration of the next step value function \( PTV \) directly, rather than via some Hölder decomposition as per Section 3. This is crucial to our analysis to provide \( \tilde{O}(H\sqrt{SAT}) \) bounds upon the expected regret.

**Lemma 3.** For any history \( \mathcal{H}_{k1} \) then with probability at least \( 1 - \frac{1}{T} \)
\[
E \left[ \sum_{h=1}^{H} \left( |w_R(x_{kh})| + |w_P(x_{kh})| \right) \right| \mathcal{H}_{k1} \right] \leq \sum_{h=1}^{H} (H+1) \sqrt{\frac{2\log(2SAT)}{\max(n_h(x_{kh})-2,1)}}. \tag{11}
\]

Note the result which we claim in Lemma 3 requires a little more care because \( V_{kh+1} \) may also depend upon the sampled \( P_k(x_{kh}) \), however it poses no real extra difficulty. The key observation is that Lemma 1 already relies upon a sequence of worst-case bounds for any \( V \in [0,1]^S \). We observe that, for a finite horizon MDP, if \( w_P \) is positive then it must place more transition mass to higher value states in \( V_{kh+1} \). We use Lemma 2 to note that the setting considered in Lemma 1 is already a worst-case for future value \( V_{kh+1} \). We replace Lemma 5 in \([21]\) with a relaxation for \( V \in \{0,h\}^S \).

This, together with some elementary concentration results for sub-Gaussian random variables completes the proof of Lemma 3. We can now use Lemma 3 to complete our proof of Theorem 1.
using a pigeonhole principle,
\[
\mathbb{E} \left[ \text{Regret}(T, \pi^{\text{PSRL}}, M^*) \right] \leq \sum_{k=1}^{\lceil T/H \rceil} \sum_{h=1}^{H}(H + 1) \sqrt{\frac{2 \log(2SAT)}{n_k(x_{kh})}} + 2SA + 1 \\
\leq 5H \sqrt{SAT \log(2SAT)}.
\]

Note at no point in our proof did we require the exact posterior distribution for \(M_k\), our only requirement was that the sampled value function was stochastically optimistic \(\tilde{V}_{kl}^k \succapprox V_{*k}\).

This suggests that a similar \(\tilde{O}(H \sqrt{SAT})\) bound for a posterior sampling algorithm with a mis-specified Gaussian posterior. We call this Gaussian PSRL and describe this algorithm in Appendix (C). It should be clear that we believe Gaussian PSRL is an inferior algorithm to PSRL. However, Gaussian PSRL may be more amenable to analysis and provides some insight into what kinds of frequentist robustness is available to PSRL.

\[\square\]

C Gaussian PSRL

Our proof for PSRL actually worked by comparing the (correct) posterior estimates for PSRL with a posterior as if the underlying system were Gaussian. This is essentially identical to RLSVI with a tabular basis functions, but without the complications from the shrinkage towards zero [21]. Importantly, RLSVI in that form treats the MDP as if it were time-inhomogeneous, which leads it to have \(\mathbb{E}[\text{Regret}(T, \pi^{\text{RLSVI}}, M^*)] = \tilde{O}(\sqrt{H^3SAT})\) rather than Gaussian PSRL which is \(\mathbb{E}[\text{Regret}(T, \pi^{\text{GaussPSRL}}, M^*)] = \tilde{O}(\sqrt{H^2SAT})\).

Algorithm 3 Gaussian PSRL

1: Input: Reward prior \(\phi\) and Dirichlet transition prior counts \(\alpha_0(s,a) \in \mathbb{R}^+_n\) for each \((s,a)\).
2: Initialize: \(H = \emptyset\), \(\alpha(s,a) \leftarrow \alpha_0(s,a)\) for all \(s,a\)
3: for episode \(k = 1, 2, \ldots\) do
4: \(Q_{k+1}^h(s,a) \leftarrow 0\) for all \(s,a\)
5: sample \(\tau_k(s,a) \sim \phi(\cdot|H)\) for all \(s,a\)
6: sample \(w_k^h(s,a) \sim N\left(\mu = 0, \sigma^2 = \frac{H^2}{\alpha(s,a)}\right)\) for all \(s,a\)
7: for time \(h = H, H - 1, \ldots, 1\) do
8: \(V_{h+1}^k(s,a) \leftarrow \max_a Q_{h+1}^k(s,a)\) for all \(s,a\)
9: \(Q_h^k(s,a) \leftarrow \tau_k(s,a) + \frac{\alpha(s,a)}{\alpha(s,a) + 1} w_k^h(s,a)\) for all \(s,a\)
10: end for
11: for time \(h = 1, 2, \ldots, H\) do
12: take action \(a_{kh} \in \arg\max_a Q_k^h(s_{kh},a)\)
13: observe \(r_{kh}\) and \(s_{kh+1}\)
14: update reward history \(H = H \cup (s_{kh}, a_{kh}, r_{kh})\)
15: update transition \(\alpha(s_{kh}, a_{kh})[s_{kh+1}]++\)
16: end for
17: end for

The proof of these regret bounds is precisely the same as the argument we give for PSRL and Theorem (I). That said, we actually believe that PSRL will satisfy a stronger result \(\tilde{O}(\sqrt{HSA\log T})\) as presented in Conjecture (I).

D Towards a tight analysis of PSRL

The key remaining loose piece of our analysis concerns the summation \(\sum_{h=1}^H w_h^P(x_{kh})\). Our current proof of Theorem (I) bounds each \(w_h^P(x_{kh})\) independently. Each term is \(\tilde{O}(\sqrt{\frac{H}{n_k(x_{kh})}})\) and we bound the resulting sum \(\tilde{O}(H \sqrt{\frac{H^2}{n_k(x_{kh})}})\). However, this approach is very loose. In particular, to repeat our geometric intuition, we have assumed a worst-case hyper-rectangle over all timesteps \(H\) when the actual geometry should be an ellipse.

It is not possible to sequentially get the “worst-case” transitions \(O(H)\) at each and every timestep during an episode, since once your sample gets one such transition then there will
be no more future value to deplete. A very similar observation is used by recent analyses in
the sample complexity setting [18] and also finite horizon MDPs [8]. This seems to suggest
that it should be possible to combine the insights of Lemma 1 with, for example, Lemma 4
of [8] to remove both the $\sqrt{s}$ and the $\sqrt{H}$ from our bounds to prove Conjecture 1.

We note that this informal argument would not apply Gaussian PSRL, since it generates $w^P$
from some Gaussian posterior which does not satisfy the Bellman operators. Therefore, we
should be able to find some evidence for this conjecture if we find domains where UCRL,
Gaussian PSRL and PSRL all demonstrate their (unique) predicted scalings.

E Chain experiments

All of the code and experiments used in this paper are available in full at https://github
com/iosband/TabulaRL. We hope that researchers will find this simple codebase useful for
quickly prototyping and experimenting in tabular reinforcement learning simulations.

In addition to the results already presented we also investigate the scaling of similar Bayesian
learning algorithms BEB [17] and BOLT [1]. We see that neither algorithms scale as gracefully
as PSRL, although BOLT comes close. However, as observed in Appendix A, BOLT can
perform poorly in more highly stochastic environments. BOLT also requires S-times more
computational cost than PSRL or BEB. We include these algorithms in Figure 9.

Figure 9: Scaling of more learning algorithms.

E.1 Rescaling confidence sets

It is well known that provably-efficient OFU algorithms can perform poorly in practice. In
response to this observation, many practitioners suggest rescaling confidence sets to obtain
better empirical performance [26, 11, 17]. In Figure 10 we present the performance of several
algorithms with confidence sets rescaled $\in \{0.01, 0.03, 0.1, 0.3, 1\}$. We can see that rescaling
for tighter confidence sets can sometimes give better empirical performance. However, it
does not change the fundamental scaling of the algorithm. Also, for aggressive scalings some
seeds may not converge at all.

E.2 Prior sensitivities

We ran all of our Bayesian algorithms with uninformative independent priors for rewards
and transitions. For rewards, we use $\pi(s, a) \sim N(0, 1)$ and updated as if the observed noise
were Gaussian with precision $\tau = \frac{1}{\sigma^2} = 1$. For transitions, we use a uniform Dirichlet prior
$P(s, a) \sim \text{Dirichlet}(\alpha)$. In Figures 11 and 12 we examine the performance of Gaussian PSRL
and PSRL on a chain of length $N = 10$ as we vary $\tau$ and $\alpha = \alpha_0 I$. 

13
We find that the algorithms are extremely robust over several orders of magnitude. Only large values of $\tau$ (which means that the agent updates its reward prior too quickly) caused problems for some seeds in this environment. Going forward, developing a more clear frequentist analysis of these Bayesian algorithms is a direction for important future research.

### E.3 Optimistic posterior sampling

We compare our implementation of PSRL with a similar optimistic variant which samples $K \geq 1$ samples from the posterior and forms the optimistic $Q$-value over the envelope of sampled $Q$-values. This algorithm is sometimes called “optimistic posterior sampling” [10]. We experiment with this algorithm over several values of $K$ but find that the resultant algorithm performs very similarly to PSRL, but at an increased computational cost. We display this effect over several magnitudes of $K$ in Figures 13 and 14.

This algorithm “Optimistic PSRL” is spiritually very similar to BOSS [2] and previous work had suggested that $K > 1$ could lead to improved performance. We believe that an important
Figure 12: Prior sensitivity in PSRL.

Figure 13: PSRL with multiple samples is almost indistinguishable.

Figure 14: PSRL with multiple samples is almost indistinguishable.
difference is that PSRL, unlike Thompson sampling, should not resample every timestep but previous implementations had compared to this faulty benchmark [10].

F Gaussian Dirichlet dominance

These results are available in Appendix G of earlier work [21].