Generalization and Exploration via Randomized Value Functions

Ian Osband[1,2] IOSBAND@STANFORD.EDU
Benjamin Van Roy[1] BVR@STANFORD.EDU
Zheng Wen[1,3] ZWEN@ADOBE.COM
[1]Stanford University, [2]Google Deepmind, [3]Adobe Research

Abstract

We propose randomized least-squares value iteration (RLSVI) – a new reinforcement learning algorithm designed to explore and generalize efficiently via linearly parameterized value functions. We explain why versions of least-squares value iteration that use Boltzmann or ϵ-greedy exploration can be highly inefficient, and we present computational results that demonstrate dramatic efficiency gains enjoyed by RLSVI. Further, we establish an upper bound on the expected regret of RLSVI that demonstrates near-optimality in a *tabula rasa* learning context. More broadly, our results suggest that randomized value functions offer a promising approach to tackling a critical challenge in reinforcement learning: synthesizing efficient exploration and effective generalization.

1. Introduction

The design of reinforcement learning (RL) algorithms that explore intractably large state-action spaces efficiently remains an important challenge. In this paper, we propose randomized least-squares value iteration (RLSVI), which generalizes using a linearly parameterized value function. Prior RL algorithms that generalize in this way require, in the worst case, learning times exponential in the number of model parameters and/or the planning horizon. RLSVI aims to overcome these inefficiencies.

RLSVI operates in a manner similar to least-squares value iteration (LSVI) and also shares much of the spirit of other closely related approaches such as TD, LSTD, and SARSA (see, e.g., (Sutton & Barto, 1998; Szepesvári, 2010)). What fundamentally distinguishes RLSVI is that the algorithm explores through randomly sampling statistically plausible value functions, whereas the aforementioned alternatives are typically applied in conjunction with action-dithering schemes such as Boltzmann or ϵ-greedy exploration, which lead to highly inefficient learning. The concept of exploring by sampling statistically plausible value functions is broader than any specific algorithm, and beyond our proposal and study of RLSVI. We view an important role of this paper is to establish this broad concept as a promising approach to tackling a critical challenge in RL: synthesizing efficient exploration and effective generalization.

We will present computational results comparing RLSVI to LSVI with action-dithering schemes. In our case studies, these algorithms generalize using identical linearly parameterized value functions but are distinguished by how they explore. The results demonstrate that RLSVI enjoys dramatic efficiency gains. Further, we establish a bound on the expected regret for an episodic *tabula rasa* learning context, where the agent has virtually no prior information about the MDP. Our bound is $\tilde{O}(\sqrt{H^3SAT})$, where S and A denote the cardinalities of the state and action spaces, T denotes time elapsed, and H denotes the episode duration. This matches the worst case lower bound for this problem up to logarithmic factors (Jaksch et al., 2010). It is interesting to contrast this against known $\tilde{O}(\sqrt{H^3S^2AT})$ bounds for other provably efficient *tabula rasa* RL algorithms (e.g., UCRL2 (Jaksch et al., 2010)) adapted to this context. To our knowledge, our results establish RLSVI as the first RL algorithm that is provably efficient in a *tabula rasa* context and also demonstrates efficiency when generalizing via linearly parameterized value functions.

There is a sizable literature on RL algorithms that are provably efficient in *tabula rasa* contexts (Brafman & Tennenholtz, 2002; Kakade, 2003; Ortner & Ryabko, 2012; Osband et al., 2013; Strehl et al., 2006). The literature on RL algorithms that generalize and explore in a provably efficient manner is sparser. There is work on model-based RL algorithms (Abbasi-Yadkori & Szepesvári, 2011; Osband & Van Roy, 2014a;b; Gopalan & Mannor, 2014), which apply to specific model classes and become computationally intractable for problems of practical scale. Value function generalization approaches have the potential to overcome those computational challenges and offer practical means

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

for synthesizing efficient exploration and effective generalization. A relevant line of work establishes that efficient RL with value function generalization reduces to efficient KWIK online regression (Li & Littman, 2010; Li et al., 2008). However, it is not known whether the KWIK online regression problem can be solved efficiently. In terms of concrete algorithms, there is optimistic constraint propagation (OCP) (Wen & Van Roy, 2013), a provably efficient RL algorithm for exploration and value function generalization in deterministic systems, and C-PACE (Pazis & Parr, 2013), a provably efficient RL algorithm that generalizes using interpolative representations. These contributions represent important developments, but OCP is not suitable for stochastic systems and is highly sensitive to model mis-specification, and generalizing effectively in high-dimensional state spaces calls for methods that extrapolate. RLSVI advances this research agenda, leveraging randomized value functions to explore efficiently with linearly parameterized value functions. The only other work we know of involving exploration through random sampling of value functions is (Dearden et al., 1998). That work proposed an algorithm for *tabula rasa* learning; the algorithm does not generalize over the state-action space.

2. Episodic reinforcement learning

A finite-horizon MDP $\mathcal{M}=(\mathcal{S},\mathcal{A},H,P,R,\pi)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, H is the number of periods, P encodes transition probabilities, R encodes reward distributions, and π is a state distribution. In each episode, the initial state s_0 is sampled from π, and, in period $h=0,1,\cdots,H-1$, if the state is s_h and an action a_h is selected then a next state s_{h+1} is sampled from $P_h(\cdot|s_h,a_h)$ and a reward r_h is sampled from $R_h(\cdot|s_h,a_h,s_{h+1})$. The episode terminates when state s_H is reached and a terminal reward is sampled from $R_H(\cdot|s_H)$.

To represent the history of actions and observations over multiple episodes, we will often index variables by both episode and period. For example, s_{lh}, a_{lh} and r_{lh} respectively denote the state, action, and reward observed during period h in episode l. A policy $\mu = (\mu_0,\mu_1,\cdots,\mu_{H-1})$ is a sequence of functions, each mapping $\mathcal{S}$ to $\mathcal{A}$. For each policy μ, we define a value function for $h = 0,..,H$:

$$V_h^\mu(s) := \mathbb{E}_{\mathcal{M}}\left[\sum_{\tau=h}^{H} r_\tau \middle| s_h=s, a_\tau=\mu_\tau(s_\tau) \text{ for } \tau=h,..,H-1\right]$$

The optimal value function is defined by $V_h^*(s) = \sup_\mu V_h^\mu(s)$. A policy μ^* is said to be optimal if $V^{\mu^*} = V^*$. It is also useful to define a state-action optimal value function for $h = 0,..,H-1$:

$$Q_h^*(s,a) := \mathbb{E}_{\mathcal{M}}\left[r_h + V_{h+1}^*(s_{h+1}) \middle| s_h = s, a_h = a\right]$$

A policy μ^* is optimal $\iff \mu_h^*(s) \in \text{argmax}_{\alpha\in\mathcal{A}} Q_h^*(s,\alpha), \forall s,h.$

An RL algorithm generates each action a_{lh} based on observations made up to period h of episode l. Over each episode, the algorithm realizes reward $\sum_{h=0}^{H} r_{lh}$. One way to quantify the performance of an RL algorithm is in terms of the *expected cumulative regret* over L episodes, or time $T=LH$, defined by

$$\text{Regret}(T,\mathcal{M}) = \sum_{l=0}^{T/H-1} \mathbb{E}_{\mathcal{M}}\left[V_0^*(s_{l0}) - \sum_{h=0}^{H} r_{lh}\right].$$

Consider a scenario in which the agent models that, for each h, $Q_h^* \in \text{span}\,[\Phi_h]$ for some $\Phi_h \in \mathbb{R}^{\mathcal{SA}\times K}$. With some abuse of notation, we use $\mathcal{S}$ and $\mathcal{A}$ to denote the cardinalities of the state and action spaces. We refer this matrix Φ_h as a *generalization matrix* and use $\Phi_h(s,a)$ to denote the row of matrix Φ_h associated with state-action pair (s,a). For $k = 1,2,\cdots,K$, we write the kth column of Φ_h as ϕ_{hk} and refer to ϕ_{hk} as a basis function. We refer to contexts where the agent's belief is correct as *coherent learning*, and refer the alternative as *agnostic learning*.

3. The problem with dithering for exploration

LSVI can be applied at each episode to estimate the optimal value function Q^* from data gathered over previous episodes. To form an RL algorithm based on LSVI, we must specify how the agent selects actions. The most common scheme is to selectively take actions at random, we call this approach dithering. Appendix A presents RL algorithms resulting from combining LSVI with the most common schemes of ϵ-greedy or Boltzmann exploration.

The literature on efficient RL shows that these dithering schemes can lead to regret that grows exponentially in H and/or $\mathcal{S}$ (Kearns & Singh, 2002; Brafman & Tennenholtz, 2002; Kakade, 2003). Provably efficient exploration schemes in RL require that exploration is directed towards potentially informative state-action pairs and consistent over multiple timesteps. This literature provides several more intelligent exploration schemes that are provably efficient, but most only apply to *tabula rasa* RL, where little prior information is available and learning is considered efficient even if the time required scales with the cardinality of the state-action space. In a sense, RLSVI represents a synthesis of ideas from efficient *tabula rasa* reinforcement learning and value function generalization methods.

To motivate some of the benefits of RLSVI, in Figure 1 we provide a simple example that highlights the failings of dithering methods. In this setting LSVI with Boltzmann or ϵ-greedy exploration requires exponentially many episodes to learn an optimal policy, even in a coherent learning context and even with a small number of basis functions.

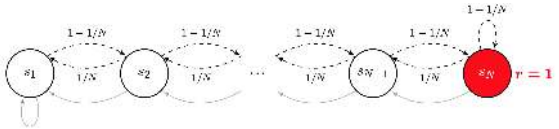


Figure 1. An MDP where dithering schemes are highly inefficient.

This environment is made up of a long chain of states $\mathcal{S} = \{1, .., N\}$. Each step the agent can transition left or right. Actions left are deterministic, but actions right only succeed with probability $1 - 1/N$, otherwise they go left. All states have zero reward except for the far right N which gives a reward of 1. Each episode is of length $H = N - 1$ and the agent will begin each episode at state 1. The optimal policy is to go right at every step to receive an expected reward of $p^* = (1-\frac{1}{N})^{N-1}$ each episode, all other policies give no reward. Example 1 establishes that, for any choice of basis function, LSVI with any ϵ-greedy or Boltzmann exploration will lead to regret that grows exponentially in $\mathcal{S}$. A similar result holds for policy gradient algorithms. x

Example 1. *Let l^* be the first episode during which state N is visited. It is easy to see that $\theta_{lh} = 0$ for all h and all $l < l^*$. Furthermore, with either ϵ-greedy or Boltzmann exploration, actions are sampled uniformly at random over episodes $l < l^*$. Thus, in any episode $l < l^*$, the red node will be reached with probability $p^* 2^{-(\mathcal{S}-1)} = p^* 2^{-H}$. It follows that $E[l^*] \geq 2^{\mathcal{S}-1} - 1$ and $\liminf_{T\to\infty} \text{Regret}(T, \mathcal{M}) \geq 2^{\mathcal{S}-1} - 1$.*

4. Randomized value functions

We now consider an alternative approach to exploration that involves randomly sampling value functions rather than actions. As a specific scheme of this kind, we propose randomized least-squares value iteration (RLSVI), which we present as Algorithm 1.[1] To obtain an RL algorithm, we simply select greedy actions in each episode, as specified in Algorithm 2.

The manner in which RLSVI explores is inspired by Thompson sampling (Thompson, 1933), which has been shown to explore efficiently across a very general class of online optimization problems (Russo & Van Roy, 2013; 2014). In Thompson sampling, the agent samples from a posterior distribution over models, and selects the action that optimizes the sampled model. RLSVI similarly samples from a distribution over plausible value functions and selects actions that optimize resulting samples. This distribution can be thought of as an approximation to a posterior distribution over value functions. RLSVI bears a close connection to PSRL (Osband et al., 2013), which maintains and samples from a posterior distribution over MDPs and is a direct application of Thompson sampling to RL. PSRL satisfies regret bounds that scale with the dimensionality, rather than the cardinality, of the underlying MDP (Osband & Van Roy, 2014b;a). However, PSRL does not accommodate value function generalization without MDP planning, a feature that we expect to be of great practical importance.

[1] Note that when $l = 0$, both A and b are empty, hence, we set $\tilde{\theta}_{l0} = \tilde{\theta}_{l1} = \cdots = \tilde{\theta}_{l,H-1} = 0$.

Algorithm 1 Randomized Least-Squares Value Iteration

Input: Data $\Phi_0(s_{i0},a_{i0}),r_{i0},..,\Phi_{H-1}(s_{iH-1},a_{iH-1}),r_{iH}$: $i < L$, Parameters $\lambda > 0$, $\sigma > 0$
Output: $\tilde{\theta}_{l0},..,\tilde{\theta}_{l,H-1}$

1: **for** $h = H-1,..,1,0$ **do**
2: Generate regression problem $A \in \mathbb{R}^{l\times K}$, $b \in \mathbb{R}^l$:

$$A \leftarrow \begin{bmatrix} \Phi_h(s_{0h},a_{0h}) \\ \vdots \\ \Phi_h(s_{l-1,h},a_{l-1,h}) \end{bmatrix}$$

$$b_i \leftarrow \begin{cases} r_{ih} + \max_\alpha \left(\Phi_{h+1}\tilde{\theta}_{l,h+1}\right)(s_{i,h+1},\alpha) & \text{if } h < H-1 \\ r_{ih} + r_{i,h+1} & \text{if } h = H-1 \end{cases}$$

3: Bayesian linear regression for the value function

$$\overline{\theta}_{lh} \leftarrow \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1} A^\top b$$
$$\Sigma_{lh} \leftarrow \left(\frac{1}{\sigma^2}A^\top A + \lambda I\right)^{-1}$$

4: Sample $\tilde{\theta}_{lh} \sim N(\overline{\theta}_{lh}, \Sigma_{lh})$ from Gaussian posterior
5: **end for**

Algorithm 2 RLSVI with greedy action

Input: Features $\Phi_0,..,\Phi_{H-1}$; $\sigma > 0$, $\lambda > 0$

1: **for** $l = 0,1,..$ **do**
2: Compute $\tilde{\theta}_{l0},..,\tilde{\theta}_{l,H-1}$ using Algorithm 1
3: Observe s_{l0}
4: **for** $h = 0,..,H-1$ **do**
5: Sample $a_{lh} \in \text{argmax}_{\alpha\in\mathcal{A}}\left(\Phi_h\tilde{\theta}_{lh}\right)(s_{lh},\alpha)$
6: Observe r_{lh} and $s_{l,h+1}$
7: **end for**
8: Observe r_{lH}
9: **end for**

5. Provably efficient tabular learning

RLSVI is an algorithm designed for efficient exploration in large MDPs with linear value function generalization. So far, there are no algorithms with analytical regret bounds in this setting. In fact, most common methods are provably *inefficient*, as demonstrated in Example 1, regardless of the choice of basis function. In this section we will establish an expected regret bound for RLSVI in a tabular setting without generalization where the basis functions $\Phi_h = I$.

The bound is on an expectation with respect to a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We define the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, P, R, \pi)$ and all other random variables we will consider with respect to this probability space. We assume that $\mathcal{S}$, $\mathcal{A}$, H, and π, are deterministic and that R and P are drawn from a prior distribution. We will assume that

rewards $R(s,a,h)$ are drawn from independent Dirichlet $\alpha^R(s,a,h) \in \mathbb{R}^2_+$ with values on $\{-1,0\}$ and transitions Dirichlet $\alpha^P(s,a,h) \in \mathbb{R}^S_+$. Analytical techniques exist to extend similar results to general bounded distributions; see, for example (Agrawal & Goyal, 2012).

Theorem 1. *If Algorithm 1 is executed with $\Phi_h = I$ for $h = 0,..,H-1$, $\lambda \geq \max_{(s,a,h)} \left(\mathbb{1}^T\alpha^R(s,a,h) + \mathbb{1}^T\alpha^P(s,a,h)\right)$ and $\sigma \geq \sqrt{H^2+1}$, then:*

$$\mathbb{E}\left[\text{Regret}(T,\mathcal{M})\right] \leq \tilde{O}\left(\sqrt{H^3SAT}\right) \quad (1)$$

Surprisingly, these scalings better state of the art optimistic algorithms specifically designed for efficient analysis which would admit $\tilde{O}(\sqrt{H^3S^2AT})$ regret (Jaksch et al., 2010). This is an important result since it demonstrates that RLSVI can be provably-efficient, in contrast to popular dithering approaches such as ϵ-greedy which are provably *inefficient*.

5.1. Preliminaries

Central to our analysis is the notion of stochastic optimism, which induces a partial ordering among random variables.

Definition 1. *For any real-valued random variables X and Y we say that X is stochastically optimistic with respect to Y if for any $u:\mathbb{R}\to\mathbb{R}$ convex and increasing*

$$\mathbb{E}[u(X)] \geq \mathbb{E}[u(Y)].$$

We will use the notation $X \succcurlyeq_{\text{so}} Y$ to express this relation.

It is worth noting that stochastic optimism is closely connected with second-order stochastic dominance: $X \succcurlyeq_{\text{so}} Y$ if and only if $-Y$ second-order stochastically dominates $-X$ (Hadar & Russell, 1969). We reproduce the following result which establishes such a relation involving Gaussian and Dirichlet random variables in Appendix G.

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

5.2. Proof sketch

Let $\tilde{Q}^l_h = \Phi_h\tilde{\theta}_{lh}$ and $\tilde{\mu}_l$ denote the value function and policy generated by RLSVI for episode l and let $\tilde{V}^l_h(s) = \max_a \tilde{Q}^l_h(s,a)$. We can decompose the per-episode regret

$$V^*_0(s_{l0}) - V^{\tilde{\mu}_l}_0(s_{l0}) = \underbrace{\tilde{V}^l_0(s_{l0}) - V^{\tilde{\mu}_l}_0(s_{l0})}_{\Delta^{\text{conc}}_l} + \underbrace{V^*_0(s_{l0}) - \tilde{V}^l_0(s_{l0})}_{\Delta^{\text{opt}}_l}.$$

We will bound this regret by first showing that RLSVI generates optimistic estimates of V^*, so that Δ^{opt}_l has non-positive expectation for any history $\mathcal{H}_l$ available prior to episode l. The remaining term Δ^{conc}_l vanishes as estimates generated by RLSVI concentrate around V^*.

Lemma 2. *Conditional on any data $\mathcal{H}$, the Q-values generated by RLSVI are stochastically optimistic with respect to the true Q-values $\tilde{Q}^l_h(s,a) \succcurlyeq_{\text{so}} Q^*_h(s,a)$ for all s,a,h.*

Proof. Fix any data $\mathcal{H}_l$ available and use backwards induction on $h = H-1,..,1$. For any (s,a,h) we write $n(s,a,h)$ for the amount of visits to that datapoint in $\mathcal{H}_l$. We will write $\hat{R}(s,a,h), \hat{P}(s,a,h)$ for the empirical mean reward and mean transitions based upon the data $\mathcal{H}_l$. We can now write the posterior mean rewards and transitions:

$$\overline{R}(s,a,h)|\mathcal{H}_l = \frac{-1 \times \alpha^R_1(s,a,h) + n(s,a,h)\hat{R}(s,a,h)}{\mathbb{1}^T\alpha^R(s,a,h) + n(s,a,h)}$$

$$\overline{P}(s,a,h)|\mathcal{H}_l = \frac{\alpha^P(s,a,h) + n(s,a,h)\hat{P}(s,a,h)}{\mathbb{1}^T\alpha^P(s,a,h) + n(s,a,h)}$$

Now, using $\Phi_h = I$ for all (s,a,h) we can write the RLSVI updates in similar form. Note that, Σ_{lh} is diagonal with each diagonal entry equal to $\sigma^2/(n(s,a,h) + \lambda\sigma^2)$. In the case of $h = H-1$

$$\overline{\theta}^l_{H-1}(s,a) = \frac{n(s,a,H-1)\hat{R}(s,a,H-1)}{n(s,a,H-1) + \lambda\sigma^2}$$

Using the relation that $\hat{R} \geq \overline{R}$ Lemma 1 means that

$$N(\overline{\theta}^l_{H-1}(s,a), \frac{1}{n(s,a,h) + \mathbb{1}^T\alpha^R(s,a,h)}) \succcurlyeq_{\text{so}} R_{H-1}|\mathcal{H}_l.$$

Therefore, choosing $\lambda > \max_{s,a,h} \mathbb{1}^T\alpha^R(s,a,h)$ and $\sigma > 1$, we must satisfy the lemma for all s,a and $h = H-1$.

For the inductive step we assume that the result holds for all s,a and $j > h$, we now want to prove the result for all (s,a) at timestep h. Once again, we can express $\overline{\theta}^l_h(s,a)$ in closed form.

$$\overline{\theta}^l_h(s,a) = \frac{n(s,a,h)\left(\hat{R}(s,a,h) + \hat{P}(s,a,h)^T\tilde{V}^l_{h+1}\right)}{n(s,a,h) + \lambda\sigma^2}$$

To simplify notation we omit the arguments (s,a,h) where they should be obvious from context. The posterior mean estimate for the next step value V^*_h, conditional on $\mathcal{H}_l$:

$$\mathbb{E}[Q^*_h(s,a)|\mathcal{H}_l] = \overline{R} + \overline{P}^TV^*_{h+1} \leq \frac{n(\hat{R} + \hat{P}^TV^*_{h+1})}{n + \lambda\sigma^2}.$$

As long as $\lambda > \mathbb{1}^T\alpha^R + \mathbb{1}^T(\alpha^P)$ and $\sigma^2 > H^2$. By our induction process $\tilde{V}^l_{h+1} \succcurlyeq_{\text{so}} V^*_{h+1}$ so that

$$\mathbb{E}[Q^*_h(s,a)|\mathcal{H}_l] \leq \mathbb{E}\left[\frac{n(\hat{R} + \hat{P}^T\tilde{V}^l_{h+1})}{n + \lambda\sigma^2} \mid \mathcal{H}_l\right].$$

We can conclude by Lemma 1 and noting that the noise from rewards is dominated by $N(0,1)$ and the noise from transitions is dominated by $N(0,H^2)$. This requires that $\sigma^2 \geq H^2 + 1$. □

Lemma 2 means RLSVI generates stochastically optimistic Q-values for any history $\mathcal{H}_l$. All that remains is to prove the remaining estimates $\mathbb{E}[\Delta_l^{\text{conc}}|\mathcal{H}_l]$ concentrate around the true values with data. Intuitively this should be clear, since the size of the Gaussian perturbations decreases as more data is gathered. In the remainder of this section we will sketch this result.

The concentration error $\Delta_l^{\text{conc}} = \tilde{V}_0^l(s_{l0}) - V_0^{\tilde{\mu}_l}(s_{l0})$. We decompose the value estimate $\tilde{V}_0^l$ explicitly:

$$\begin{aligned}\tilde{V}_0^l(s_{l0}) &= \frac{n(\hat{R} + \hat{P}^T\tilde{V}_{h+1}^l)}{n+\lambda\sigma^2} + w^\sigma \\ &= \overline{R} + \overline{P}^T\tilde{V}_{h+1}^l + b^R + b^P + w_0^\sigma\end{aligned}$$

where w^σ is the Gaussian noise from RLSVI and $b^R = b^R(s_{l0}, a_{l0}0), b^P = b^P(s_{l0}, a_{l0}0)$ are optimistic bias terms for RLSVI. These terms emerge since RLSVI shrinks estimates towards zero rather than the Dirichlet prior for rewards and transitions.

Next we note that, conditional on $\mathcal{H}_l$ we can rewrite $\overline{P}^T\tilde{V}_{h+1}^l = \tilde{V}_{h+1}^l(s') + d_h$ where $s' \sim P^*(s,a,h)$ and d_h is some martingale difference. This allows us to decompose the error in our policy to the estimation error of the states and actions we actually visit. We also note that, conditional on the data $\mathcal{H}_l$ the true MDP is independent of the sampling process of RLSVI. This means that:

$$\mathbb{E}[V_0^{\tilde{\mu}_l}(s_{l0})|\mathcal{H}_l] = \overline{R} + \overline{P}^T V_{h+1}^{\tilde{\mu}_l}.$$

Once again, we can replace this transition term with a single sample $s' \sim P^*(s,a,h)$ and a martingale difference. Combining these observations allows us to reduce the concentration error

$$\mathbb{E}[\tilde{V}_0^l(s_{l0}) - V_0^{\tilde{\mu}_l}(s_{l0})|\mathcal{H}_l] = \sum_{h=0}^{H-1}\left\{b^R(s_{lh},a_{lh},h) + b^P(s_{lh},a_{lh},h) + w_h^\sigma\right\}.$$

We can even write explicit expressions for b^R, b^P and w^σ.

$$\begin{aligned}b^R(s,a,h) &= \frac{n\hat{R}}{n+\lambda\sigma^2} - \frac{n\hat{R}-\alpha_1^R}{n+\mathbb{1}^T\alpha^R} \\ b^P(s,a,h) &= \frac{n\hat{P}^T\tilde{V}_{h+1}^l}{n+\lambda\sigma^2} - \frac{(n\hat{P}+\alpha^P)^T\tilde{V}_{h+1}^l}{n+\mathbb{1}^T\alpha^P} \\ w_h^\sigma &\sim N\left(0, \frac{\sigma^2}{n+\lambda\sigma^2}\right)\end{aligned}$$

The final details for this proof are technical but the argument is simple. We let $\lambda = \mathbb{1}^T\alpha^R + \mathbb{1}^T\alpha^P$ and $\sigma = \sqrt{H^2+1}$. Up to $\tilde{O}$ notation $b^R \simeq \frac{\alpha_1^R}{n+\mathbb{1}^T\alpha^P}$, $b^P \simeq \frac{H\mathbb{1}^T\alpha^P}{n+\mathbb{1}^T\alpha^P}$ and $w_h^\sigma \simeq \frac{H}{\sqrt{n+H^2\mathbb{1}^T\alpha^R+\mathbb{1}^T\alpha^P}}$. Summing using a pigeonhole principle for $\sum_{s,a,h} n(s,a,h) = T$ gives us an upper bound on the regret. We write $K(s,a,h) := \left(\alpha_1^R(s,a,h) + H\mathbb{1}^T\alpha^P(s,a,h)\right)$ to bound the effects of the prior mistmatch in RLSVI arising from the bias terms b^R, b^P. The constraint $\alpha^T\mathbb{1} \geq 2$ can only be violated twice for each s,a,h. Therefore up to $O(\cdot)$ notation:

$$\mathbb{E}\left[\sum_{l=0}^{T/H-1}\mathbb{E}[\Delta_l^{\text{conc}}|\mathcal{H}_l]\right] \leq 2SAH + \sum_{s,a,h} K(s,a,h)\log(T + K(s,a,h)) + H\sqrt{SAHT\log(T)}$$

□

6. Experiments

Our analysis in Section 5 shows that RLSVI with tabular basis functions acts as an effective Gaussian approximation to PSRL. This demonstrates a clear distinction between exploration via randomized value functions and dithering strategies such as Example 1. However, the motivation for RLSVI is not for tabular environments, where several provably efficient RL algorithms already exist, but instead for large systems that require generalization.

We believe that, under some conditions, it may be possible to establish polynomial regret bounds for RLSVI with value function generalization. To stimulate thinking on this topic we present a conjecture of a result that may be possible in Appendix B. For now, we will present a series of experiments designed to test the applicability and scalability of RLSVI for exploration with generalization.

Our experiments are divided into three sections. First, we present a series of didactic chain environments similar to Figure 1. We show that RLSVI can effectively synthesize exploration with generalization with both coherent and agnostic value functions that are intractable under any dithering scheme. Next, we apply our Algorithm to learning to play Tetris. We demonstrate that RLSVI leads to faster learning, improved stability and a superior learned policy in a large-scale video game. Finally, we consider a business application with a simple model for a recommendation system. We show that an RL algorithm can improve upon even the optimal myopic bandit strategy. RLSVI learns this optimal strategy when dithering strategies do not.

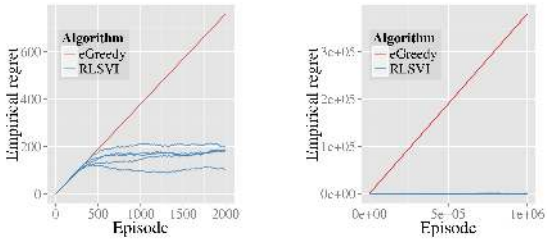
6.1. Testing for efficient exploration

We now consider a series of environments modelled on Example 1, where dithering strategies for exploration are provably inefficient. Importantly, and unlike the tabular setting of Section 5, our algorithm will only interact with the MDP but through a set of basis function Φ which generalize across states. We examine the empirical performance of RLSVI and find that it does efficiently balance exploration and generalization in this didactic example.

6.1.1. Coherent learning

In our first experiments, we generate a random set of K basis functions. This basis is coherent but the individual basis

functions are not otherwise informative. We form a random linear subspace V_{hK} spanned by $(\mathbb{1}, Q_h^*, \tilde{w}_1, .., \tilde{w}_{k-2})$. Here w_i and $\tilde{w}_i$ are IID Gaussian $\sim N(0, I) \in \mathbb{R}^{SA}$. We then form Φ_h by projecting $(\mathbb{1}, w_1, .., w_{k-1})$ onto V_{hK} and renormalize each component to have equal 2-norm[2]. Figure 2 presents the empirical regret for RLSVI with $K=10, N=50, \sigma=0.1, \lambda=1$ and an ϵ-greedy agent over 5 seeds[3].



(a) First 2000 episodes (b) First 10^6 episodes

Figure 2. Efficient exploration on a 50-chain

Figure 1 shows that RLSVI consistently learns the optimal policy in roughly 500 episodes. Any dithering strategy would take at least 10^{15} episodes for this result. The state of the art upper bounds for the efficient optimistic algorithm UCRL given by appendix C.5 in (Dann & Brunskill, 2015) for $H=15, S=6, A=2, \epsilon=1, \delta=1$ only kick in after more than 10^{10} suboptimal episodes. RLSVI is able to effectively exploit the generalization and prior structure from the basis functions to learn much faster.

We now examine how learning scales as we change the chain length N and number of basis functions K. We observe that RLSVI essentially maintains the optimal policy once it discovers the rewarding state. We use the number of episodes until 10 rewards as a proxy for learning time. We report the average of five random seeds.

Figure 3 examines the time to learn as we vary the chain length N with fixed $K=10$ basis functions. We include the dithering lower bound 2^{N-1} as a dashed line and a lower bound scaling $\frac{1}{10}H^2SA$ for tabular learning algorithms as a solid line (Dann & Brunskill, 2015). For $N=100$, $2^{N-1} > 10^{28}$ and $H^2SA > 10^6$. RLSVI demonstrates scalable generalization and exploration to outperform these bounds.

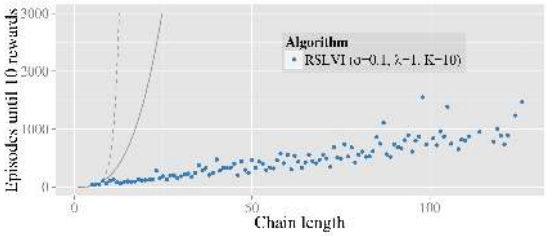


Figure 3. RLSVI learning time against chain length.

Figure 4 examines the time to learn as we vary the basis functions K in a fixed $N=50$ length chain. Learning time scales gracefully with K. Further, the marginal effect of K decrease as $\dim(V_{hK})=K$ approaches $\dim(\mathbb{R}^{SA})=100$. We include a local polynomial regression in blue to highlight this trend. Importantly, even for large K the performance is far superior to the dithering and tabular bounds[4].

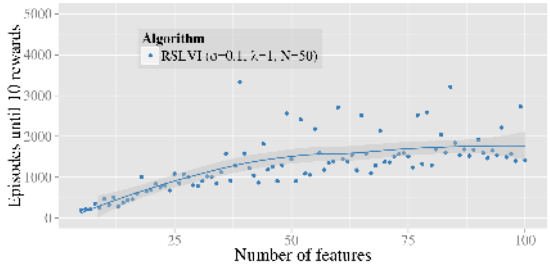


Figure 4. RLSVI learning time against number of basis features.

Figure 5 examines these same scalings on a logarithmic scale. We find the data for these experiments is consistent with polynomial learning as hypothesized in Appendix B. These results are remarkably robust over several orders of magnitude in both σ and λ. We present more detailed analysis of these sensitivities in Appendix C.

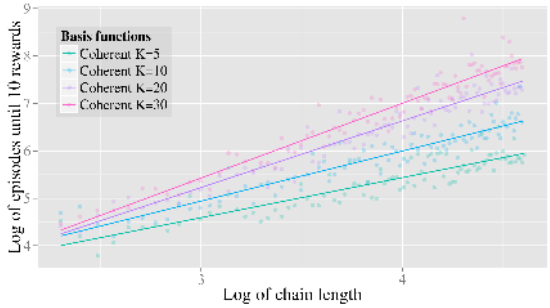


Figure 5. Empirical support for polynomial learning in RLSVI.

6.1.2. Agnostic learning

Unlike the example above, practical RL problems will typically be agnostic. The true value function Q_h^* will not lie within V_{hK}. To examine RLSVI in this setting we generate basis functions by adding Gaussian noise to the true value function $\phi_{hk} \sim N(Q_h^*, \rho I)$. The parameter ρ determines the scale of this noise. For $\rho = 0$ this problem is coherent but for $\rho > 0$ this will typically not be the case. We fix $N = 20, K = 20, \sigma = 0.1$ and $\lambda = 1$.

For $i=0,..,1000$ we run RLSVI for 10,000 episodes with $\rho = i/1000$ and a random seed. Figure 6 presents the number of episodes until 10 rewards for each value of ρ. For large values of ρ, and an extremely misspecified basis, RLSVI is not effective. However, there is some region $0 < \rho < \rho^*$ where learning remains remarkably stable[5].

[2]For more details on this experiment see Appendix C.

[3]In this setting any choice of ϵ or Boltzmann η is equivalent.

[4]For chain $N=50$, the bounds $2^{N-1} > 10^{14}$ and $H^2SA > 10^5$.

[5]Note $Q_h^*(s,a) \in \{0,1\}$ so $\rho=0.5$ represents significant noise.

This simple example gives us some hope that RLSVI can be useful in the agnostic setting. In our remaining experiments we will demonstrate that RLSVI can achieve state of the art results in more practical problems with agnostic features.

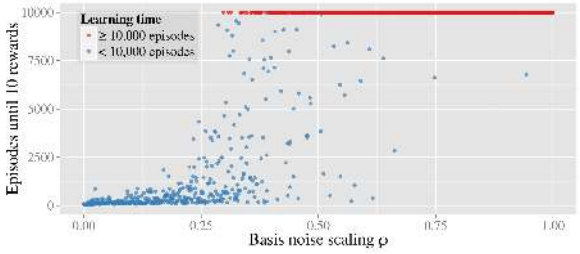


Figure 6. RLSVI is somewhat robust to model mispecification.

6.2. Tetris

We now turn our attention to learning to play the iconic video game Tetris. In this game, random blocks fall sequentially on a 2D grid with 20 rows and 10 columns. At each step the agent can move and rotate the object subject to the constraints of the grid. The game starts with an empty grid and ends when a square in the top row becomes full. However, when a row becomes full it is removed and all bricks above it move downward. The objective is to maximize the score attained (total number of rows removed) before the end of the game.

Tetris has been something of a benchmark problem for RL and approximate dynamic programming, with several papers on this topic (Gabillon et al., 2013). Our focus is not so much to learn a high-scoring Tetris player, but instead to demonstrate the RLSVI offers benefits over other forms of exploration with LSVI. Tetris is challenging for RL with a huge state space with more than 2^{200} states. In order to tackle this problem efficiently we use 22 benchmark features. These features give the height of each column, the absolute difference in height of each column, the maximum height of a column, the number of "holes" and a constant. It is well known that you can find far superior linear basis functions, but we use these to mirror their approach.

In order to apply RLSVI to Tetris, which does not have fixed episode length, we made a few natural modifications to the algorithm. First, we approximate a time-homogeneous value function. We also only the keep most recent $N{=}10^5$ transitions to cap the linear growth in memory and computational requirements, similar to (Mnih, 2015). Details are provided in Appendix D. In Figure 7 we present learning curves for RLSVI $\lambda{=}1,\sigma{=}1$ and LSVI with a tuned ϵ-greedy exploration schedule[6] averaged over 5 seeds. The results are significant in several ways.

First, both RLSVI and LSVI make significant improvements over the previous approach of LSPI with the same basis functions (Bertsekas & Ioffe, 1996). Both algorithms reach higher final performance ($\simeq$ 3500 and 4500 respectively) than the best level for LSPI (3183). They also reach this performance after many fewer games and, unlike LSPI do not "collapse" after finding their peak performance. We believe that these improvements are mostly due to the memory replay buffer, which stores a bank of recent past transitions, rather than LSPI which is purely online.

Second, both RLSVI and LSVI learn from scratch where LSPI required a scoring initial policy to begin learning. We believe this is due to improved exploration schemes, LSPI is completely greedy so struggles to learn without an initial policy. LSVI with a tuned ϵ schedule is much better. However, we do see a significant improvement through exploration via RLSVI even when compared to the tuned ϵ scheme. This outperformance becomes much more extreme on a variant of Tetris with only 5 rows that highlights the need for efficient exploration. More details are available in Appendix D.

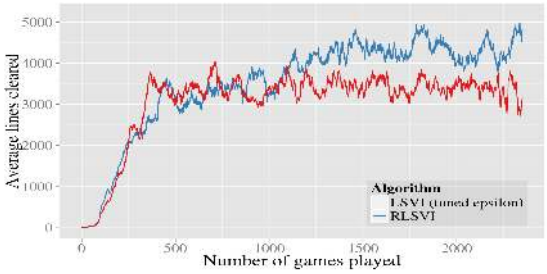


Figure 7. Learning Tetris with Bertsekas-Ioffe features.

6.3. A recommendation engine

We will now show that efficient exploration and generalization can be helpful in a simple model of customer interaction. Consider an agent which recommends $J \leq N$ products from $\mathcal{Z} = \{1, 2, \ldots, N\}$ sequentially to a customer. The conditional probability that the customer likes a product depends on the product, some items are better than others. However it also depends on what the user has observed, what she liked and what she disliked. We represent the products the customer has seen by $\tilde{\mathcal{Z}} \subseteq \mathcal{Z}$. For each product $n \in \tilde{\mathcal{Z}}$ we will indicate $x_n \in \{-1, +1\}$ for her preferences {dislike, like} respectively. If the customer has not observed the product $n \notin \tilde{\mathcal{Z}}$ we will write $x_n = 0$. We model the probability that the customer will like a new product $a \notin \tilde{\mathcal{Z}}$ by a logistic transformation linear in x:

$$\mathbb{P}(a|x) = 1/\left(1 + \exp\left(-\left[\beta_a + \textstyle\sum_n \gamma_{an} x_n\right]\right)\right). \quad (2)$$

Importantly, this model reflects that the customers' preferences may evolve as their experiences change. For example, a customer may be much more likely to watch the second season of the TV show "Breaking Bad" if they have watched the first season and liked it.

[6] We found that we could not acheive good performance for any fixed ϵ. We used an annealing exploration schedule that was tuned to give good performance. See Appendix D

The agent in this setting is the recommendation system, whose goal is to maximize the cumulative amount of items liked through time for each customer. The agent does not know $p(a|x)$ initially, but can learn to estimate the parameters β, γ through interactions across different customers. Each customer is modeled as an episode with horizon length $H = J$ with a "cold start" and no previous observed products $\tilde{\mathcal{Z}} = \emptyset$. For our simulations we set $\beta_a = 0$ $\forall a$ and sample a random problem instance by sampling $\gamma_{an} \sim N(0, c^2)$ independently for each a and n.

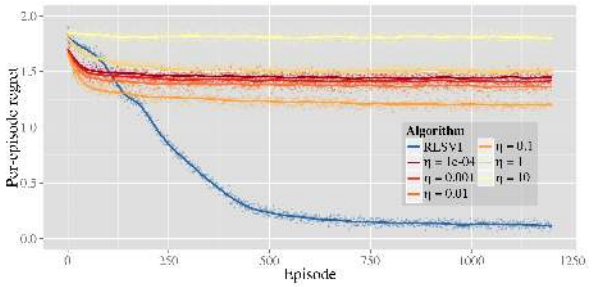


Figure 8. RLSVI performs better than Boltzmann exploration.

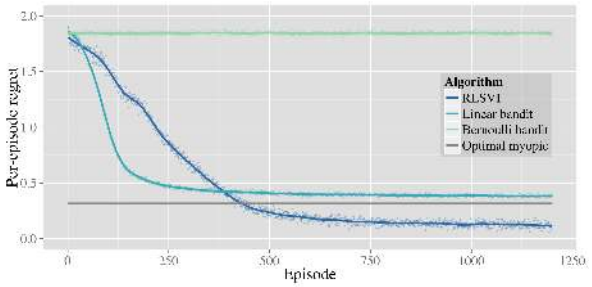


Figure 9. RLSVI can outperform the optimal myopic policy.

Although this setting is simple, the number of possible states $|\mathcal{S}| = |\{-1, 0, +1\}|^H = 3^J$ is exponential in J. To learn in time less than $|\mathcal{S}|$ it is crucial that we can exploit generalization between states as per equation (2). For this problem we constuct the following simple basis functions: $\forall 1 \leq n, m, a \leq N$, let $\phi_m(x, a) = \mathbf{1}\{a = m\}$ and $\phi_{mn}(x, a) = x_n \mathbf{1}\{a = m\}$. In each period h form $\Phi_h = ((\phi_n)_n, (\phi_m)_m)$. The dimension of our function class $K = N^2 + N$ is exponentially smaller than the number of states. However, barring a freak event, this simple basis will lead to an agnostic learning problem.

Figure 8 and 9 show the performance of RLSVI compared to several benchmark methods. In Figure 8 we plot the cumulative regret of RLSVI when compared against LSVI with Boltzmann exploration and identical basis features. We see that RLSVI explores much more efficiently than Boltzmann exploration over a wide range of temperatures.

In Figure 9 we show that, using this efficient exploration method, the reinforcement learning policy is able to outperform not only benchmark bandit algorithms but even the optimal myopic policy[7]. Bernoulli Thompson sampling does not learn much even after 1200 episodes, since the algorithm does not take *context* into account. The linear contextual bandit outperforms RLSVI at first. This is not surprising, since learning a myopic policy is simpler than a multi-period policy. However as more data is gathered RLSVI eventually learns a richer policy which outperforms the myopic policy.

Appendix E provides pseudocode for this computational study. We set $N = 10$, $H = J = 5$, $c = 2$ and $L = 1200$. Note that such problems have $|\mathcal{S}| = 4521$ states; this allows us to solve each MDP exactly so that we can compute regret. Each result is averaged over 100 problem instances and for each problem instance, we repeat simulations 10 times. The cumulative regret for both RLSVI (with $\lambda = 0.2$ and $\sigma^2 = 10^{-3}$) and LSVI with Boltzmann exploration (with $\lambda = 0.2$ and a variety of "temperature" settings η) are plotted in Figure 8. RLSVI clearly outperforms LSVI with Boltzmann exploration.

Our simulations use an extremely simplified model. Nevertheless, they highlight the potential value of RL over multi-armed bandit approaches in recommendation systems and other customer interactions. An RL algorithm may outperform even an optimal myopic system, particularly where large amounts of data are available. In some settings, efficient generalization and exploration can be crucial.

7. Closing remarks

We have established a regret bound that affirms efficiency of RLSVI in a *tabula rasa learning* context. However the real promise of RLSVI lies in its potential as an efficient method for exploration in large-scale environments with generalization. RLSVI is simple, practical and explores efficiently in several environments where state of the art approaches are ineffective.

We believe that this approach to exploration via randomized value functions represents an important concept beyond our specific implementation of RLSVI. RLSVI is designed for generalization with linear value functions, but many of the great successes in RL - from Backgammon (Tesauro, 1995) to Atari[8] (Mnih, 2015) - have made use of highly nonlinear "deep" neural networks. The insights of this paper and of generalization and exploration via randomized value functions should extend to nonlinear contexts. For example, one could approximate posterior samples of nonlinearly parameterized value functions via the bootstrap (Osband & Van Roy, 2015).

[7] The optimal myopic policy knows the true model defined in Equation 2, but does not plan over multiple timesteps.

[8] Interestingly, recent work has been able to reproduce similar performance using linear value functions (Liang et al., 2015).

References

Abbasi-Yadkori, Yasin and Szepesvári, Csaba. Regret bounds for the adaptive control of linear quadratic systems. *Journal of Machine Learning Research - Proceedings Track*, 19:1–26, 2011.

Agrawal, Shipra and Goyal, Navin. Further optimal regret bounds for Thompson sampling. *arXiv preprint arXiv:1209.3353*, 2012.

Bertsekas, Dimitri P and Ioffe, Sergey. Temporal differences-based policy iteration and applications in neuro-dynamic programming. *Lab. for Info. and Decision Systems Report LIDS-P-2349, MIT, Cambridge, MA*, 1996.

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

Dann, Christoph and Brunskill, Emma. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2800–2808, 2015.

Dearden, Richard, Friedman, Nir, and Russell, Stuart J. Bayesian Q-learning. In *AAAI/IAAI*, pp. 761–768, 1998.

Gabillon, Victor, Ghavamzadeh, Mohammad, and Scherrer, Bruno. Approximate dynamic programming finally performs well in the game of tetris. In *Advances in Neural Information Processing Systems*, pp. 1754–1762, 2013.

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

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

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

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

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

Lagoudakis, Michail, Parr, Ronald, and Littman, Michael L. Least-squares methods in reinforcement learning for control. In *Second Hellenic Conference on Artificial Intelligence (SETN-02)*, 2002.

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

Li, Lihong and Littman, Michael. Reducing reinforcement learning to KWIK online regression. *Annals of Mathematics and Artificial Intelligence*, 2010.

Li, Lihong, Littman, Michael L., and Walsh, Thomas J. Knows what it knows: a framework for self-aware learning. In *ICML*, pp. 568–575, 2008.

Liang, Yitao, Machado, Marlos C., Talvitie, Erik, and Bowling, Michael H. State of the art control of atari games using shallow reinforcement learning. *CoRR*, abs/1512.01563, 2015. URL http://arxiv.org/abs/1512.01563.

Mnih, Volodymyr et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Ortner, Ronald and Ryabko, Daniil. Online regret bounds for undiscounted continuous reinforcement learning. In *NIPS*, 2012.

Osband, Ian and Van Roy, Benjamin. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014a.

Osband, Ian and Van Roy, Benjamin. Near-optimal reinforcement learning in factored MDPs. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2014b.

Osband, Ian and Van Roy, Benjamin. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

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

Pazis, Jason and Parr, Ronald. PAC optimal exploration in continuous space Markov decision processes. In *AAAI*. Citeseer, 2013.

Russo, Dan and Van Roy, Benjamin. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pp. 2256–2264. Curran Associates, Inc., 2013.

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

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

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

Szepesvári, Csaba. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

Tesauro, Gerald. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

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

Wen, Zheng and Van Roy, Benjamin. Efficient exploration and value function generalization in deterministic systems. In *NIPS*, pp. 3021–3029, 2013.