CSC2125 Project Report: Safe Exploration with Bayesian Reinforcement Learning

Eric Langlois
edl@cs.toronto.edu
University of Toronto
Toronto, Ontario

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; *Planning and scheduling*; *Search with partial observations.*

KEYWORDS

safe exploration, bayesian reinforcement learning

1 INTRODUCTION

Reinforcement learning (RL) can be a powerful method for developing controllers for complex environments, including autonomous driving. RL algorithms learn online and must explore the environment to discover good control policies. Unfortunately, standard RL algorithms make no attempt to avoid danger when exploring and are therefore unsuitable for safety-critical applications.

Safe reinforcement learning [2] is the study of online learning that attempts to respect safety constraints given uncertainty about the environment. The safety criteria must be specified explicitly and it can be difficult to design appropriate safety constraints. Example constraints include avoiding a set of undesirable states or a minimum threshold on total reward.

Bayesian reinforcement learning [3] also considers environment uncertainty but is concerned with finding optimal learning behaviour. Bayesian RL is primarily motivated by the explore / exploit trade-off in RL. By optimizing for the learning behaviour that produces maximum expected total reward, this approach automatically identifies the most profitable explore-exploit balance.

I demonstrate that the same property also causes Bayesian RL to explore cautiously when there are uncertain dangers in the environment that can impact long-term reward. Instead of ad-hoc safety constraints, safe behaviour arises naturally when considering a distribution of possible environments. Bayesian reinforcement learning therefore presents an alternative principled approach to safe reinforcement learning. [1]

2 RELATED WORK

Both Bayesian reinforcement learning and safe reinforcement learning are established domains with sizeable bodies of research (surveyed in the citations above). To my knowledge, they have remained distinct and there has not been any investigation into potential emergent safe behaviour from Bayesian RL. Kim et al. [5] develop a Bayesian RL algorithm with safe exploration but it uses explicit safety constraints.

The algorithms used here are all from existing work. They are described in greater detail in the following sections.

[1] The introduction and related work text are mostly copied from the project proposal.

3 BACKGROUND

3.1 Bayesian Reinforcement Learning

Bayesian Reinforcement Learning is the task of learning to behave given a distribution over possible Markov Decision Processes (MDPs). The agent has access to the prior distribution but does not know the specific MDP it is interacting with at test time. An effective Bayesian RL agent must reason about how potential future observations will affect the posterior MDPs distribution and deliberately explore accordingly. In contrast, non-Bayesian RL agents do not explicitly represent a distribution over MDPs and instead use heuristics for exploration.

A common approach to Bayesian RL is to transform the problem into an MDP called the Bayes-Adaptive Markov Decision Process (BAMDP) [1]. A BAMDP augments the original state space by including a distributions over possible MDPs, representing the posterior MDP distribution at that state. Transitions between states in ths BAMDP implement both the original state transition (marginalized over the MDP posterior) and a Bayes-rule update to the belief state (the posterior distribution).

The BAMDP is an MDP and in principle, standard reinforcement learning algorithms could be employed to solve it. Unfortunately, this is often not possible in practice for two reasons: (1) the augmented state space is exponentially large compared to the original state space; and (2) BAMDP state transitions involve Bayes-rule updates that are often difficult or impossible to evaluate exactly.

3.2 BAMCP Algorithm

The Bayes-Adaptive Monte-Carlo Planning (BAMCP) algorithm by Guez et al. [4] is a relatively efficient algorithm for planning over BAMDPs. BAMCP repeatedly samples an MDP from the current posterior then searches in the sampled MDP. This avoids costly Bayesian updates during planning; Bayesian updates only happen when an action is performed in the true environment.

BAMCP uses the UCT tree-search algorithm [6] to perform its search and to aggregate the results of different MDP samples. UCT is effective at searching large state spaces and is commonly used for game play, including in the highly successful AlphaZero [7].

4 METHODOLOGY

A Bayesian RL algorithm was implemented, along with several baseline RL agents. These were evaluated on a simple environment distribution with uncertain terminal actions and compared with respect to performance and safety.

4.1 Deadly Bandits Environment

To evaluate safe exploration, a "deadly bandits" [2]environment was developed. "Deadly bandits" is a modification of the standard multi-armed bandit environment in which arms also have chance of terminating the episode. As always, the objective is to maximize total (discounted) reward earned in an episode.

An instance of the deadly bandits MDP consists of a single state with n actions. Each action a yields a deterministic reward $r_a \in [0, 1]$. With termination probability p_a, the episode terminates, otherwise the episode continues.

The evaluation environment consists of a distribution over deadly bandit MDPs with 10 arms. Arm rewards are drawn uniformly and independently from $\{0.2, 0.4, 0.6, 0.8, 1\}$. [3] Termination probabilities are drawn uniformly and independently from $\{0, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}\}$.

In the "Unknown Deadly Bandits" environment, rewards are visible to the agent but the termination probabilities are not. Agents are run on a particular MDP sample for one episode only so all an agent knows about action termination probabilities are that it has survived all actions it has applied so far. Agents never have the opportunity to learn from an action that terminates.

A modified "Known Deadly Bandits" environment makes the termination probabilities accessible to the agent. Instead of a single state, the environment has one state per termination probability. The states are functionally identical but indicate the termination probability index of the previous action.

In both versions of this environment, Bayesian updates to the posterior given observations are quite simple. It is not the objective of this project to compare Bayesian RL techniques (which should all have the same asymptotic behaviour), but there is an opportunity to try Bayesian RL algorithms other than BAMCP in case higher quality solutions can be generated more efficiently.

4.2 Evaluation

Evaluation consists of repeated trials in which an agent interacts with the environment until a terminal state is reached or the maximum number of steps (500) occurs. The primary objective of these experiments is to evaluate the safety properties of the agents in question. As such, each agent instance is run for exactly one episode. Unlike the typical RL paradigm, these agents have no opportunity to learn from failure. A successful agent must behave safely on its first pass through a sample MDP.

4.3 BAMCP Implementation

The BAMCP algorithm was chosen as the representative Bayesian RL solution. While BAMCP avoids any parameterization of its safety behaviour, it does require a number parameters to control the search. A discount factor (0.999) and discount threshold (0.01) determine the depth of the BAMCP searches. An internal epsilon-greedy Q-learning policy is parameterized by a learning rate (0.05) and a random action probability (0.1). Multiple searches (5000) are performed before each action, and more (20000) before the first action.

[2]The original intention for the project was to focus on autonomous vehicle applications. I created the "deadly bandits" environment as a minimal test environment with the desired characteristics, namely, repeatable dangerous actions that necessitate safe exploration. Given the time constraint, I was unable to expand into more concrete applications. Nevertheless, "deadly bandits" can be a relevant model. For example, it is analogous to the task of repeatedly choosing between a set of routes with different lengths (negative reward) and different road conditions (success probability).

[3]The reward space must be finite otherwise the BAMCP search tree has an infinite branching factor.

4.4 Baseline Agents

The following baseline agents were implemented:

Uniform Random : Selects actions uniformly at random.
Constant : Always selects the first action.
Q-Learning : Epsilon-greedy tabular Q learning[8]. Same parameters as the BAMCP internal Q learner.

None of the Safe RL algorithms that I investigated appeared to be well suited to this domain. Safe RL typically involves learning an uncertainty model over unknown states or actions. In the deadly bandits environment, actions are discrete and independent so an agent cannot learn anything about action without trying it. Furthermore, the observed history is entirely deterministic until termination occurs so an agent without access to the prior MDP distribution cannot learn about uncertainty from its experiences.

The **Constant** agent serves as the baseline safe RL agent. It is maximally cautious: never trying more than one action. A more intelligent Bayesian agent might attempt to notice if its initial action is one with high termination probability and explore just enough to find another action with lower termination probability. However, as noted above, no non-Bayesian agent can learn about termination probabilities.

5 RESULTS

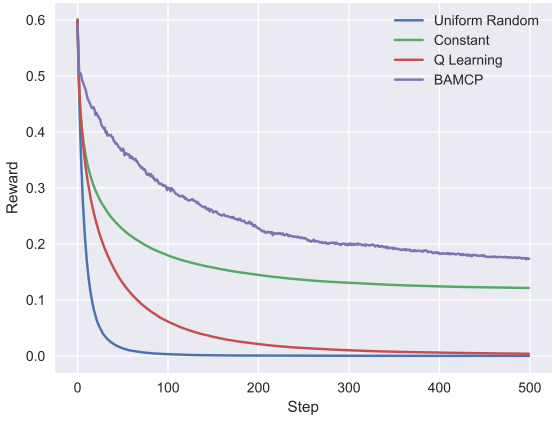

Figure 1: Mean per-step reward on Known Deadly Bandits.

The results of the "Known Deadly Bandits" experiment are shown in Figures 1, 2, and 3. Table 1 presents a summary of the episode statistics.

The Uniform Random and Q-Learning agents both quickly try all possible actions (Figure 3) and hit terminal states. Figure 2 shows the fraction of agents that survive to a given step index. As a result, their mean per-step rewards are low (Figure 1). In the case of Q-learning, the smaller number of surviving states outweighs any increased rewards those surviving agents achieve.

Table 1: Deadly bandits mean episode statistics.

Environment	Agent	Num. Episodes	Reward	Survival Rate	Runtime (ms)
Known Deadly	Uniform	100,000	6	6e-5	0.9
	Constant	100,000	80	0.202	5.9
	Q-Learning	100,000	23	0.006	5.9
	BAMCP	1,000	122	0.211	487,720

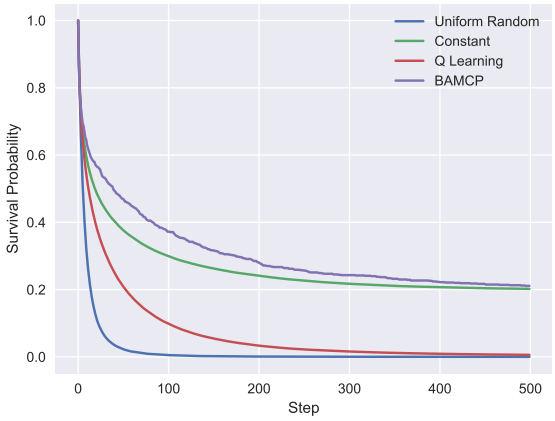

Figure 2: Mean survival rate on Known Deadly Bandits.

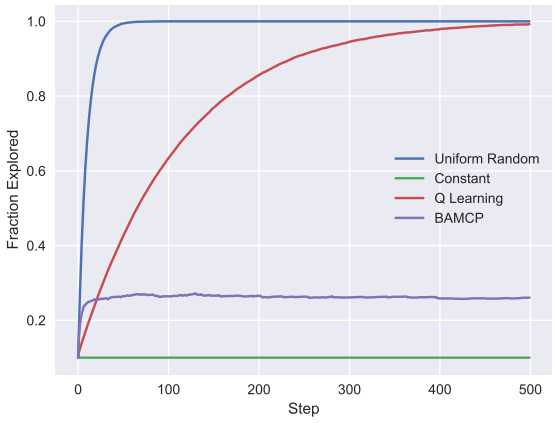

Figure 3: Fraction of actions explored on Known Deadly Bandits.

The Constant agent earns a moderate amount of reward by avoiding any exploration. It still suffers terminations since the action it chooses may have non-zero termination probability. But those agents that get lucky with a 100% success action will survive the entire episode.

Meanwhile, BAMCP agent performs the best with respect to both total reward and survival rate. The average step reward is higher than the Constant agent indicating that it performs some searching in the beginning for good actions (also visible in Figure 3). However, once it finds a satisfactory action it stops exploring and maintains a high survival rate. Interestingly, the survival rate is also better than that of the Constant agent. This suggests that the BAMCP agent also searches for a less risky action than whatever action it might have first tried.

Similar experiments were performed for the Unknown Deadly Bandits environment. The behaviour of the BAMCP agent is sufficiently unreasonable (very rapid exploration and termination) that I suspect an implementation error in the environment.

6 DISCUSSION

These preliminary experiments show that Bayesian RL agents can be both safe and effective learners. As part of future work I would like to diversify the evaluation environments and improve the quality of the baselines. In particular, I would like to find domains in which Bayesian RL can be compared against more typical Safe RL algorithms.

REFERENCES

[1] Michael O'Gordon Duff. 2002. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes.* Ph.D. Dissertation. University of Massachusetts at Amherst.

[2] Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16 (2015), 1437–1480.

[3] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. 2015. Bayesian Reinforcement Learning: A Survey. *Foundations and Trends in Machine Learning* 8, 5-6 (2015), 359–483.

[4] Arthur Guez, David Silver, and Peter Dayan. 2013. Scalable and Efficient Bayes-Adaptive Reinforcement Learning Based on Monte-Carlo Tree Search. *J. Artif. Intell. Res.* 48 (2013), 841–883.

[5] Dongho Kim, Kee-Eung Kim, and Pascal Poupart. 2012. Cost-Sensitive Exploration in Bayesian Reinforcement Learning. In *NIPS*. 3077–3085.

[6] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *ECML (Lecture Notes in Computer Science)*, Vol. 4212. Springer, 282–293.

[7] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR* abs/1712.01815 (2017).

[8] Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards.* Ph.D. Dissertation. King's College, Cambridge, UK. http://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf