Variational Bayesian Reinforcement Learning with Regret Bounds

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the exploration-exploitation trade-off in reinforcement learning and show that an agent endowed with an exponential epistemic-risk-seeking utility function explores efficiently, as measured by regret. The state-action values induced by the exponential utility satisfy a Bellman recursion, so we can use dynamic programming to compute them. We call the resulting algorithm K-learning (for knowledge) and the risk-seeking utility ensures that the associated state-action values (K-values) are optimistic for the expected optimal Q-values under the posterior. The exponential utility function induces a Boltzmann exploration policy for which the 'temperature' parameter is equal to the risk-seeking parameter and is carefully controlled to yield a Bayes regret bound of $\tilde{O}(L^{3/2}\sqrt{SAT})$, where L is the time horizon, S is the number of states, A is the number of actions, and T is the total number of elapsed timesteps. We conclude with a numerical example demonstrating that K-learning is competitive with other state-of-the-art algorithms in practice.

1 Introduction and related work

In reinforcement learning (RL) an agent interacts with an environment in an episodic manner and attempts to maximize its return [48, 40]. In this work we shall assume that the environment is a Markov decision process (MDP) and we consider the Bayesian case where the agent has some prior information and as it gathers data it updates its posterior beliefs about the environment. In this setting the agent is faced with the choice of visiting well understood states or exploring the environment to determine the value of other states which might lead to a higher return. This trade-off is called the *exploration-exploitation* dilemma. One way to measure how well an agent balances this trade-off is a quantity called *regret*, which measures how sub-optimal the rewards the agent has received are so far, relative to the (unknown) optimal policy [11]. In the Bayesian case the natural quantity to consider is the Bayes regret, which is the expected regret under the agents prior information [15].

The optimal Bayesian policy can be formulated using *belief states*, but this is intractable for all but small problems [15]. Approximations to the optimal Bayesian policy exist, one of the most successful being Thompson sampling [47, 49] wherein the agent samples from the posterior over value functions and acts greedily with respect to that sample [34, 38, 21, 37]. It can be shown that this strategy yields both Bayesian and frequentist regret bounds under certain assumptions [3]. In practice, maintaining a posterior over value functions is intractable, and so instead the agent maintains the posterior over MDPs, and at each episode an MDP is sampled from this posterior, the value function for that sample is computed, and the policy acts greedily with respect to that value function. Due to the repeated sampling and computing of value functions this is practical only for very small problems, though attempts have been made to extend it [33, 32].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this work we endow an agent with a particular *epistemic risk-seeking*, utility function, where 'epistemic risk' refers to the uncertainty that the agent has about the optimal value function of the MDP. In the context of RL acting so as to maximize a risk-seeking utility function, which assigns higher values to more uncertain actions, is a form of *optimism in the face of uncertainty*, a well-known heuristic to encourage exploration [19, 4]. Any increasing convex function could be used as a risk-seeking utility, however only the exponential utility function has a decomposition property which is required to derive a Bellman recursion [1, 39, 18, 41]. We show that the *certainty equivalent value* under the risk-seeking utility satisfies a Bellman recursion with a new optimistic Bellman operator. We call the fixed point of this Bellman operator the 'K-values' for *knowledge* since they compress the quality and the uncertainty at any state-action into a single quantity. A high K-value captures the fact that the state-action has a high Q-value or high uncertainty. Following a Boltzmann policy over the K-values yields a practical algorithm that we call 'K-learning' which attains a Bayes regret upper bounded by $\tilde{O}(L^{3/2}\sqrt{SAT})$ where L is the time horizon, S is the number of states, A is the number of actions per state, and T is the total number of elapsed timesteps [10]. This regret bound matches the best known bound for Thompson sampling up to log factors [34] and is within a factor of $\sqrt{L}$ of the known minimax lower bound of $\Omega(L\sqrt{SAT})$ [19].

The *information ratio* is a statistic that captures how an algorithm is trading off regret and information acquisition [43, 22]. A small information ratio implies that the agent is either enjoying low regret or is acquiring information about the optimal policy. If we know a bound on the information ratio then it can be directly translated into a bound on the Bayes regret, which is the case for Thompson sampling and Information Directed Sampling [44]. This is similar to the approach in this paper based on utilities. If an agent visits a state with high K-value, *i.e.*, a high utility, either the expected Q-value is high (low regret) or the uncertainty is high (information acquisition), and this can be translated into a bound on Bayes regret. This suggests a connection between the information ratio and epistemic risk-seeking utilities, though we leave formalizing that connection to future work.

The update rule we derive is similar to that used in 'soft' Q-learning (so-called since the 'hard' max is replaced with a soft-max) [5, 14, 16, 26, 42]. These approaches are very closely related to maximum entropy reinforcement learning techniques which add an entropy regularization 'bonus' to prevent early convergence to deterministic policies and thereby heuristically encourage exploration [51, 52, 24, 30, 2]. In our work the soft-max operator and entropy regularization arise naturally from the view of the agent as maximizing a risk-seeking exponential utility. Furthermore, in contrast to these other approaches, the entropy regularization is not a fixed hyper-parameter but something we explicitly control (or optimize for) in order to carefully trade-off exploration and exploitation.

1.1 Summary of main results

- We consider an agent endowed an epistemic-risk-seeking utility function and derive a new optimistic Bellman operator that incorporates the 'value' deriving from epistemic uncertainty about the MDP. The new operator replaces the usual max operator with a soft-max and it incorporates a 'bonus' that depends on state-action visitation. In the limit of zero uncertainty the new operator reduces to the standard optimal Bellman operator.
- At each episode we solve the optimistic Bellman equation for the 'K-values' which represent the utility of a particular state and action. If the agent follows a Boltzmann policy over the K-values with a carefully chosen temperature schedule then it will enjoy a sublinear Bayes regret bound.
- To the best of our knowledge this is the first work to show that soft-max operators and maximum entropy policies in RL can provably yield good performance as measured by Bayes regret. Similarly, we believe this is the first result deriving a Bayesian regret bound for a Boltzmann policy in RL. This puts maximum entropy, soft-max operators, and Boltzmann exploration in a principled Bayesian context and shows that they are naturally derived from endowing the agent with an exponential utility function.

2 Markov decision processes

In a Markov decision process (MDP) an agent interacts with an environment in a series of episodes and attempts to maximize the cumulative rewards. A finite horizon, discrete-time, discrete MDP is given by the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, R, P, L, \rho\}$, where $\mathcal{S} = \{1, \ldots, S\}$ is the state-space, $\mathcal{A} =$

$\{1, \ldots, A\}$ is the action-space, $R(s, a)$ is a probability distribution over the rewards received by the agent at state s taking action a, $P_l(s' \mid s, a) \in [0, 1]$ is the probability the agent will transition to state s' after taking action a in state s, $L \in \mathbb{N}$ is the episode length, and ρ is the initial state distribution. Concretely, the initial state $s \in \mathcal{S}$ of the agent is sampled from ρ, then for timesteps $l = 1, \ldots, L$ the agent is in state $s \in \mathcal{S}$, selects action $a \in \mathcal{A}$, receives reward $r \sim R(s, a)$ and transitions to the next state s' with probability $P_l(s' \mid s, a)$. After timestep L the episode terminates and the state is reset. We assume that at the beginning of learning the agent does not know the reward or transition probabilities and must learn about them by interacting with the environment. We consider the Bayesian case in which the mean reward μ and the transition probabilities P are sampled from a known prior ϕ. For the analysis we will use the following assumption.

Assumption 1. *The MDP is time-inhomogeneous.*

A time-inhomogeneous MDP is one in which the state space factorizes as the union of L distinct sets, *i.e.*, $\mathcal{S} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_L$, where at timestep l the agent is in state $s \in \mathcal{S}_l$. This assumption implies that a state cannot be visited again within the same episode, *i.e.*, that the transition structure of the MDP gives the edges of a directed acyclic graph (sometimes this is referred to as 'loop-free'). This may seem very strong but in fact we may assume this without loss of generality since any finite horizon discrete MDP can be converted into one that satisfies this by 'unrolling' the MDP over time, whereby each state is replaced by L copies of itself and each copy indexed with the corresponding timestep l for $l = 1, \ldots, L$. In cases where the assumption does not hold we will first perform the unrolling procedure and then run our algorithm on this modified MDP, for which the assumption does hold. In these cases the state space cardinality goes from S to LS, which will affect the regret bound we derive. So with this in mind, hereafter we shall use $\mathcal{X}$ to represent the state-space, where

$$|\mathcal{X}| = \begin{cases} S & \text{if assumption 1 is true} \\ LS & \text{otherwise.} \end{cases}$$

An agent following policy $\pi : \mathcal{X} \times \mathcal{A} \to [0, 1]$ at state $s \in \mathcal{S}_l$ at time l selects actions as a with probability $\pi_l(s, a)$. The Bellman equation relates the value of actions taken at the current timestep to future returns through the *Q-values* and the associated *value function* [7], which for policy π are denoted $Q^\pi \in \mathbb{R}^{|\mathcal{X}| \times A}$ and $V^\pi \in \mathbb{R}^{|\mathcal{X}|}$, and satisfy

$$Q_l^\pi = \mathcal{T}_l^\pi Q_{l+1}^\pi, \quad V_l^\pi(s) = \sum_{a \in \mathcal{A}} \pi_l(s, a) Q_l^\pi(s, a), \tag{1}$$

for $l = 1, \ldots, L$ where $Q_{L+1} \equiv 0$ and where the Bellman operator for policy π at step l is defined as

$$(\mathcal{T}_l^\pi Q_{l+1}^\pi)(s, a) \coloneqq \mu_l(s, a) + \sum_{s' \in \mathcal{S}} P_l(s' \mid s, a) \sum_{a' \in \mathcal{A}} \pi_l(s', a') Q_{l+1}^\pi(s', a').$$

The expected performance of policy π is denoted $J^\pi = \mathbb{E}_{s \sim \rho} V_1^\pi(s)$. An optimal policy satisfies $\pi^\star \in \operatorname{argmax}_\pi J^\pi$ and induces associated optimal Q-values and value function given by

$$Q_l^\star = \mathcal{T}_l^\star Q_{l+1}^\star, \quad V_l^\star(s) = \max_a Q_l^\star(s, a). \tag{2}$$

for $l = 1, \ldots, L$, where $Q_{L+1}^\star \equiv 0$ and where the optimal Bellman operator is defined at step l as

$$(\mathcal{T}_l^\star Q_{l+1}^\star)(s, a) \coloneqq \mu_l(s, a) + \sum_{s' \in \mathcal{S}} P_l(s' \mid s, a) \max_{a'} Q_{l+1}^\star(s', a'). \tag{3}$$

2.1 Regret

If the mean reward μ and transition function P are known exactly then (in principle) we could solve (2) via dynamic programming [8]. However, in practice these are not known and so the agent must gather data by interacting with the environment over a series of episodes. The key trade-off is the *exploration-exploitation* dilemma, whereby an agent must take suboptimal actions in order to learn about the MDP. Here we are interested in the *regret* up to time T, which is how sub-optimal the agent's policy has been so far which for MDP $\mathcal{M}$ is defined as

$$\mathcal{R}_\mathcal{M}(T) \coloneqq \sum_{t=1}^{N} \mathbb{E}_{s \sim \rho}(V_1^\star(s) - V_1^{\pi^t}(s)),$$

where $N \coloneqq \lceil T/L \rceil$ is the number of elapsed episodes, and π^t is the policy that the agent follows at episode t. In this manuscript we take the case where $\mathcal{M}$ is sampled from a known prior ϕ and we want to minimize the expected regret of our algorithm under that prior distribution. This is referred to as the *Bayes regret*:

$$\mathcal{BR}_\phi(T) \coloneqq \mathbb{E}_{\mathcal{M}\sim\phi}\mathcal{R}_\mathcal{M}(T) = \mathbb{E}_{\mathcal{M}\sim\phi}\sum_{t=1}^{N}\mathbb{E}_{s\sim\rho}(V_1^\star(s) - V_1^{\pi^t}(s)). \tag{4}$$

In the Bayesian view of the RL problem the quantities μ and P are random variables, and consequently the optimal Q-values $Q^\star$, policy $\pi^\star$, and value function $V^\star$ are also random variables that must be learned about by gathering data from the environment. We shall denote by $\mathcal{F}_t$ the sigma-algebra generated by all the history *before* episode t where $\mathcal{F}_1 = \emptyset$ and we shall use $\mathbb{E}^t$ to denote $\mathbb{E}(\,\cdot \mid \mathcal{F}_t)$, the expectation conditioned on $\mathcal{F}_t$. For example, with this notation $\mathbb{E}^t Q^\star$ denotes the expected optimal Q-values under the posterior before the start of episode t.

3 K-learning

Now we present Knowledge Learning (K-learning), a Bayesian RL algorithm that comes with regret guarantees. At a high level the agent explores by maximizing an exponential risk-seeking utility function, where 'risk' refers to the epistemic uncertainty in the environment. This utility function is carefully tuned to trade-off exploration and exploitation. In this section we develop the intuition behind the approach; all the proofs are presented in the appendix.

3.1 Epistemic risk-seeking utility functions

A utility function $u : \mathbb{R} \to \mathbb{R}$ measures the benefit that an agent derives from a particular outcome [50]. For random variable $X : \Omega \to \mathbb{R}$ the expected utility is $\mathbb{E}u(X)$. If u is convex then it is referred to as *risk-seeking*, since due to Jensen's inequality an agent with a convex utility will prefer less predictable payoffs. The particular utility function we shall use is the exponential utility $u(x) \coloneqq \tau(\exp(x/\tau) - 1)$ for some $\tau \geq 0$. The *certainty equivalent value* of a random variable under utility u measures how much guaranteed payoff is equivalent to a random payoff, and for $Q_l^\star(s,a)$ under the exponential utility is given by

$$\mathcal{Q}_l^t(s,a) \coloneqq u^{-1}(\mathbb{E}^t u(Q_l^\star(s,a)) = \tau \log \mathbb{E}^t \exp(Q_l^\star(s,a)/\tau).$$

Jensen's inequality implies that $\mathcal{Q}_l^t(s,a) \geq \mathbb{E}^t Q_l^\star(s,a)$, *i.e.*, the certainty equivalent value is *optimistic* for the expected value of $Q_l^\star(s,a)$. The choice of the exponential utility may seem arbitrary, but in fact it is the unique utility function that has the property that the certainty equivalent value of the sum of two independent random variables is equal to the sum of their certainty equivalent values [1, 39, 18, 41]. This property is crucial for deriving a Bellman recursion which is necessary for dynamic programming to be applicable.

3.2 Optimistic Bellman operator

A risk-seeking agent would compute the certainty equivalent value of the Q-values under the endowed utility function and then act to maximize this value. The key result (proved in the appendix) is that $\mathcal{Q}^t$ satisfies a Bellman recursion with a particular optimistic (*i.e.*, risk-seeking) Bellman operator, which for episode t and timestep l is given by

$$\mathcal{B}_l^t(\tau, K)(s,a) \coloneqq \mathbb{E}^t\mu_l(s,a) + \frac{\sigma^2 + (L-l)^2}{2\tau(n_l^t(s,a) \vee 1)} + \sum_{s'\in\mathcal{S}} \mathbb{E}^t P_l(s' \mid s,a)(\tau \log \sum_{a'\in\mathcal{A}} \exp(K(s',a')/\tau)) \tag{5}$$

for inputs $\tau \geq 0$, $K \in \mathbb{R}^{|\mathcal{S}_{l+1}|\times A}$ where $n_l^t(s,a)$ is the visitation count of the agent to state-action (s,a) at timestep l before episode t and $(\cdot \vee 1) \coloneqq \max(\cdot, 1)$. Concretely we show that for any $(s,a) \in \mathcal{S}\times\mathcal{A}$

$$\mathcal{Q}_l^t(s,a) \leq \mathcal{B}_l^t(\tau, \mathcal{Q}_{l+1}^t)(s,a), \quad l = 1,\ldots,L.$$

From this fact we show that the fixed points of the optimistic Bellman operator yield a guaranteed upper bound on $\mathcal{Q}^t$, *i.e.*,

$$\Big(K_l^t = \mathcal{B}_l^t(\tau, K_{l+1}^t),\ l = 1,\ldots,L\Big) \Rightarrow \Big(K_l^t \geq \mathcal{Q}_l^t,\ l = 1,\ldots,L\Big).$$

We refer to the fixed point values as the 'K-values' since, very loosely speaking, they represent knowledge that the agent might acquire by visiting that state and action.

Let us compare the optimistic Bellman operator $\mathcal{B}^t$ to the optimal Bellman operator $\mathcal{T}^\star$ defined in (3). The first difference is that the random variables μ and P are replaced with their expectation under the posterior; in $\mathcal{T}^\star$ they are assumed to be known. Secondly, the rewards in the optimistic Bellman operator have been augmented with a bonus that depends on the visitation counts n^t. This bonus encourages the agent to visit state-actions that have been visited less frequently. Finally, the hard-max of the optimal Bellman operator has been replaced with a soft-max. Note that in the limit of zero uncertainty in the MDP (take $n_l^t(s,a) \to \infty$ for all (s,a)) we have $\mathcal{B}_l^t(0,\cdot) = \mathcal{T}_l^\star$ and we recover the optimal Bellman operator, and consequently in that case $K_l^t(s,a) = \mathcal{Q}_l^t(s,a) = Q_l^\star(s,a)$. In other words, the optimistic Bellman operator and associated K-values generalize the optimal Bellman operator and optimal Q-values to the epistemically uncertain case, and in the limit of zero uncertainty we recover the optimal quantities.

3.3 Maximum entropy policy

An agent that acts to maximize its K-values is (approximately) acting to maximize its risk-seeking utility. In the appendix we show that the policy that maximizes the expected K-values with *entropy regularization* is the natural policy to use, which is motivated by the variational description of the soft-max

$$\max_{\pi_l(s)\in\Delta_A} \left(\sum_{a\in\mathcal{A}} \pi_l(s,a) K_l^t(s,a) + \tau_t H(\pi_l(s)) \right) = \tau_t \log \sum_{a\in\mathcal{A}} \exp(K_l^t(s,a)/\tau_t) \tag{6}$$

where Δ_A is the probability simplex of dimension $A-1$ and H is entropy, *i.e.*, $H(\pi_l(s)) = -\sum_{a\in\mathcal{A}} \pi_l(s,a) \log \pi_l(s,a)$ [12]. The maximum is achieved by the Boltzmann (or Gibbs) distribution with temperature τ_t

$$\pi_l^t(s,a) \propto \exp(K_l^t(s,a)/\tau_t). \tag{7}$$

This variational principle also arises in statistical mechanics where Eq. (6) refers to the negative Helmholtz free energy and the distribution in Eq. (7) describes the probability that the system at temperature τ_t is in a particular 'state' [20]. Equation (6) is used to derive the bound on the Bayes regret (4) using the following maximal inequality

$$\mathbb{E}^t \max_a Q_l(s,a) \le \tau_t \log \sum_{a\in\mathcal{A}} \exp(\mathcal{Q}_l^t(s,a)/\tau_t) \le \tau_t \log \sum_{a\in\mathcal{A}} \exp(K_l^t(s,a)/\tau_t). \tag{8}$$

3.4 Choosing the risk-seeking parameter / temperature

The optimistic Bellman operator, K-values, and associated policy depend on the parameter τ_t. As $\tau \to 0$ the agent becomes more risk-seeking, *i.e.*, more exploratory, and by carefully controlling this parameter we ensure that the agent balances exploration and exploitation. We present two ways to do so, the first of which is to follow the schedule

$$\tau_t = \sqrt{\frac{(\sigma^2 + L^2) A |\mathcal{X}| (1 + \log t)}{4tL \log A}}. \tag{9}$$

Alternatively we find the τ_t that yields the tightest bound in the maximal inequality in (8). This turns out to be a convex optimization problem

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_{s\sim\rho} \left(\tau \log \sum_{a\in\mathcal{A}} \exp(K_1(s,a)/\tau) \right) \\ \text{subject to} \quad & K_l \ge \mathcal{B}_l^t(\tau, K_{l+1}), \quad l = 1,\dots,L, \\ & K_{L+1} \equiv 0, \end{aligned} \tag{10}$$

with variables $\tau \ge 0$ and $K \in \mathbb{R}^{|\mathcal{X}|\times A}$. This is convex jointly in τ and K since the Bellman operator is convex in both arguments and the perspective of the soft-max term in the objective is convex [9]. This generalizes the linear programming formulation of dynamic programming to the case where we

Algorithm 1 K-learning for episodic MDPs

Input: MDP $\mathcal{M} = \{\mathcal{X}, \mathcal{A}, R, P, L, \rho\}$,
for episode $t = 1, 2, \ldots$ **do**
  calculate τ_t using (9) or (10)
  set $K^t_{L+1} \equiv 0$
  compute $K^t_l = \mathcal{B}^t_l(\tau_t, K^t_{l+1})$, for $l = L, \ldots, 1$, using (5)
  execute policy $\pi^t_l(s, a) \propto \exp(K^t_l(s, a)/\tau_t)$, for $l = 1, \ldots, L$
end for

have uncertainty over the parameters that define the MDP [40, 8]. Problem (10) is an *exponential cone program* and can be solved efficiently using modern methods [27, 28, 13, 46].

Both of these schemes for choosing τ_t yield a Bayes regret bound, though in practice the τ_t obtained by solving (10) tends to perform better. K-learning is a randomized strategy since the actions are randomly sampled from a distribution. If K-learning is run with the optimal choice of temperature $\tau^\star_t$ then it is additionally a *stationary* strategy, in the same way as Thompson sampling, in that the action distribution depends solely on the posteriors and is otherwise independent of the time period [45].

3.5 Regret analysis

K-learning for episodic MDPS is presented as Algorithm 1. For the analysis of this algorithm we require some additional assumptions, beginning with the following.

Assumption 2. *The posterior over the mean value of the reward at each state-action is sub-Gaussian, and concentrates with data at least as fast as a Gaussian, i.e.,*

$$\tau \log \mathbb{E}^t \exp(\mu_l(s, a)/\tau) \leq \mathbb{E}^t \mu_l(s, a) + \frac{\sigma^2}{2\tau(n^t_l(s, a) \vee 1)}$$

for all $\tau \geq 0$ and for some $\sigma \geq 0$.

For example, if both the reward noise and the prior over μ is Gaussian then the assumption is satisfied. Similarly when the mean rewards are categorical and the prior is Dirichlet [23]. The next two assumptions are required due to the uncertainty in the transition function P.

Assumption 3. *The support of the prior ϕ is such that the mean rewards are bounded in $[0, 1]$ a.s.*

This only affects the *mean* reward, the noise in the rewards may be outside this range. Our analysis is easily extended to the case where the rewards are bounded in some $[R_{\min}, R_{\max}]$.

Assumption 4. *The prior for $P_l(\,\cdot \mid s, a)$ is Dirichlet.*

Now we have established the assumptions we require we present the main theorem.

Theorem 1. *Under assumptions 2, 3, and 4 the K-learning algorithm 1 satisfies Bayes regret bound*

$$\begin{aligned} \mathcal{BR}_\phi(T) &\leq 2\sqrt{(\sigma^2 + L^2)T|\mathcal{X}|A \log A(1 + \log T/L)} \\ &= \begin{cases} \tilde{O}(L\sqrt{TSA}) & \textit{if assumption 1 holds} \\ \tilde{O}(L^{3/2}\sqrt{TSA}) & \textit{otherwise.} \end{cases} \end{aligned} \tag{11}$$

The Bayes regret bound in the above theorem matches that of Thompson sampling up to log factors [34] and is within a factor of $\sqrt{L}$ of the known minimax lower bound of $R(T) \geq \Omega(L\sqrt{SAT})$ [19]. The full proof is included in the appendix and formalizes what we developed in this section.

Intuitively speaking K-learning propagates K-values that capture both the quality and uncertainty associated with each state and action through the MDP. The K-values are higher when the agent has high epistemic uncertainty about the true value of those states and actions, and this will make the agent more likely to take the actions that lead to those states. Over time states with high uncertainty will be visited and the uncertainty about them will be reduced, and the risk-seeking parameter τ_t is managing how visiting uncertain states is balanced with taking exploiting actions. Since the uncertainty is finite, the policy associated with the K-values eventually converges to the optimal policy for the MDP.

4 Numerical experiments

In this section we compare the performance of both the temperature scheduled and optimized temperature variants of K-learning against several other methods in the literature. We consider a small tabular MDP called *DeepSea*, adapted from [35], shown in Figure 1. This MDP can be visualized as an $L \times L$ grid where the agent starts at the top row and leftmost column. At each time-period the agent can move left or right and descends one row. The only positive reward is at the bottom right cell. In order to reach this cell the agent must take the right action every timestep. After choosing action 'left' the agent receives a reward of mean zero, and after choosing right the agent receives a reward with small negative mean. At the bottom rightmost corner of the grid the agent receives a reward with mean one. Although this is a toy example it provides a challenging 'needle in a haystack' exploration problem. Any algorithm that does exploration via a simple heuristic like local dithering will take time exponential in the depth L. Policies that perform deep exploration can learn much faster [34, 32].

In Figure 2 we show the time required to 'solve' the problem as a function of the depth of the environment, averaged over 5 seeds for each algorithm. We define the time to solve to be the first episode at which the agent has reached the rewarding state in at least 10% of the episodes so far. We compare two dithering approaches, epsilon-greedy ($\epsilon = 0.1$) and soft-Q-learning [16] ($\tau = 0.05$), against principled exploration strategies RLSVI [35], UCBVI [6], Thompson sampling [34] and two variants of K-learning, one using the τ_t schedule (9) and the other using the optimal choice $\tau_t^\star$ from solving (10). Soft Q-learning is similar to K-learning with two major differences: the temperature term is a fixed hyperparameter and there is no visitation bonus added to the rewards. These differences prevent soft Q-learning from satisfying a regret bound and typically it cannot solve difficult exploration tasks in general [31].

As expected, the two 'dithering' approaches are unable to handle the problem as the depth exceeds a small value; they fail to solve the problem within 10^5 episodes for problems larger than $L = 6$ for epsilon-greedy and $L = 14$ for soft-Q-learning. These approaches are taking time exponential in the size of the problem to solve the problem, which is seen by comparing their performance to the grey dashed line which plots 2^{L-1}. The other approaches scale more gracefully, however clearly Thompson sampling and K-learning are the most efficient. The optimal choice of K-learning appears to perform slightly better than the scheduled temperature variant, which is unsurprising since it is derived from a tighter upper bound on the regret.

In Figures 3 and 4 we show the progress of K-learning using $\tau_t^\star$ and Soft Q-learning for a single seed running on the $L = 50$ depth DeepSea environment. In the top row of each figure we show the value of each state over time, defined for K-learning as

$$\tilde{V}_l^t(s) = \tau_t \log \sum_{a \in \mathcal{A}} \exp(K_l^t(s,a)/\tau_t), \tag{12}$$

and analogously for Soft Q-learning with fixed $\tau_t = 0.05$ for all t. The bottom row shows the log of the visitation counts over time. Although both approaches start similarly they quickly diverge in their behavior. If we examine the K-learning plots, it is clear that the agent is visiting more of the bottom right corner as the episodes proceed. This is driven by the fact that the value, which incorporates the epistemic uncertainty, is high for the unvisited states. Concretely, take the $t = 300$ case; at this point the K-learning agent has not yet visited the rewarding bottom right state, but the value is very high for that region of the state space and shortly after this it reaches the reward for the first time. By the $t = 1000$ plot the agent is travelling along the diagonal to gather the reward consistently, as can be seen by the counts and the value. Contrast this to Soft Q-learning where the agent does not make it even halfway across the grid, even after $t = 1000$ episodes. This is because the soft Q-values do not capture any uncertainty about the environment, so the agent has no incentive to explore and visit new states. The only exploration that soft Q-learning is performing is merely the local dithering arising from using a Boltzmann policy with a nonzero temperature. Indeed, the soft value function barely changes in this case since the agent is consistently gathering zero reward.

5 Conclusions

In this work we endowed a reinforcement learning agent with a *risk-seeking* utility, which encourages the agent to take actions that lead to less epistemically certain states. This yields a Bayesian

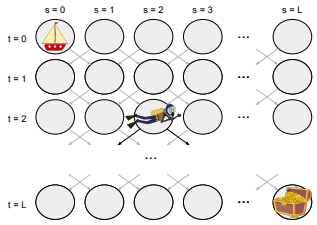


Figure 1: The DeepSea MDP

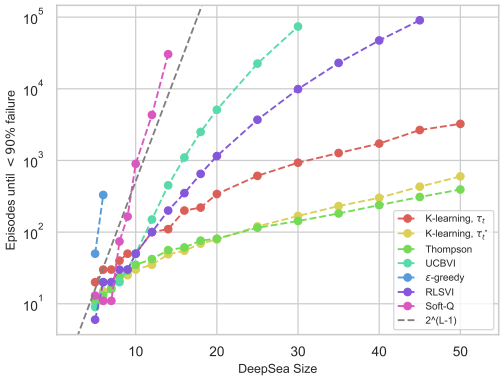
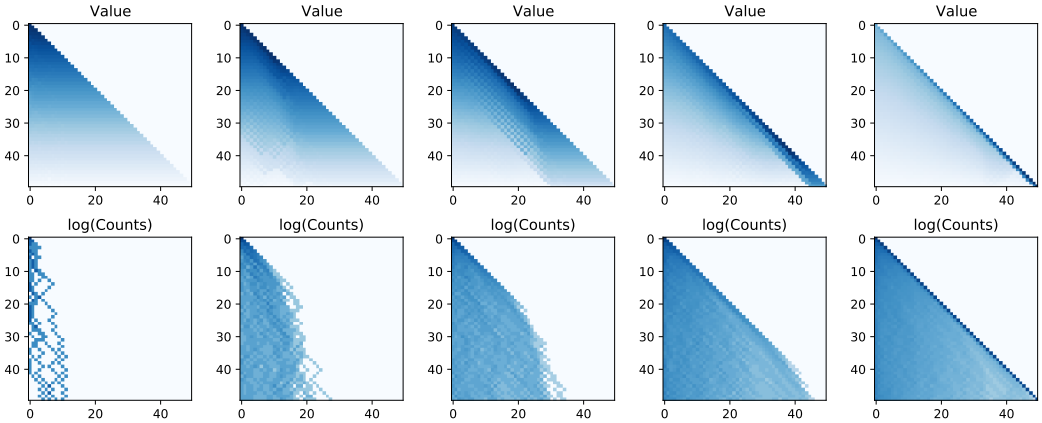


Figure 2: Learning time on DeepSea.

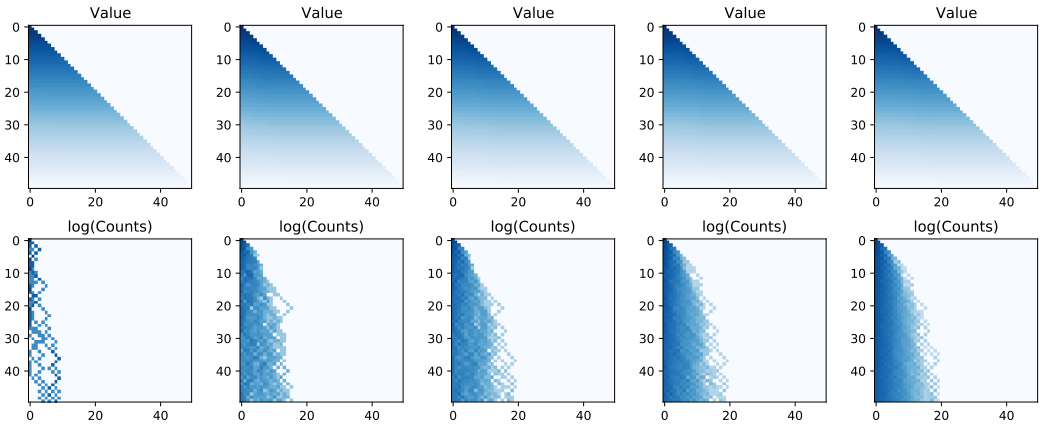


(a) $t = 5$. (b) $t = 50$. (c) $t = 100$. (d) $t = 300$. (e) $t = 1000$.

Figure 3: Value (as defined in (12)) and log visitation count for each state in DeepSea under K-learning using $\tau_t^\star$. Darker color indicates larger value.

algorithm with a bound on regret which matches the best-known regret bound for Thompson sampling up to log factors and is close to the known lower bound. We call the algorithm 'K-learning', since the 'K-values' capture the epistemic *knowledge* that the agent has about each state-action. In the limit of zero uncertainty the K-values reduce to the optimal Q-values.

Although K-learning and Thompson sampling have similar theoretical and empirical performance, K-learning has some major advantages. For one, it was recently shown that K-learning extends naturally to the case of two-player games and continues to enjoy a sub-linear regret bound, whereas Thompson sampling can suffer linear regret [29]. Secondly, Thompson sampling requires sampling from the posterior over MDPs and solving the sampled MDP exactly at each episode. The resulting policy varies significantly from episode to episode, which is necessary for Thompson sampling to achieve good performance. By contrast K-learning requires solving a Bellman equation that changes slowly as more data is accumulated, so the optimal K-values at one episode are close to the optimal K-values for the next episode, and similarly for the policies. This suggests the possibility of approximating K-learning in an online manner and making use of modern deep RL techniques such as representing the K-values using a deep neural network [25]. Exploring this extension to K-learning is left as future work.



(a) $t = 5$. (b) $t = 50$. (c) $t = 100$. (d) $t = 300$. (e) $t = 1000$.

Figure 4: Value (as defined in (12)) and log visitation count for each state in DeepSea under soft Q-learning. Darker color indicates larger value.

References

[1] A. E. Abbas. Invariant utility functions and certain equivalent transformations. *Decision Analysis*, 4(1):17–31, 2007.

[2] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. M. N. Heess, and M. Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] S. Agrawal and N. Goyal. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5):30, 2017.

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

[5] M. G. Azar, V. Gómez, and H. J. Kappen. Dynamic policy programming. *Journal of Machine Learning Research*, 13(Nov):3207–3245, 2012.

[6] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

[7] R. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[8] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific, 2005.

[9] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] N. Cesa-Bianchi, C. Gentile, G. Neu, and G. Lugosi. Boltzmann exploration done right. In *Advances in Neural Information Processing Systems*, pages 6287–6296, 2017.

[11] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[12] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[13] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[14] R. Fox, A. Pakman, and N. Tishby. Taming the noise in reinforcement learning via soft updates. arXiv preprint arXiv:1207.4708, 2015.

[15] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[16] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[17] J. Hadar and W. R. Russell. Rules for ordering uncertain prospects. *The American economic review*, 59(1):25–34, 1969.

[18] R. A. Howard. Value of information lotteries. *IEEE Transactions on Systems Science and Cybernetics*, 3(1):54–60, 1967.

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

[20] E. T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[21] Z. C. Lipton, J. Gao, L. Li, X. Li, F. Ahmed, and L. Deng. Efficient exploration for dialogue policy learning with BBQ networks & replay buffer spiking. *arXiv preprint arXiv:1608.05081*, 2016.

[22] X. Lu, B. V. Roy, V. Dwaracherla, M. Ibrahimi, I. Osband, and Z. Wen. Reinforcement learning, bit by bit, 2021.

[23] O. Marchal and J. Arbel. On the sub-Gaussianity of the Beta and Dirichlet distributions. arXiv preprint arXiv:1705.00048, 2017.

[24] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015.

[26] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2772–2782, 2017.

[27] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016.

[28] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. SCS: Splitting conic solver, version 2.0.2. `https://github.com/cvxgrp/scs`, Nov. 2017.

[29] B. O'Donoghue, T. Lattimore, and I. Osband. Stochastic matrix games with bandit feedback. *arXiv preprint arXiv:2006.05145*, 2020.

[30] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih. Combining policy gradient and Q-learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[31] B. O'Donoghue, I. Osband, and C. Ionescu. Making sense of reinforcement learning and probabilistic inference. In *International Conference on Learning Representations (ICLR)*, 2020.

[32] B. O'Donoghue, I. Osband, R. Munos, and V. Mnih. The uncertainty Bellman equation and exploration. *arXiv preprint arXiv:1709.05380*, 2017.

[33] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped DQN. In *Advances In Neural Information Processing Systems*, pages 4026–4034, 2016.

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

[35] I. Osband, D. Russo, Z. Wen, and B. Van Roy. Deep exploration via randomized value functions. *arXiv preprint arXiv:1703.07608*, 2017.

[36] I. Osband and B. Van Roy. Gaussian-Dirichlet posterior dominance in sequential learning. *arXiv preprint arXiv:1702.04126*, 2017.

[37] I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

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

[39] J. Pfanzag. A general theory of measurement applications to utility. *Naval Research Logistics (NRL)*, 6(4):283–294, 1959.

[40] M. L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[41] H. Raiffa. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Addison Wesley, 1968.

[42] P. H. Richemond and B. Maginnis. A short variational proof of equivalence between policy gradients and soft Q learning. *arXiv preprint arXiv:1712.08650*, 2017.

[43] D. Russo and B. Van Roy. An information-theoretic analysis of Thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[44] D. Russo and B. Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

[45] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

[46] S. A. Serrano. *Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone*. PhD thesis, Stanford University, 2015.

[47] M. Strens. A Bayesian framework for reinforcement learning. In *ICML*, pages 943–950, 2000.

[48] R. Sutton and A. Barto. *Reinforcement Learning: an Introduction*. MIT Press, 1998.

[49] 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.

[50] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

[51] R. J. Williams and J. Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[52] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] This paper is a purely theoretical work with little societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We included a description of the data generation process for the simulations we ran.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] These experiments involved no training on external data.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] I produced Fig 2 with error bars, but it looked very cluttered and obscured the message without providing any additional insight.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Included in appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]