Making Sense of Reinforcement Learning and Probabilistic Inference

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning (RL) combines a control problem with statistical estimation: the system dynamics are not known to the agent, but can be learned through experience. A recent line of research casts 'RL as inference' and suggests a particular framework to generalize the RL problem as probabilistic inference. Our paper surfaces key shortcomings in that approach, and clarifies the sense in which RL can be coherently cast as an inference problem. In particular, an RL agent must consider the effects of its actions upon future rewards and observations: the exploration-exploitation tradeoff. In all but the most simple settings, the resulting inference is computationally intractable so that practical RL algorithms must resort to approximation. We show that the popular 'RL as inference' approximation can perform poorly in even the simplest settings. Despite this, we demonstrate that with a small modification the RL as inference framework can provably perform well, and we connect the resulting algorithm with Thompson sampling and the recently proposed K-learning algorithm.

1 Introduction

Probabilistic inference is a procedure of making sense of uncertain data using Bayes' rule. The optimal control problem is to take actions in a known system in order to maximize the cumulative rewards through time. Probabilistic graphical models (PGMs) offer a coherent and flexible language to specify causal relationships, for which a rich literature of learning and inference techniques have developed (Koller & Friedman, 2009). Although control dynamics might also be encoded as a PGM, the relationship between action planning and probabilistic inference is not immediately clear. For inference, it is typically enough to specify the system and pose the question, and the objectives for learning emerge automatically. In control, the system and objectives are trivially known, but the question of how to approach a solution may remain extremely complex (Bertsekas, 2005).

Perhaps surprisingly, there is a deep sense in which inference and control can represent a dual view of the same problem. This relationship is most clearly stated in the case of linear quadratic systems, where the Ricatti equations relate the optimal control policy in terms of the system dynamics (Welch et al., 1995). In fact, this connection extends to a wide range of systems, where control tasks can be related to a dual inference problem through rewards as exponentiated probabilities in a distinct, but coupled, PGM (Todorov, 2007; 2008). A great benefit of this connection is that it can allow the tools of inference to make progress in control problems, and vice-versa. In both cases the connections provide new insights, inspire new algorithms and enrich our understanding (Toussaint & Storkey, 2006; Ziebart et al., 2008; Kappen et al., 2012).

Reinforcement learning (RL) is the problem of learning to control an unknown system (Sutton & Barto, 2018). Like the control setting, an RL agent should take actions to maximize its cumulative rewards through time. Like the inference problem, the agent is initially uncertain of the system dynamics, but can learn through the states and actions it observes. This leads to a fundamental tradeoff: the agent may be able to improve its understanding through exploring poorly-understood states and actions, but it may be able to attain higher immediate reward through exploiting its existing knowledge (Kearns & Singh, 2002). In many ways, RL combines control and inference into a general framework for decision making under uncertainty. Although there has been ongoing research in this area for many decades, there has been a recent explosion of interest as RL techniques have made high-profile breakthroughs in grand challenges of artificial intelligence research (Mnih et al., 2013; Silver et al., 2016).

A popular line of research has sought to cast 'RL as inference', mirroring the dual relationship for control in known systems. This approach is most clearly stated in the tutorial and review of Levine (2018), and provides a key reference for research in this field. It suggests that a *generalization* of the RL problem can be cast as probabilistic inference through inference over exponentiated rewards, in a continuation of previous work in optimal control (Todorov, 2009). This perspective promises several benefits: a probabilistic perspective on rewards, the ability to apply powerful inference algorithms to solve RL problems and a natural exploration strategy. In this paper we will outline important ways in which this perspective is incomplete. These shortcomings ultimately result in algorithms that can perform poorly in even very simple decision problems.

In this paper we revisit an alternative framing of 'RL as inference'. In fact, we show that the *original* RL problem was already an inference problem all along.[1] Importantly, this inference problem includes inference over the agent's future actions and observations. Of course, this perspective is not new, and has long been known as simply the Bayes-optimal solution, see, *e.g.*, Ghavamzadeh et al. (2015). The problem is that, due to the exponential lookahead, this inference problem is fundamentally intractable for all but the simplest problems Gittins (1979). For this reason, RL research focuses on computationally efficient approaches that maintain a level of statistical efficiency (Osband et al., 2017).

We provide a review of the RL problem in Section 2, together with a simple and coherent framing of 'RL as inference'. In Section 3 we present three approximations to the intractable Bayes-optimal policy. We begin with the celebrated Thompson sampling algorithm, then we review the popular 'RL as inference' framing, as presented by Levine (2018), and highlight some clear and simple shortcomings in this approach. Finally, we review K-learning (O'Donoghue, 2018), which we reinterpret as a modification to the RL as inference framework that provides a principled approach to the statistical inference problem, as well as a presenting a relationship with Thompson sampling. In Section 4 we present computational studies that support our claims.

2 Reinforcement learning

We consider the problem of an agent taking actions in an unknown environment in order to maximize cumulative rewards through time. For simplicity, this paper will model the environment as a finite horizon, discrete Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, H, \rho)$.[2] Here $\mathcal{S} = \{1, .., S\}$ is the state space, $\mathcal{A} = \{1, .., A\}$ is the action space and each episode is of fixed length $H \in \mathbb{N}$. Each episode $\ell \in \mathbb{N}$ begins with state $s_0 \sim \rho$ then for timesteps $h = 0, .., H-1$ the agent selects action a_h, observes transition s_{h+1} with probability $\mathcal{P}(s_{h+1}, s_h, a_h) \in [0,1]$ and receives reward $r_{h+1} \sim \mathcal{R}(s_h, a_h)$, where we denote by $\mu(s_h, a_h) = \mathbb{E} r_{h+1}$ the mean reward. We define a policy π to be a mapping from $\mathcal{S}$ to distributions over $\mathcal{A}$ and write Π for the space of all policies. For any timestep $t = (\ell, h)$, we define $\mathcal{F}_t = (s_0^0, a_0^0, r_1^0, .., s_{h-1}, a_{h-1}, r_h)$ to be the sequence of observations made before time t. An RL algorithm maps histories to policies $\pi_t = \mathrm{alg}(\mathcal{S}, \mathcal{A}, \mathcal{F}_t)$.

Our goal in the design of RL algorithms is to obtain good performance (cumulative rewards) for an unknown $M \in \mathcal{M}$, where $\mathcal{M}$ is some *family* of possible environments. Note that this is a different problem from typical 'optimal control', that seeks to optimize performance for one particular known MDP M; although you might still fruitfully apply an RL *algorithm* to solve problems of that type. For any environment M and any policy π we can define the action-value function,

$$Q_h^{M,\pi}(s,a) = \mathbb{E}_{\pi,M}\left[\sum_{j=h+1}^{H} r_j \mid s_h = s, a_h = a\right]. \quad (1)$$

Where the expectation in (1) is taken with respect to the action selection a_j for $j > h$ from the policy π and evolution of the fixed MDP M. We define the value function $V_h^{M,\pi}(s) = \mathbb{E}_{\alpha\sim\pi} Q_h^{M,\pi}(s,\alpha)$ and write $V_h^{M,\star}(s) = \max_{\pi\in\Pi} V_h^{M,\pi}(s)$ for the optimal value over policies.

[1] Note that, unlike control, connecting RL with inference will not involve a separate 'dual' problem.

[2] This choice is for clarity; continuous or partially-observed environments do not alter our narrative.

In order to compare algorithm performance across different environments, it is natural to normalize in terms of the *regret*, or shortfall in cumulative rewards relative to the optimal value,

$$\mathrm{Regret}(M, \mathrm{alg}, L) = \mathbb{E}_{M,\mathrm{alg}}\left[\sum_{\ell=1}^{L}\left(V_0^{M,\star}(s_0^\ell) - \sum_{h=1}^{H} r_h^\ell\right)\right]. \tag{2}$$

This quantity depends on the unknown MDP M, but the expectations are taken with respect to the dynamics of M and the learning algorithm alg. For any particular MDP M, the optimal regret of zero can be attained by the non-learning algorithm alg_M that returns the optimal policy for M.

In order to assess the quality of a reinforcement learning algorithm, which is designed to work across some *family* of $M \in \mathcal{M}$, we need some method to condense performance over a set to a single number. There are two main approaches to this:

$$\mathrm{BayesRegret}(\phi, \mathrm{alg}, L) = \mathbb{E}_{M\sim\phi}\mathrm{Regret}(M, \mathrm{alg}, L), \tag{3}$$

$$\mathrm{MinimaxRegret}(\mathcal{M}, \mathrm{alg}, L) = \max_{M\in\mathcal{M}} \mathrm{Regret}(M, \mathrm{alg}, L), \tag{4}$$

where ϕ is a prior over the famliy $\mathcal{M}$. These differing objectives are often framed as Bayesian (average-case) (3) and frequentist (worst-case) (4) RL. Although these two settings are typically studied in isolation, it should be clear that they are intimately related through the choice of $\mathcal{M}$ and ϕ. Our next section will investigate what it would mean to 'solve' the RL problem. Importantly, we show that both frequentist and Bayesian perspectives already amount to a problem in probabilistic inference, without the need for additional re-interpretation.

2.1 Solving the RL problem through probabilistic inference

If you want to 'solve' the RL problem, then formally the objective is clear: find the RL algorithm that minimizes your chosen objective, (3) or (4). To anchor our discussion, we introduce a simple decision problem designed to highlight some key aspects of reinforcement learning. We will revisit this problem setting as we discuss approximations to the optimal policy.

Problem 1 (One unknown action)**.** *Fix $N \in \mathbb{N} \geq 3, \epsilon > 0$ and define $\mathcal{M}_{N,\epsilon} = \{M^+_{N,\epsilon}, M^-_{N,\epsilon}\}$. Both M^+ and M^- share $\mathcal{S} = \{1\}, H = 1$ and $\mathcal{A} = \{1, .., N\}$; they only differ through their rewards:*

$$\mathcal{R}^+(1) = 1, \quad \mathcal{R}^+(2) = +2, \quad \mathcal{R}^+(a) = 1-\epsilon \textit{ for } a = 3, .., N,$$
$$\mathcal{R}^-(1) = 1, \quad \mathcal{R}^-(2) = -2, \quad \mathcal{R}^-(a) = 1-\epsilon \textit{ for } a = 3, .., N.$$

Where $\mathcal{R}(a) = x \in \mathbb{R}$ is a shorthand for deterministic reward of x when choosing action a.

Problem 1 is extremely simple, it involves no generalization and no long-term consequences: it is an independent bandit problem with only one unknown action. For *known* M^+, M^- the optimal policy is trivial: choose $a_t = 2$ in M^+ and $a_t = 1$ in M^- for all t. An RL agent faced with *unknown* $M \in \mathcal{M}$ should attempt to optimize the RL objectives (3) or (4). Unusually, and only because Problem 1 is so simple, we can actually compute the *optimal* solutions to both in terms of L (the total number of episodes) and $\phi = (p^+, p^-)$ where $p^+ = \mathbb{P}(M = M^+)$, the probability of being in M^+.

For $L > 3$ an optimal *minimax* RL algorithm is to first choose $a_0 = 2$ and observe r_1. If $r_1 = 2$ then you know you are in M^+ so pick $a_t = 2$ for all $t = 1, 2..$, for $\mathrm{Regret}(L) = 0$ for all $L = 1, 2, ...$ If $r_1 = -2$ then you know you are in M^- so pick $a_t = 1$ for all $t = 1, 2..$, for $\mathrm{Regret}(L) = 3$ for all $L = 1, 2, ...$ The minimax regret of this algorithm is 3, which cannot be bested by any algorithm.

Actually, the same RL algorithm is also *Bayes*-optimal for any $\phi = (p^+, p^-)$ provided $p^+L > 3$. This relationship is not a coincidence. All admissible solutions to the minimax problem (4) are in given by solutions to the average-case (3) for some 'worst-case' prior $\tilde{\phi}$ (Wald, 1950). As such, for ease of exposition, our discussion will focus on the Bayesian (or average-case) setting. However, readers should understand that the same arguments apply to the minimax objective.

In Problem 1, the key probabilistic inference the agent must consider is the effects of it own *actions* upon the future rewards, or whether it has chosen action 2. More generally, where actions are independent and episode length $H = 1$, the optimal RL algorithm can be computed via Gittins indices, but these are very much the exception (Gittins, 1979). In problems with generalization or long-term consequences, computing the Bayes-optimal solution is computationally intractable. One example of an algorithm that converges to Bayes-optimal solution in the limit of infinite computation is given by Bayes-adaptive Monte-Carlo Planning (Guez et al., 2012). The problem is that, even for very simple problems, the lookahead tree of interactions between actions, observations and algorithmic updates grows exponentially in the search depth (Strehl et al., 2006). Worse still, direct computational approximations to the Bayes-optimal solution can fail exponentially badly should they fall short of the required computation (Munos, 2014). As a result, research in reinforcement learning amounts to trying to find computationally tractable approximations to (3), that maintain some degree of statistical efficiency.

3 Approximations for computational and statistical efficiency

The exponential explosion of future actions and observations means the solving for Bayes-optimal solution is computationally intractable. To counter this, most computationally efficient approaches to RL simplify the problem at time t to only consider inference over the data $\mathcal{F}_t$ that has been gathered prior to time t. The most common family of these algorithms are 'certainty equivalent' (under an identity utility): they take a point estimate for their best guess of the environment $\hat{M}$, and try to optimize their control given these estimates $V^{\hat{M},*}$. Typically, these algorithms are used in conjunction with some dithering scheme for random action selection (*e.g.*, epsilon-greedy), to mitigate premature and suboptimal convergence (Watkins, 1989). However, since these algorithms do not prioritize their exploration, they may take exponentially long to find the optimal policy (Osband et al., 2014).

In order for an RL algorithm to be statistically efficient, it must consider the value of information. To do this, an agent must first maintain some notion of epistemic uncertainty, so that it can direct its exploration towards states and actions that it does not understand well (O'Donoghue et al., 2018). Here again, probabilistic inference finds a natural home in RL: we should build up posterior estimates for the unknown problem parameters, and use this *distribution* to drive efficient exploration.[3]

3.1 Thompson sampling

One of the oldest heuristics for balancing exploration with exploitation is given by Thompson sampling, or probability matching (Thompson, 1933). Each episode, Thompson sampling (TS) randomly selects a policy according to the probability it is the optimal policy, conditioned upon the data seen prior to that episode. Thompson sampling is a simple and effective method that successfully balances exploration with exploitation (Russo et al., 2018).

Implementing Thompson sampling amounts to an inference problem at each episode. For each s, a, h define the binary random variable $\mathcal{O}_h(s,a)$ where $\mathcal{O}_h(s,a) = 1$ denotes the event that action a is optimal for state s in timestep h.[4] The TS policy for episode L is thus given by the inference problem,

$$\pi^{\mathrm{TS}} \sim \mathbb{P}(\mathcal{O} \mid \mathcal{F}_L), \tag{5}$$

where $\mathbb{P}(\mathcal{O} \mid \mathcal{F}_L)$ is the *joint* probability over all the binary optimality variables. To understand how Thompson sampling guides exploration let us consider its performance in Problem 1 when implemented with a uniform prior $\phi = (\frac{1}{2}, \frac{1}{2})$. In the first timestep the agent samples $M_0 \sim \phi$. If it samples M^+ it will choose action $a_0 = 2$ and learn the true system dynamics, choosing the optimal arm thereafter. If it samples M^- it will choose action $a_0 = 1$ and repeat the identical decision in the next timestep. Note that this procedure achieves BayesRegret 2.5 according to ϕ, but *also* minimax regret 3, which matches the optimal minimax performance despite its uniform prior.

Recent interest in TS was kindled by strong empirical performance in bandit tasks (Chapelle & Li, 2011). Following work has shown that this algorithm satisfies strong Bayesian regret bounds close to the lower bounds for MDPs (Osband & Van Roy, 2017; 2016). However, although much simpler than the Bayes-optimal solution, the inference problem in (5) can still be prohibitively expensive. Table 1 describes on approach to sampling from (5) implicitly by maintaining an explicit model over

[3] For the purposes of this paper, we will focus on *optimistic* approaches to exploration, although more sophisticated information-seeking approaches merit investigation in future work (Russo & Van Roy, 2014).

[4] For the problem definition in Section 2 there is always a deterministic optimal policy for M.

MDP parameters. This algorithm can be computationally intractable as the MDP becomes large and so attempts to scale Thompson sampling to complex systems have focused on *approximate* posterior samples via randomized value functions, but it is not yet clear under which settings these approximations should be expected to perform well (Osband et al., 2017). As we look for practical, scalable approaches to posterior inference one promising (and popular) approach is known commonly as 'RL as inference'.

Table 1: Model-based Thompson sampling.

Before episode L	Sample $M_L = (\mathcal{S}, \mathcal{A}, \mathcal{R}^L, \mathcal{P}^L, H, \rho) \sim \phi \mid \mathcal{F}_L$
Bellman equation	$Q_h^L(s,a) = \mu^L(s,a) + \sum_{s'} \mathcal{P}^L(s', s, a) V_{h+1}^L(s')$ $V_h^L(s) = \max_a Q_h^L(s,a)$
Policy	$\pi_h^{\mathrm{TS}}(s,a) \in \operatorname{argmax} Q_h^L(s,a)$

3.2 The 'RL as inference' framework and its limitations

The computational challenges of Thompson sampling suggest an approximate algorithm that replaces (5) with a parametric distribution suitable for expedient computation. It is possible to view the algorithms of the 'RL as inference' approach in this light (Rawlik et al., 2013; Todorov, 2009; Toussaint, 2009; Deisenroth et al., 2013); see Levine (2018) for a recent survey. These algorithms choose to model the probability of optimality according to,

$$\tilde{\mathbb{P}}(\mathcal{O}_h(s,a) | \tau_h(s,a)) \propto \exp\left(\sum_{(s',a') \in \tau_h(s,a)} \beta \mathbb{E}^L \mu(s',a') \right). \qquad (6)$$

for some $\beta > 0$, where $\tau_h(s,a)$ is a trajectory (a sequence of state-action pairs) starting from (s,a) at timestep h, and where $\mathbb{E}^L$ denotes the expectation under the posterior at episode L. With this potential in place one can perform Bayesian inference over these unobserved 'optimality' variables, obtaining posteriors over the policy or other variables of interest. (This optimality variable is slightly different to the presentation in Levine (2018) but ultimately it produces the same family of algorithms; we provide such a derivation in the appendix for completeness).

Applying inference procedures to (6) leads naturally to RL algorithms with some 'soft' Bellman updates, and added entropy regularization. We describe the general structure of these algorithms in Table 2. These algorithmic connections can help reveal connections to policy gradient, actor-critic, and maximum entropy RL methods (Mnih et al., 2016; O'Donoghue et al., 2017; Haarnoja et al., 2017; 2018; Eysenbach et al., 2018). The problem is that this resultant 'posterior' from (6) does not generally bear any close relationship to the agent's epistemic probability that (s,a,h) is optimal.

Table 2: Soft Q-learning.

Bellman equation	$\tilde{Q}_h(s,a) = \mathbb{E}^L \mu(s,a) + \sum_{s'} \mathbb{E}^L \mathcal{P}(s', s, a) \tilde{V}_{h+1}(s')$ $\tilde{V}_h(s) = \beta^{-1} \log \sum_a \exp \beta \tilde{Q}_h(s,a)$
Policy	$\pi_h^{\mathrm{SQ}}(s,a) \propto \exp \beta \tilde{Q}_h(s,a)$

To understand how 'RL as inference' guides decision making, let us consider its performance in Problem 1. Practical implementations of 'RL as inference' estimate $\mathbb{E}^L \mu$ through observations. For N large, and without prior guidance, the agent in then extremely unlikely to select action $a_t = 2$ and so resolve its epistemic uncertainty. Even for an informed prior $\phi = (\frac{1}{2}, \frac{1}{2})$ action selection according to the exploration strategy of Boltzmann dithering is exponentially unlikely to sample action 2 for which $\mathbb{E}^L \mu(2) = 0$ (Levine, 2018; Cesa-Bianchi et al., 2017). This is because the $N-1$ actions with $\mathbb{E}^L \mu \geq 1 - \epsilon$ are much more likely to be sampled.

This problem is the same problem that afflicts most dithering approaches to exploration. ‘RL as inference’ as a framework does not incorporate an agents epistemic uncertainty, and so can lead to poor policies for even simple problems. While (6) allows the construction of a dual ‘posterior distribution’, this distribution does not generally bear any relation to the typical posterior an agent might compute conditioned upon the data it has gathered according to (5). Despite these shortcomings RL as inference has inspired many interesting and novel techniques, as well as delivered algorithms with good performance on problems where exploration is not the bottleneck (Gregor et al., 2016). However, due to the use of language about ‘optimality’ and ‘posterior inference’ *etc.*, it may come as a surprise to some that this framework does not truly tackle the Bayesian RL problem. Indeed, algorithms using ‘RL as inference’ can perform very poorly on problems where accurate uncertainty quantification is crucial to performance. We hope that this paper sheds some light on the topic.

3.3 Making sense of ‘RL as Inference’ via K-learning

In this section we suggest a subtle alteration to the ‘RL as inference’ framework that develops a coherent notion of optimality. The K-learning algorithm was originally introduced through a risk-seeking exponential utility (O’Donoghue, 2018). In this paper we re-derive this algorithm as a principled approximate inference procedure with clear connections to Thompson sampling, and highlight its similarities to the ‘RL as inference’ framework. We believe that this may offer a road towards combining the respective strengths of Thompson sampling and ‘RL as inference’ frameworks. First, consider the following approximate conditional optimality probability at state-action (s, a):

$$\tilde{\mathbb{P}}(\mathcal{O}_h(s,a)|Q_h^{M,\star}(s,a)) \propto \exp \beta Q_h^{M,\star}(s,a), \tag{7}$$

for some $\beta > 0$, and note that this is conditioned on the random variable $Q_h^{M,\star}(s,a)$. We can marginalize over possible Q-values yielding

$$\tilde{\mathbb{P}}(\mathcal{O}_h(s,a)) = \int \tilde{\mathbb{P}}(\mathcal{O}_h(s,a)|Q_h^{M,\star}(s,a)) d\mathbb{P}(Q_h^{M,\star}(s,a)) \propto \exp G_h^Q(s,a,\beta), \tag{8}$$

where $G_h^Q(s,a,\cdot)$ is the cumulant generating function of the random variable $Q_h^{M,\star}(s,a)$ (Kendall, 1946). Equations (6) and (7) are closely linked, but there is a crucial difference. The K-learning expectation (7) is with respect to the *posterior* over $Q_h^{M,\star}(s,a)$ to give a parametric approximation to the probability of optimality. This includes the epistemic uncertainty explicitly and so results in more uncertain actions being taken more frequently.

Table 3: K-learning.

Before episode L	Calculate $\beta_L = \rho\sqrt{L}$
Bellman equation	$K_h(s,a) = \beta_L^{-1} G^{\mu}(s,a,\beta_L) + \sum_{s_{h+1}} \mathbb{E}^L \mathcal{P}(s',s,a) V_{h+1}^{\mathrm{KL}}(s')$ $V_h^{\mathrm{KL}}(s) = \beta_L^{-1} \log \sum_a \exp \beta_L K_h(s,a)$
Policy	$\pi_h^{\mathrm{KL}}(s,a) \propto \exp \beta_L K_h(s,a)$

Following the policy in (8) requires computation of the cumulant generating function, which is non-trivial. O’Donoghue (2018) showed that the K-values that are the solution to a particular Bellman equation produced a guaranteed upper bound on the cumulant generating function at β, which by Jensen’s inequality is optimistic for the expected Q-value,

$$K_h(s,a) \geq \beta_t^{-1} G_h^Q(s,a)(\beta_t) \geq \mathbb{E}^L Q_h^{M,\star}(s,a). \tag{9}$$

Following a Boltzmann policy over these K-values satisfies a Bayesian regret bound which matches the current best bound for Thompson sampling in tabular MDPs up to logarithmic factors. We summarize the K-learning algorithm in table (3), where $\rho > 0$ is a constant.

Comparing tables 2 and 3 it is clear that soft Q-learning and K-learning share some similarities: They both use a ‘soft’ value function and Boltzmann policies. However, the differences are important.

Firstly, K-learning has an explicit schedule for the inverse temperature parameter $\beta_L = \rho\sqrt{L}$, and secondly it includes a 'bonus' based on visit counts in the reward signal. These two relatively small changes make K-learning a principled exploration and inference strategy.

To understand how K-learning drives exploration, consider its performance on Problem 1. Since this is a bandit problem we can compute the cumulant generating functions for each arm and then use the policy given by (8). For any non-trivial prior and choice of $\beta > 0$ the cumulant generating function is optimistic for arm 2 which results in the policy selecting arm 2 more frequently, thereby resolving its epistemic uncertainty. As $\beta \to \infty$ K-learning will converge on pulling arm 2 with probability one. In contrast to Soft Q-learning, actions $a = 3, .., N$ are exponentially *unlikely* to be selected as the exploration parameter β grows.

3.3.1 The K-learning policy approximates the Thompson sampling policy

Since K-learning can be viewed as approximating the posterior probability of optimality of each action one may ask how similar are the two distributions. A natural way to measure this similarity is the KL divergence between the distributions,

$$D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid \pi_h(s)^{\mathrm{KL}}) = \sum_a \mathbb{P}(\mathcal{O}_h(s,a)) \log(\mathbb{P}(\mathcal{O}_h(s,a))/\pi_h^{\mathrm{KL}}(s,a)).$$

Note that this is different to the usual notion of distance that is taken in variational Bayesian methods, which would typically reverse the order of the arguments in the KL divergence (Blundell et al., 2015). However, in RL that 'direction' is not appropriate: a distribution minimizing $D_{KL}(q \mid\mid \mathbb{P}(\mathcal{O}_h(s)))$ may put zero probability on regions of support of $\mathbb{P}(\mathcal{O}_h(s))$. This means an action with non-zero probability of being optimal might *never* be taken. On the other hand a policy minimizing $D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid q)$ must assign a non-zero probability to every action that has a non-zero probability of being optimal, or incur an infinite KL divergence penalty. With this characterization in mind, our next result links the policies of K-learning to Thompson sampling.

Theorem 1. *The K-learning value function V^{KL} and policy π^{KL} defined in table 3 satisfy the following bound at every state $s \in \mathcal{S}$ and $h = 0, \ldots H$:*

$$V_h^{\mathrm{KL}}(s) \geq \mathbb{E}V_h^{M,\star}(s) + \beta^{-1} D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid \pi_h(s)^{\mathrm{KL}}). \tag{10}$$

We defer the proof to Appendix 5.2. This theorem tells us that the distance between the true probability of optimality, *i.e.*, the Thompson sampling policy, and the K-learning policy is bounded for any choice of $\beta < \infty$. In other words, if there is an action that might be optimal then K-learning will eventually take that action.

3.4 Why is 'RL as Inference' so popular?

The sections above outline some surprising ways that the 'RL as inference' framework can drive suboptimal behaviour in even simple domains. The question remains, why do so many popular and effective algorithms lie within this class? The first, and most important point, is that these algorithms can also perform extremely well in domains where efficient exploration is not a bottleneck. Further they are often easy to implement and amenable to function approximation (Peters et al., 2010; Kober & Peters, 2009; Abdolmaleki et al., 2018). Our discussion of K-learning in Section 3.3 shows that a relatively simple fix to this problem formulation can result in a framing of RL as inference that maintains a coherent notion of optimality. Computational results show that, in tabular domains, K-learning can be competitive with, or even outperform Thompson sampling strategies, but extending these results to large-scale domains with generalization is an open question (O'Donoghue, 2018; Osband et al., 2017).

The other observation is that the 'RL as inference' can provide useful insights to the structure of particular *algorithms* for RL. It is valid to note that, under certain conditions, following policy gradient is equivalent to a dual inference problem where the 'probabilities' play the role of dummy variables, but are not supposed to represent the probability of optimality in the RL problem. In this light, Levine (2018) presents the inference framework as a way to generalize a wide range of state of the art RL algorithms. However, when taking this view, you should remember that this inference duality is limited to certain RL algorithms, and without some modifications (e.g. Section 3.3) this perspective is in danger of overlooking important aspects of the RL problem.

4 Computational experiments

4.1 One unknown action (Problem 1)

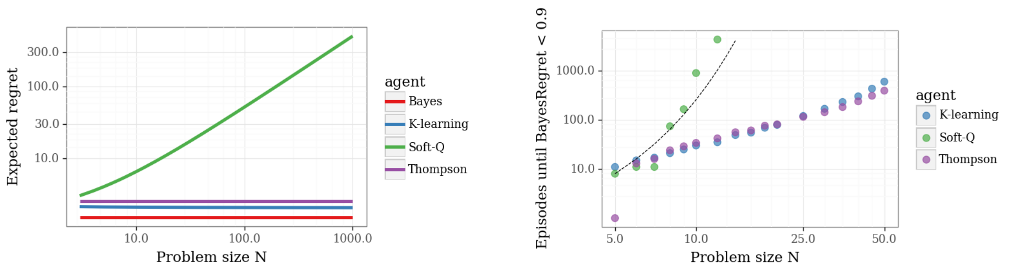
Consider the environment of Problem 1 with uniform prior $\phi = (\frac{1}{2}, \frac{1}{2})$. We fix $\epsilon = 1e - 3$ and consider how the Bayesian regret varies with $N > 3$. Figure 1a compares how the regret scales for Bayes-optimal (1.5), Thompson sampling (2.5), K-learning (≤ 2.2) and Soft Q-learning (which grows linearly in N for the optimal $\beta \to 0$, but would typically grow exponentially for $\beta > 0$). This highlights that, even in a simple problem, there can be great value in considering the value of information.

4.2 'DeepSea' problem with tabular representation

Our next set of experiments considers the the 'DeepSea' MDPs as introduced by Osband et al. (2017). At a high level this problem represents a 'needle in a haystack', and is designed to require efficient exploration, the complexity of which grows with the problem size $N \in \mathbb{N}$. DeepSea is a scalable variant of the 'chain MDPs' long considered in exploration research (Jaksch et al., 2010). Algorithms that do not perform *deep exploration* will take an exponential number of episodes to learn the optimal policy, but those that prioritize informative states and action can learn in polynomial time of N. Figure 1b presents results for Thompson sampling (PSRL), K-learning and Soft-Q learning with identical priors and posterior updates (Gaussian rewards, Dirichlet transitions). As expected, Thompson sampling and K-learning scale gracefully to large domains, Soft-Q does not.

4.3 Behaviour suite for Reinforcement Learning

So far our results have purely focused on the tabular setting, but the main focus of 'RL as inference' is for scalable algorithms that work with generalization. To show that the problems we have identified are not fixed by implementing Soft-Q learning with neural networks we evaluate this algorithm on `bsuite`: a suite of benchmark tasks designed to highlight key issues in RL (Osband et al., 2019). As expected, the core issues of uncertainty and exploration remain poor for Soft-Q even when paired with deep neural architectures. We push these results to Appendix 6.



(a) Regret scaling on Problem 1

(b) Learning time on DeepSea (dashed line 2^{N-2}).

Figure 1: Ignoring the value of information can lead to exponentially slower learning.

5 Conclusion

This paper aims to make sense of reinforcement learning and probabilistic inference. We review the reinforcement learning problem and show that this problem of optimal learning already combined the problems of control and inference. As we highlight this connection, we also clarify some potentially confusing details in the popular 'RL as inference' framework. We show that, since this problem formulation ignores the role of epistemic uncertainty, that algorithms derived from that framework can perform poorly on even simple tasks. Importantly, we also offer a way forward, to reconcile the views of RL and inference in a way that maintains the best pieces of both. In particular, we show that a simple variant to the RL as inference framework (K-learning) can incorporate uncertainty estimates to drive efficient exploration. We support our claims with a series of simple didactic experiments. We leave the crucial questions of how to scale these insights up to large complex domains for future work.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

Søren Asmussen and Peter W Glynn. *Stochastic simulation: Algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.

Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific, 2005.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Nicolò Cesa-Bianchi, Claudio Gentile, Gergely Neu, and Gabor Lugosi. Boltzmann exploration done right. In *Advances in Neural Information Processing Systems*, pp. 6287–6296, 2017.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Arthur Guez, David Silver, and Peter Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, pp. 1025–1033, 2012.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

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

Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

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

Maurice George Kendall. *The advanced theory of statistics.* Charles Griffin and Co., Ltd., London, 1946.

Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pp. 849–856, 2009.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015. URL `http://dx.doi.org/10.1038/nature14236`.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1928–1937, 2016.

Rémi Munos. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

Brendan O'Donoghue. Variational Bayesian reinforcement learning with regret bounds. *arXiv preprint arXiv:1807.09647*, 2018.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and Q-learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty Bellman equation and exploration. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

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

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

Ian Osband, Daniel Russo, Zheng Wen, and Benjamin Van Roy. Deep exploration via randomized value functions. *arXiv preprint arXiv:1703.07608*, 2017.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, , Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepezvari, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado Van Hasselt. Behaviour suite for reinforcement learning. 2019.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*. Atlanta, 2010.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pp. 1583–1591, 2014.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 881–888. ACM, 2006.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

William 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.

Emanuel Todorov. Linearly-solvable markov decision problems. In *Advances in neural information processing systems*, pp. 1369–1376, 2007.

Emanuel Todorov. General duality between optimal control and estimation. In *2008 47th IEEE Conference on Decision and Control*, pp. 4286–4292. IEEE, 2008.

Emanuel Todorov. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483, 2009.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1049–1056. ACM, 2009.

Marc Toussaint and Amos Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pp. 945–952. ACM, 2006.

Abraham Wald. Statistical decision functions. 1950.

Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

Greg Welch, Gary Bishop, et al. An introduction to the Kalman filter. 1995.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. 2008.

APPENDIX

5.1 SOFT Q-LEARNING DERIVATION

We present a derivation of soft Q-learning from the RL as inference parametric approximation to the probability of optimality. Although our presentation is slightly different to that of Levine (2018) we show here that the resulting algorithms are essentially identical. Recall from equation (6) that the parametric approximation is given by

$$P(\mathcal{O}_h(s,a)|\tau) \propto \exp(\sum_{t=h}^{T} \mathbb{E}^L \mu(s',a')|(s',a') \in \tau(s,a,h))$$

where we use the notation $\tau(s,a,h)$ to denote a trajectory starting from state-action (s,a) at timestep h. Now we must marginalize out τ using the (unknown) system dynamics. Since this is a certainty-equivalent algorithm we shall use the expected value of the transition probabilities, under the posterior at timestep L. In which case we obtain

$$\begin{aligned} P(\mathcal{O}_h(s,a)) &= \sum_{(s',a') \in \tau(s,a,h)} P(\mathcal{O}_h(s,a)|\tau)P(\tau) \\ &\propto \exp(\mathbb{E}^L \mu(s,a)) \sum_{(s',a') \in \tau(s,a,h)} \mathbb{E}^L \mathcal{P}(s',s,a) p(y|x) P(\mathcal{O}_h(s',a')|\tau)P(\tau) \end{aligned}$$

now we make the additional assumption that $p(y|s)$ is uniform across all actions so we can ignore it (this assumption is also required in Levine (2018), with this we can rewrite

$$\log P(\mathcal{O}_h(s,a)) = \mathbb{E}^L \mu(s,a) + \log \sum_{(s',a')} \mathbb{E}^L \mathcal{P}(s',s,a) P(\mathcal{O}_h(s',a')) - Z$$

where Z is the normalization constant, from Jensen's then we get

$$\log P(\mathcal{O}_h(s,a)) \geq \mathbb{E}^L \mu(s,a) + \sum_x \mathbb{E}^L \mathcal{P}(x,s,a) \log \sum_{a'} P(\mathcal{O}_h(s',a')) - Z$$

now if we introduce the soft Q-values that satisfy the Bellman equation

$$Q_h(s,a) = \mathbb{E}^L \mu(s,a,h) + \sum_s \mathbb{E}^L \mathcal{P}(x,s,a) \log \sum_{a'} \exp Q_h(s',a')$$

then

$$P(\mathcal{O}_h(s,a)) \approx \exp Q_h(s,a) / \sum_b \exp Q_h(s,b)$$

and we have the soft Q-learning algorithm (the approximation comes from the fact we used Jensen's inequality.

5.2 PROOF OF THEOREM 1

Theorem. *The K-learning value function V^{KL} and policy π^{KL} defined in table 3 satisfy the following bound at every state $s \in \mathcal{S}$ and $h = 0, \ldots H$:*

$$V_h^{\mathrm{KL}}(s) \geq \mathbb{E} V_h^{M,\star}(s) + \beta^{-1} D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \;||\; \pi_h(s)^{\mathrm{KL}}).$$

Proof. Fix some particular state $s \in \mathcal{S}$, and let the joint posterior over value and policy be denoted by

$$\mathbb{P}(V_h(s), \mathcal{O}_h(s,a)) = f(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a))\mathbb{P}(\mathcal{O}_h(s,a)),$$

where we use f to denote the conditional distribution over Q-values conditioned on optimality, and where $\pi^{ts}_{s_h,a_h} = \mathbb{P}(\mathcal{O}_h(s,a))$ is the Thompson sampling policy. Recall that from equation (7) we have approximated the posterior probability of optimality as

$$\tilde{\mathbb{P}}(\mathcal{O}_h(s,a)) \propto \exp G_h^Q(s,a).$$

Form Bayes rule this implies the following approximation to the conditional distribution

$$\tilde{f}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) = \mathbb{P}(Q_h^{M,\star}(s,a))\exp(\beta Q_h^{M,\star}(s,a) - G_h^Q(s,a)(\beta)).$$

This is known as the *exponential tilt* of the posterior distribution $\mathbb{P}(Q_h^{M,\star}(s,a))$ and has a myriad of applications in statistics (Asmussen & Glynn, 2007). This yields the following approximation to the joint distribution $\tilde{f}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a))\tilde{\mathbb{P}}(\mathcal{O}_h(s,a))$. However, the K-learning policy does not follow (7) since computing the cumulant generating function is non-trivial. Instead we compute the K-values, which are the solution to a Bellman equation that provide a guaranteed upper bound on the cumulant generating function, and the K-learning policy is thus

$$\pi_h(s,a)^{\mathrm{KL}} \propto \exp(\beta K_h(s,a)),$$

and recall we have

$$\beta K_h(s,a) \geq G_h^Q(s,a)(\beta). \tag{11}$$

With that in mind we take our approximate joint posterior to be

$$\tilde{\mathbb{P}}(V_h(s), \mathcal{O}_h(s,a)) = \tilde{f}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a))\pi_h(s,a)^{\mathrm{KL}}.$$

Now consider the KL-divergence between the true joint posterior and our approximate one, a quick calculation yields

$$D_{KL}(P_h(s) \mid\mid \tilde{\mathbb{P}}_h(s)) = D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid \pi_h(s)^{\mathrm{KL}}) + \sum_a \mathbb{P}(\mathcal{O}_h(s,a)) D_{KL}(f_h(s) \mid\mid \tilde{f}_h(s)), \tag{12}$$

where we used the subscript s to denote quantities restricted to state s. Considering the terms in (12) separately we have

$$D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid \pi_h(s)^{kl}) = -\mathcal{H}(\mathbb{P}(\mathcal{O}_h(s))) - \beta \sum_a \pi^{ts}_{s_h,a_h} K_h(s,a) + \log \sum_a \exp \beta K_h(s,a)$$

where $\mathcal{H}$ denotes the entropy, and

$$\begin{aligned}\sum_a \mathbb{P}(\mathcal{O}_h(s,a)) D_{KL}(f \mid\mid \tilde{f}) = \sum_a \pi^{ts}_{s_h,a_h} G_h^Q(s,a)(\beta) - \beta \sum_a \mathbb{P}(\mathcal{O}_h(s,a))\mathbb{E}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) \\ + \sum_a \mathbb{P}(\mathcal{O}_h(s,a)) D_{KL}(\mathbb{P}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) \mid\mid \mathbb{P}(Q_h^{M,\star}(s,a))).\end{aligned}$$

Now we sum these two terms, using (11) and the following identities

$$\sum_a \mathbb{P}(\mathcal{O}_h(s,a))\mathbb{E}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) = \mathbb{E}\max_a Q_h^{M,\star}(s,a) = \mathbb{E}V_h^{M,\star}(s)$$

and

$$\begin{aligned}&\sum_a \mathbb{P}(\mathcal{O}_h(s,a)) D_{KL}(\mathbb{P}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) \mid\mid \mathbb{P}(Q_h^{M,\star}(s,a))) \\ &\quad = \sum_a \mathbb{P}(\mathcal{O}_h(s,a)) \int \mathbb{P}(Q_h^{M,\star}(s,a)|\mathcal{O}_h(s,a)) \log(\mathbb{P}(\mathcal{O}_h(s,a)|Q_h^{M,\star}(s,a))) + \mathcal{H}(\mathbb{P}(\mathcal{O}_h(s))) \\ &\quad \leq \mathcal{H}(\mathbb{P}(\mathcal{O}_h(s))),\end{aligned}$$

we obtain

$$D_{KL}(P_h(s) \mid\mid \tilde{\mathbb{P}}_h(s)) \leq \log \sum_a \exp \beta K_h(s,a) - \beta \mathbb{E} V_h^{M,\star}(s).$$

The theorem follows from this and the fact that the K-learning value function is defined as

$$V_h(s)^{kl}(\beta) = \beta^{-1} \log \sum_a \exp \beta K_h(s,a)$$

as well as the fact that $D_{KL}(\mathbb{P}(\mathcal{O}_h(s)) \mid\mid \pi_h(s)^{\mathrm{KL}}) \leq D_{KL}(P_h(s) \mid\mid \tilde{\mathbb{P}}_h(s))$ from equation (12). □

5.3 Problem 1 K-learning details

For a bandit problem the K-learning policy is given by

$$\pi_i^{\mathrm{KL}} \propto \exp G_i^{\mu}(\beta),$$

which requires the cumulant generating function of the posterior over each arm. For arm 1 and the distractor arms there is no uncertainty, in which case the cumulant generating function is given by

$$G_i^{\mu}(\beta) = \mu_i \beta, \quad i = 1, 3, \dots N.$$

In the case of arm 2 the cumulant generating function is

$$G_2^{\mu}(\beta) = \log\big((1/2)\exp(2\beta) + (1/2)\exp(-2\beta)\big).$$

In (O'Donoghue, 2018) it was shown that the optimal choice of β is given by

$$\beta^\star = \operatorname*{argmin}_{\beta \geq 0} \left(\beta^{-1} \log \sum_{i=1}^{N} \exp G_i^{\mu}(\beta)\right),$$

which requires solving a convex optimization problem in variable β^{-1}. In the case of problem 1 the optimal choice of $\beta \approx 10.23$, which yields $\pi_2^{kl} \approx 0.94$. Then, once arm 2 has been pulled once and the true reward of arm 2 has been revealed, its cumulant generating function has the same form as the others, and then the optimal choice of β is simply

$$\beta^\star = \operatorname*{argmin}_{\beta \geq 0} \left(\beta^{-1} \log \sum_{i=1}^{N} \exp \mu_i \beta\right) = \infty,$$

at which point K-learning is greedy with respect to the optimal arm.

6 bsuite report: Varying β in Soft Q learning

The *Behaviour Suite for Reinforcement Learning*, or bsuite for short, is a collection of carefully-designed experiments that investigate core capabilities of a reinforcement learning (RL) agent. The aim of the bsuite project is to collect clear, informative and scalable problems that capture key issues in the design of efficient and general learning algorithms and study agent behaviour through their performance on these shared benchmarks. This report provides a snapshot of agent performance on bsuite2019, obtained by running the experiments from github.com/deepmind/bsuite Osband et al. (2019).

6.1 Agent definition

All agents correspond to different instantiations of the DQN agent with Soft Q learning (Mnih et al., 2015; O'Donoghue et al., 2017). We use a 2 layer MLP with 50 hidden units, Adam optimizer and learning rate 1e-3. The only parameter we vary is the temperature β as defined in Table 2.

6.2 Summary scores

Each bsuite experiment outputs a summary score in [0,1]. We aggregate these scores by according to key experiment type, according to the standard analysis notebook. A detailed analysis of each of these experiments may be found in a notebook hosted on Colaboratory: (withheld for anonynomity).

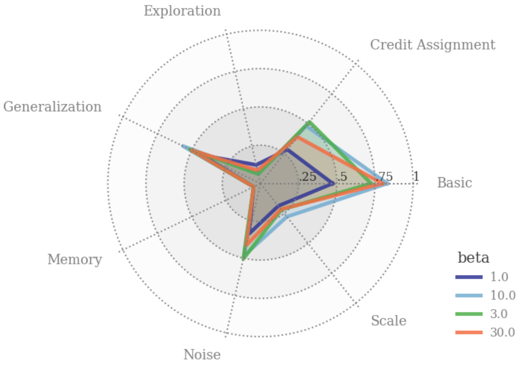


Figure 2: Snapshot of agent behaviour.

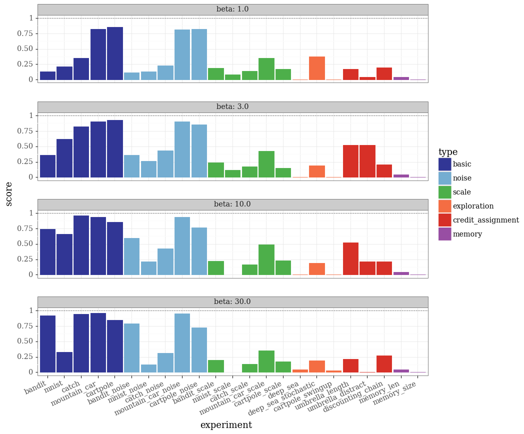


Figure 3: Score for each bsuite experiment.

6.3 Results commentary

Although we swept the temperature parameter β over a large range of values, we do not see a huge difference in the performance of the agent. For each setting of β the agent performs reasonably on the basic tasks (except for very small $\beta = 1$ where action selection is too random). Importantly, there is no setting of β that appreciably improves the agent's performance on the hard *exploration* tasks. See the colab for a full description of results.