THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Sample Efficient Bayesian Reinforcement Learning

DIVYA GROVER

Division of Data Science and AI
Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2020

Sample Efficient Bayesian Reinforcement Learning
DIVYA GROVER

Thesis for the degree of Licentiate of Engineering
ISSN 1652-876X
Technical Report No. 213L
Division of Data Science and AI

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Phone: +46 (0)31 772 10 00

Author e-mail: divya.grover@chalmers.se

Cover:
This shows how to build intelligent agents capable of taking actions under uncertainty.

Printed by Chalmers Reproservice
Göteborg, Sweden year

To my parents, Monila and Suraj Grover.

ABSTRACT

Artificial Intelligence (AI) has been an active field of research for over a century now. The research field of AI may be grouped into various tasks that are expected from an intelligent agent; two major ones being *learning & inference* and *planning*. The act of storing new knowledge is known as learning while inference refers to the act to extracting conclusions given agent's limited knowledge base. They are tightly knit by the design of its knowledge base. The process of deciding long-term actions or plans given its current knowledge is called planning.

Reinforcement Learning (RL) brings together these two tasks by posing a seemingly benign question "How to act optimally in an unknown environment?". This requires the agent to learn about its environment as well as plan actions given its current knowledge about it. In RL, the environment can be represented by a mathematical model and we associate an intrinsic value to the actions that the agent may choose.

In this thesis, we present a novel Bayesian algorithm for the problem of RL. *Bayesian RL* is a widely explored area of research but is constrained by scalability and performance issues. We provide first steps towards rigorous analysis of these types of algorithms. Bayesian algorithms are characterized by the belief that they maintain over their unknowns; which is updated based on the collected evidence. This is different from the traditional approach in RL in terms of problem formulation and formal guarantees. Our novel algorithm combines aspects of planning and learning due to its inherent Bayesian formulation. It does so in a more scalable fashion, with formal PAC guarantees. We also give insights on the application of Bayesian framework for the estimation of model and value, in a joint work on Bayesian backward induction for RL.

Keywords: Bayesian Reinforcement Learning, Decision Making under Uncertainty

ACKNOWLEDGMENTS

I would not have been able to write this licentiate thesis without the support of many people around me. I want to thank a few of them here for the time they gave me.

First, I would like to thank my advisor Christos Dimitrakakis for his enormous support. Your support and guidance has been invaluable to me. You fill the gaps in my knowledge and your sharp counter-examples have helped me wrap my head around very many things. Thanks for encouraging me to collaborate, e.g., the Harvard trip that you arranged for me. Also thank you for helping me out with this thesis. Next, is my co-supervisor Daniil Ryabko his support. I recall our many email exchanges, and his expertise in Bandits to help me with some proofs. I like to thank my co-author Debabrota Basu for his continued support and encouragement to pursue the right problems. Furthermore, I would like to express my sincere gratitude to Frans A. Oliehoek for taking his time and energy to read this work and lead the discussions of my licentiate seminar.

I cannot forget to thank Devdatt Dubashi for our many interactions, as a guide, colleague and examiner. I recall our chats on India, Chalmers, Sweden and everything in-between. I am grateful to Aristide Tossou for his bits of advice about life in Sweden; Hannes Eriksson and Emilio Jorge for the many fruitful discussions on RL; Shirin Tavara for being a very supportive office-mate. Furthermore, I show my gratitude to the many people whose name are not mentioned here but worked behind the scenes to help me. Finally, I am indebted to my wife, family and friends who increased my motivation to continue this thesis.

LIST OF PUBLICATIONS

This thesis is based on the following manuscripts.

- **Divya Grover**, Christos Dimitrakakis. "Deeper & Sparser Exploration" in 35^{th} *International Conference on Machine Learning. Exploration in RL workshop*, Stockholm, Sweden, July 10-15, 2018.
- **Divya Grover**, Debabrota Basu, Christos Dimitrakakis. "Bayesian Reinforcement Learning via Deep, Sparse Sampling" in 23^{rd} *International Conference on Artificial Intelligence and Statistics*, Palermo, Italy, June 3-5, 2020.

The following manuscript is under review.

- Christos Dimitrakakis, Hannes Eriksson, Emilio Jorge, **Divya Grover**, Debabrota Basu. "Inferential Induction: Joint Bayesian Estimation of MDPs and Value Functions", arXiv preprint arXiv:2002.03098.

Contents

I EXTENDED SUMMARY **1**

1 Introduction **3**

2 Background **7**

2.1 Preliminaries 7

2.1.1 Markov Decision Process (MDP) 7

2.1.2 Bayes Adaptive MDP (BAMDP) 9

2.2 Discussion 11

2.2.1 Bandit problem 12

2.2.2 Model-based Bayesian RL 13

2.2.3 POMDP literature 16

2.2.4 Bayesian value function 17

3 Efficient Bayesian RL **19**

3.1 Deep, Sparse Sampling 20

3.2 Bayesian backward induction (BBI) 23

4 Concluding Remarks **27**

II PUBLICATIONS **35**

List of Figures

2.1 Full tree expansion. 12

3.1 Deeper & Sparser tree expansion. 22

Part I

EXTENDED SUMMARY

Chapter 1

Introduction

The field of Operations Research can be seen as a precursor to modern AI. It is a discipline that deals with the application of advanced analytical methods in making better decisions. During world wars, many problems ranging from project planning, network optimization, resource allocation, resource assignment and scheduling were studied within OR. The techniques used to solve them were extensively studied by J. Von Neumann, A. Wald, Bellman and many others. These are now colloquially referred to as Dynamic Programming (DP) techniques. DP is arguably the most important method for dealing with a large set of mathematical problems, known as decision making problems.

Decision making:

Decision making refers to those situations where an algorithm must interact with a system to achieve a desired objective. A fundamental characteristic of such problems is the feedback effect that these interactions have on the system. In AI, this is analogous to the situation where an autonomous agent acts in an environment. In many cases, we model this situation with a mathematical model that encapsulates the basics of an interacting system.

Decision making problems can be divided two types, depending on the level of difficulty in solving them. First is decision making under no uncertainty, which

refers to the situation where we have full knowledge of the system's[1] behaviour. Even in this case, it is not a trivial problem to decide how to act ? This process of developing long-term actions is known as *planning*.

Decision making under uncertainty:

The second type is decision making under uncertainty. In real world processes, along with inherent (aleatoric) uncertainty[2], there also exists uncertainty of our knowledge (epistemic) about it. Reinforcement Learning (RL) is an important problem in this category. According to Duff [2002], "RL attempts to import concepts from classical decision theory and utility theory to the domain of abstract agents operating in uncertain environments. It lies at the intersection of control theory, operations research, artificial intelligence and animal learning." Its first successful application was the development of a state-of-the-art Backgammon playing AI [Tesauro, 1994]. More recent successes include game [Mnih et al., 2015, Silver et al., 2017] playing AI.

Planning in this general setting requires taking into account future events and observations that may change our conclusions. Typically, this involves creating long-term plans covering possible future eventualities, i.e. when planning under uncertainty, we also need to take into account the possible future knowledge that could be generated while acting. Executing actions also involve trying out new things, to gather more information, but it is hard to tell whether this information will be beneficial. The choice between acting in a manner that is known to produce good results, or experimenting with something new, is known as the *exploration-exploitation* dilemma. It is central to RL research.

Exploration-Exploitation:

Consider the problem of choosing your education stream for your long-term career. Let's say you are inclined towards Engineering. However, Project Management has recently been growing quite popular and is financially more rewarding. It is tempting to try it out! But there is a risk involved. It may turn out to be much worse than Engineering, in which case you will regret switching streams. On the other hand, it could also be much better. What should

[1] Potentially stochastic.

[2] Like the randomness associated with skating on ice.

you do? It all depends on how much information you have about either career choices and how many more years are you willing to spend to get a degree. If you already have a PhD, then its probably a better idea to go with Engineering. However, if you just finished your bachelors degree, Project Management may be a good bet. If you are lucky, you will get a much higher salary for the remainder of your life, while otherwise you would miss out only by a year, making the potential risk quite small.

Bayesian Reinforcement Learning:

One way to approach the exploration-exploitation dilemma is to take decisions that explicitly take into account the uncertainty, both in the present and the future. One may use the Bayesian approach for this; essentially any algorithm is Bayesian in nature if it maintains probabilistic beliefs on quantities of interest and updates them using evidence collected. Formulating the RL problem in a Bayesian framework is known as Bayesian RL. Planning trees are data structures used in various planning algorithms. A belief tree is a planning tree used to plan actions while explicitly taking into accounts their future effects, like in a Bayesian RL setting.

Main contribution:

Our main contribution is a novel Bayesian RL algorithm, for planning in belief trees. We perform a thorough analysis of this algorithm by giving a performance bound for it.

Thesis outline:

In Chapter 2, we formally define the terms of interest and introduce the necessary background that will help understand the remainder of this thesis. In Chapter 3, we summarize the contributions of this thesis. Section (3.1) discusses the algorithm we introduced, discussing its practical benefit and providing a theoretical guarantee for it. Section (3.2) discusses our other work, performed jointly, that takes an orthogonal approach to the same Bayesian RL problem. Chapter 4 concludes the thesis and discusses some interesting future work. The remainder of this thesis is a reprint of the full version papers [Grover and Dimitrakakis, 2018, Grover et al., 2019, Dimitrakakis et al., 2020].

Chapter 2

Background

We divide this chapter into two sections. We introduces the necessary preliminaries for this work in section (2.1), followed by an in-depth discussion on many possible approaches to Bayesian RL in section (2.2).

2.1 Preliminaries

We first define the mathematical model used to describe an environment, known as Markov Decision Process (MDP). We then define a specific type of MDP, called Bayes Adaptive MDP, that arises when we are uncertain about the underlying environment, which is the focus of this thesis.

2.1.1 Markov Decision Process (MDP)

Markov Decision Process (MDP) is a discrete-time stochastic process that provides a formal framework for RL problems.

Definition 1 (MDP)**.** *An MDP $\mu = (S, A, P, R)$ is composed of a state space S, an action space A, a reward distribution R and a transition function P. The transition function $P \triangleq \mathbb{P}_\mu(s_{t+1}|s_t, a_t)$ dictates the distribution over next states s_{t+1} given the present state-action pair (s_t, a_t). The reward distribution $R \triangleq \mathbb{P}_\mu(r_{t+1}|s_t, a_t)$ dictates the obtained reward that belongs to the interval*

$[0, 1]$. *We shall also use* $\mathbb{P}_\mu(r_{t+1}, s_{t+1}|s_t, a_t)$ *to denote the joint distribution of next states and actions of MDP* μ.

A policy π belonging to a policy space Π is an algorithm for selecting actions given the present state and previous observations. The objective of an agent is to find the policy π that maximizes the sum of discounted reward average over all uncertainties.
The value function of a policy π for an MDP μ is the expected sum of discounted rewards obtained from time t to T while selecting actions in the MDP μ:

$$V_{\mu,t}^{\pi,T}(s) = \mathbb{E}_\mu^\pi(\sum_{k=1}^{T} \gamma^k r_{t+k} \mid s_t = s), \tag{2.1}$$

where $\gamma \in (0, 1]$ is called the discount factor and $\mathbb{E}_\mu^\pi$ denotes the expectation under the trajectory generated by a policy π acting on the MDP μ. Let us define the infinite horizon discounted value function of a policy π on an MDP μ as $V_\mu^\pi \triangleq \lim_{T\to\infty} V_{\mu,0}^{\pi,T}$. Now, we define the optimal value function to be $V_\mu^* \triangleq \max_\pi V_\mu^\pi$, and the optimal policy to be $\pi_\mu^* \triangleq \arg\max_\pi V_\mu^\pi$.

A note on policies: The largest policy set, denote Π_T, can have a cardinality of $|A|^T$. Some other important policy sets are:

1. Stationary policies: All policies that are consistent over time, i.e, $\pi(a_t|s_t, s_{t-1} \dots s_{t-k}) = \pi(a_{t'}|s_{t'}, s_{t'-1} \dots s_{t'-k}) \quad \forall t, t'$.
2. K-order Markov policies: All polices that only depend on previous K states, i.e., $\pi(a_t|s_t, s_{t-1} \dots s_0) = \pi(a_t|s_t, s_{t-1} \dots s_{t-k})$. They may or may not be stationary.
3. Deterministic policies: Policies with a trivial action distribution, i.e, $\pi(a_t = a| \dots) = 1$ for an action a.

We focus in this thesis on history-dependent polices, commonly referred to as adaptive policies.

Define Bellman operator, $\mathscr{B}_\mu^\pi : V \to V$, as follows :

$$\mathscr{B}_\mu^\pi V(s) \triangleq \mathbb{E}_\mu^\pi(r) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}_\mu^\pi(s' \mid s) V(s'),$$

It allows us to compute the value function recursively via $V^\pi_{\mu,t} = \mathscr{B}^\pi_\mu V^\pi_{\mu,t+1}$. Repeated application of the relevant Bellman operator[1] on the initial value function is referred here as *backward induction*.

If the MDP is known, the optimal policy and value function is computable via backward induction (also known as *value iteration*). Another important result is that there exists an optimal policy in the set of 1-order deterministic stationary Markov policies for an MDP [Puterman, 1994].

2.1.2 Bayes Adaptive MDP (BAMDP)

In reality, the underlying MDP is unknown to the RL algorithm, which gives rise to the exploration-exploitation dilemma. Bayesian Reinforcement Learning (BRL), specifically the information state formulation [Duff, 2002], provides a framework to quantify this trade-off using a Bayesian representation.
Following the Bayesian formulation, we maintain a belief distribution β_t over the possible MDP models $\mu \in \mathcal{M}$.[2] Starting with an appropriate prior belief $\beta_0(\mu)$, we obtain a sequence of posterior beliefs $\beta_t(\mu)$ that represent our subjective belief over the MDPs at time t, depending on the latest observations. By Bayes' rule, the posterior belief at time $t+1$ is

$$\beta_{t+1}(\mu) \triangleq \frac{\mathbb{P}_\mu(r_{t+1}, s_{t+1}|s_t, a_t)\beta_t(\mu)}{\int_{\mathcal{M}} \mathbb{P}'_\mu(r_{t+1}, s_{t+1}|s_t, a_t)\beta_t(\mu')d\mu'}. \tag{2.2}$$

Now, we define the Bayesian value function v analogously to the MDP value function:

$$v^\pi_\beta(s) \triangleq \int_{\mathcal{M}} V^\pi_\mu(s)\beta(\mu)d\mu. \tag{2.3}$$

The Bayesian value function is the expected utility of the decision maker according to its current belief β and policy π for selecting future actions from state s. The optimal policy for the Bayesian value function can be adaptive in general. For completeness, we also define the Bayes-optimal utility $v^*_\beta(s)$, i.e. the utility of the Bayes-optimal policy:

$$v^*_\beta(s) \triangleq \max_{\pi\in\Pi} \int_{\mathcal{M}} V^\pi_\mu(s)\beta(\mu)d\mu. \tag{2.4}$$

[1] BAMDP has a slightly different one.

[2] More precisely, we can define a measurable space $(\mathcal{M}, \mathfrak{M})$, where $\mathcal{M}$ is the possible set of MDPs, and $\mathfrak{M}$ is a suitable σ-algebra.

It is well known that by combining the original MDPs state s_t and belief β_t into a hyper-state ω_t, we obtain another MDP called the Bayes Adaptive MDP (BAMDP). The optimal policy for a BAMDP is the same as the Bayes-optimal policy for the corresponding MDP.

Definition 2 (BAMDP)**.** *A Bayes Adaptive Markov Decision Process (BAMDP)* $\tilde{\mu} \triangleq (\Omega, A, \nu, \tau)$ *is a representation for an unknown MDP* $\mu = (S, A, P, R)$ *with a space of information states* $\Omega = S \times \mathfrak{B}$*, where* $\mathfrak{B}$ *is an appropriate set of belief distributions on* $\mathcal{M}$*. At time t, the agent observes the information state* $\omega_t = (s_t, \beta_t)$ *and takes action* $a_t \in A$*. We denote the transition distribution as* $\nu(\omega_{t+1}|\omega_t, a_t)$*, the reward distribution as* $\tau(r_{t+1}|\omega_t, a_t)$*, and* A *as the common action space.*

For each s_{t+1}, the next hyper-state $\omega_{t+1} = (s_{t+1}, \beta_{t+1})$ is uniquely determined since β_{t+1} is unique given (ω_t, s_{t+1}) and can be computed using eq. (2.2). Therefore the information state ω_t preserves the Markov property. This allows us to treat the BAMDP as an infinite-state MDP with $\nu(\omega_{t+1}|\omega_t, a_t)$, and $\tau(r_{t+1}|\omega_t, a_t)$ defined as the corresponding transition and reward distributions respectively. The transition and reward distributions are defined as the marginal distributions

$$\nu(\omega_{t+1}|\omega_t, a_t) \triangleq \int_{\mathcal{M}} \mathbb{P}_\mu(s_{t+1}|s_t, a_t)\beta_t(\mu)d\mu,$$
$$\tau(r_{t+1}|\omega_t, a_t) \triangleq \int_{\mathcal{M}} \mathbb{P}_\mu(r_{t+1}|s_t, a_t)\beta_t(\mu)d\mu.$$

Though the Bayes-optimal policy is generally adaptive in the original MDP, it is Markov with respect to the hyper-state of the BAMDP. In other words, ω_t represents a sufficient statistic for the observed history.

Since the BAMDP is an MDP on the space of hyper-states, we can use value iteration starting from the set of terminal hyper-states Ω_T and proceeding backwards from horizon T to t following

$$V_t^*(\omega) = \max_{a \in A} \mathbb{E}[r \mid \omega, a] + \gamma \sum_{\omega' \in \Omega_{t+1}} \nu(\omega'|\omega, a) V_{t+1}^*(\omega'), \tag{2.5}$$

where Ω_{t+1} is the reachable set of hyper-states from hyper-state ω_t. Equa-

tion (2.4) implies Equation (2.5) and vice-versa[3], i.e. $v^*_\beta(s) = V^*_0(\omega)$ for $\omega = (s, \beta)$. Hence, we can obtain Bayes-optimal policies through backward induction. Due to the large hyper-state space, this is only feasible for a small finite horizon in practice, as shown in Algorithm 1.

Algorithm 1 FHTS (Finite Horizon Tree Search)

Parameters: Horizon T
Input: current hyper-state ω_h and depth h.
if $h = T$ **then**
  return V(ω_h) = 0
end if
for all actions a **do**
  for all next states s_{h+1} **do**
    β_{h+1} = UpdatePosterior(ω_h,s_{h+1},a) (eq. 2.2)
    ω_{h+1} = (s_{h+1}, β_{h+1})
    $V(\omega_{h+1}) =$ FHTS$(\omega_{h+1}, h+1)$
  end for
end for
Q(ω_h, a) = 0
for all ω_{h+1}, a **do**
  $Q(\omega_h, a)$ += $\nu(\omega_{h+1}|\omega_h, a) \times V(\omega_{h+1})$
end for
return $\max_a Q(\omega_h, a)$

2.2 Discussion

In this section, we first discuss the motivation for Bayesian RL formulation. We do this by making an analogy to the Bandit problem, for which the theory is much more clear and developed. Then we discuss algorithms that directly attack the BAMDP problem, followed by a POMDP perspective on Bayesian RL. Finally, we discuss Bayesian value function algorithms, which are premise

[3]The equivalence can be obtained by expanding the integral eq. (2.4) using definition of value function and applying Bayes rule to its second term. This gives the desired recursive equation.

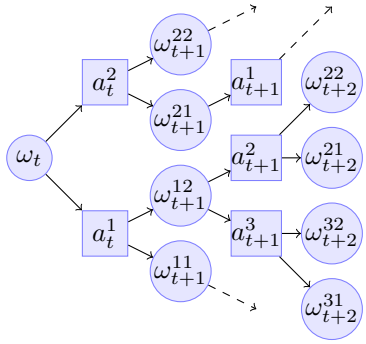

Figure 2.1: Full tree expansion.

to our work on Bayesian backward induction.

2.2.1 Bandit problem

Consider a very simple MDP, with only a single state and multiple actions. This is known as stochastic multi-armed bandit problem and is well studied in Bandit theory. RL in multistate MDPs, in addition to exploration-exploitation dilemma, presents the difficulties of delayed reward and non independence of consecutive samples from the MDP process. But the Bandit theory without these additional difficulties presents a cleaner view of the dilemma.

A stochastic multi-armed bandit model, denoted by $\nu = (\nu_1, ..., \nu_K)$, is a collection of K arms (or actions according to MDP notation), where each arm ν_a when selected, generates a reward with a probability distribution[4]. Agent interacts with a Bandit MDP by choosing at each time t an arm A_t to play. This action results in a realization X_t from the respective distribution ν_{A_t}. The distribution is assumed to be parameterized by $\theta \in \Theta$. Optimality of an agent is defined in terms of Regret:

$$R_\theta \triangleq T\mu^* - \sum_{t=0}^{T} X_t$$

where μ^* is the mean of the optimal arm ν^*.

[4]From here onwards, we refer to ν_i as the arm as well as its underlying reward distribution.

Frequentist and Bayesian analysis: A fundamental difference between Frequentist and Bayesian approach lies in their respective treatment of the unknown parameters. Frequentist interpretation of Bandit problem assumes there is a fixed unknown θ_a associated with each arm, while a Bayesian interpretation assumes that θ_a itself is generated from a prior distribution Ψ over the set Θ. These two views also have different goals:

1. An agent that is optimal in the Frequentist sense chooses to minimize R_θ^π for all $\theta \in \Theta$, that is, find $\pi^* = \arg\min_\pi \max_\theta E_\theta^\pi[R_\theta]$.

2. An agent optimal in the Bayesian sense chooses to minimize the expected regret $E_{\Psi,\theta}^\pi[R_\theta]$, that is, find $\pi^* = \arg\min_\pi E_{\Psi,\theta}^\pi[R_\theta]$ for known prior Ψ. The more general question of finding $\pi^* = \arg\min_\pi \max_\Psi E_{\Psi,\theta}^\pi[R_\theta]$ is open, but we expect the bounds to be worse than the Frequentist ones because nature's optimal strategy may not be deterministic.

The celebrated Gittins index [Gittins, 1979] gives a solution to the Bayesian formulation of discounted Bandit problem. According to Kaufmann [2014] (sec. 1.3.4), Chang and Lai [1987] developed a closed-form asymptotic approximations to Gittins index. These approximations take the form of explicit bonus terms added to point estimate of the mean $\hat{\theta}$, for Gaussian bandits. Bonus terms added to point estimates are essential in proving Frequentist optimality for Bandits [Cappé et al., 2013] and MDP [Jaksch et al., 2010, Tossou et al., 2019], using typical arguments of Optimism in Face of Uncertainty (OFU) principle. She shows experimentally that Finite Horizon Gitins index actually performs much better in terms of the Frequentist regret, against those designed to be optimal for it (e.g. KL-UCB). We *conjecture* that even for RL in MDP, the Bayes-optimal policy would inherently explore enough without need for additional explicit optimism. This connection may also be seen in a previous attempt of Duff and Barto [1997], where they try to use Gittins index for BAMDP.

2.2.2 Model-based Bayesian RL

BAMDP was initially investigated by Silver [1963] and Martin [1967]. The problem of computational intractability of the Bayes-optimal solution motivated researchers to design approximate techniques. These are referred to as

Bayesian RL (BRL) algorithms. BRL algorithms discussed here are all model-based. Ghavamzadeh et al. [2015] compiles a survey of BRL algorithms. They can be further classified based on whether they directly approximate the belief tree structure (lookahead) or not (myopic). Hence we first discuss BRL algorithms based on their design and then their theoretical motivation.

We classify them in two categories based on their functioning: Myopic and Lookahead.

Myopic: Myopic algorithms do not explicitly take into account the information to be gained by future actions, and yet may still be able to learn efficiently. One example of such an algorithm is Thompson sampling [Thompson, 1933], which maintains a posterior distribution over models, samples one of them and then chooses the optimal policy for the sample. A reformulation of this for BRL was investigated in [Strens, 2000]. The Best Of Sampled Set (BOSS) [Asmuth et al., 2009] algorithm generalizes this idea to a multi sample optimistic approach. BEB [Kolter and Ng, 2009] at the first look, seems to work directly on Bayesian value function (eq. 2.5), but it simply adds an explicit bonus term to the mean MDP estimates of the state-action value. Similar to BEB, MMBI [Dimitrakakis, 2011] assumes constant belief and therefore rolls-in the hyper-state value into the state-action value, but unlike BEB, it directly approximates the Bayesian value function. This assumption removes the exponential dependence (due to path dependent belief) on planning horizon. Then, backward induction is performed using value of the next step optimal adaptive policy. The final output is the stationary policy obtained at the root through backward induction[5]. Take home point is that, even for constant belief, mean MDP policy is not the best adaptive policy.

Lookahead: Lookahead algorithms take into account the effect of their future actions towards their knowledge about the environment and quantify its benefit for current decision making. The simplest algorithm is to calculate and solve the BAMDP up to some horizon T, as outlined in Algorithm 1 and illustrated in Figure 2.1. Sparse sampling [Kearns et al., 1999] is a simple modification to it, which instead only iterates over a set of sampled states. Kearns algorithm when

[5] Although it shouldn't be hard to store the intermediate optimal constant-belief adaptive policy, since its computed (step-8, Algo.1) anyways.

applied to BAMDP belief tree[6] would still have to consider all primitive actions. Wang et al. [2005] improved upon this by using Thompson sampling to only consider a subset of promising actions. The high branching factor of belief tree still makes planning with a deep horizon computationally expensive. Thus more scalable algorithms, such as BFS3 [Asmuth and Littman, 2011], BOLT [Araya et al., 2012] and BAMCP [Guez et al., 2012], were proposed. Similar to [Wang et al., 2005], BFS3 also selects a subset of actions but with an optimistic action selection strategy, though the backups are still performed using Bellman equation. BOLT includes optimism in the transition function instead. BAMCP takes a Monte-Carlo approach to sparse lookahead in belief-augmented version of Markov decision process. BAMCP also uses optimism for action selection. Unlike BFS3, the next set of hyper-states are sampled from an MDP sampled at the root[7]. Since posterior inference is expensive for any non-trivial belief model, BAMCP further applies lazy sampling and rollout policy, inspired by their application in tree search problems [Kocsis and Szepesvári, 2006].

Analysis: We discuss here BEB and BOLT, which have theoretical results similar to us. Both are PAC-BAMDP and derive their result by achieving certain level of certainty about (s, a) tuples, similar to Kearns and Singh [1998] who define such tuples as 'known'. BEB's authors rely on how many (s, a) tuples are already 'known' to prove their result. They go on to prove that both finite horizon Bayesian-optimal policy and BEB's policy decay the exploration rate so fast that they are not PAC-MDP[8], providing with a 3-state MDP counter example. BOLT's authors prove that if the probability of unknown states is small enough, BOLT's Bayes value function is close to optimal, if not, then such events (of seeing 'unkown' (s, a) tuples) occur only a limited number of times (by contradiction), and that this required amount of exploration is ensured by BOLT's optimism. The authors also extend BEB's PAC-BAMDP result to infinite horizon case. The problem with the above approach of 'knowing' all the (s, a) tuples enough is that the Bayes-optimal policy no longer remains interesting; Martin

[6] We freely use the term 'tree' or 'belief tree' to denote the planning tree generated by the algorithms in the hyper-state space of BAMDP.

[7] Note that ideally the next observations should be sampled from the $P(s_{t+1}|\omega_t)$ instead of $P(s_{t+1}|\omega_{t_o})$, i.e. the next-state marginal at the root belief.

[8] This is first of any result on Frequentist nature of Bayes-optimal policy.

[1967] proves that in such a case, the Bayes-optimal policy approaches the optimal policy of underlying MDP. This leads to unfaithful approximation of the true Bayes-optimal policy. A better approach is to prove near optimality without such assumption, e.g., BOP [Fonteneau et al., 2013] uses upper bound on Bayesian value for branch-and-bound tree search. In practice, their exponential dependence on the branching factor is still quite strong (proposition.2). We on the other hand, beat state-of-the-art by making reasonable assumption on the belief convergence rate. We formalize this idea in [Grover et al., 2019], which approaches Bayes optimality with constraint only on the computational budget. BAMCP is also successful in practice due to computational reasons, that is, it is a Monte Carlo technique which samples large number of nodes from belief tree. Its result is although only asymptotically optimal.

2.2.3 POMDP literature

Partially Observable MDP (POMDP) is a well studied [Åström, 1965, Sondik, 1978, Kaelbling et al., 1998] generalized MDP. Consider the field of robotics, where even though the dynamics of a robot may be known, there is uncertainty in the sensory input, i.e, the current state of the environment. Such problems are modeled by a POMDP. It is an MDP where we maintain a distribution over the possible states and plan accordingly. A natural idea would be to apply the already existing literature of POMDP to BAMDP. Significant effort was made by Duff [2002], where he shows almost mechanical translation of the POMDP alpha-vector[9] formulation to BAMDP (sec. 5.3). He notes that due to the belief having continuous support in BAMDP, in contrast to the discrete support (over states) in POMDP, fundamental differences[10] arise in the application of Monahan's algorithm [Monahan, 1982]. He comments (sec. 5.3.3) how the alpha functions due to backward induction are just a mixture of alpha functions at previous iteration[11], by extension of which any closed set of functions representing the Bayesian value initially will imply a function from same family locally at the root belief. Therefore the "idea of characterizing the value function in terms of a finite set of elements generalizes from the POMDP case". Although he quickly

[9] A vector, $\alpha : [0, 1]^{|S|} \to \mathbb{R}$, compactly representing the Bayesian value over all belief space.
[10] Alpha vectors become alpha functions, their dot product with belief becomes integral.
[11] More precisely, reward is added to this mixture by the definition of Bellman operator.

points out how this approach is computationally infeasible: Exact methods for PODMP [Sondik, 1978, Kaelbling et al., 1998] crucially depend on eliminating the exponentially growing alpha vectors with respect to planning horizon, and for this, they a solve set of linear equation constraints. These constraints in BAMDP case turn into integral constraints, which usually don't have an easily computable solution. Hence, the curse of exponentially memory usage with respect to planning depth still remains. Poupart et al. [2006] claim to show that alpha functions in BAMDP are multivariate polynomials in shape, but their main Theorem only relies on backward induction a as proof. It is unclear to us how the initial alpha functions should be multivariate polynomials. Duff [2002] goes on to argue and develop a general finite-state (memory) controller method for both POMDP and BAMDP problem. This approach holds much promise and may be a future research direction of this thesis. One key observation usually missed is that belief convergence doesn't exist in POMDP[12] and hence we miss out by blindly using POMDP algorithms.

2.2.4 Bayesian value function

Estimating the Bayesian state-value function distribution directly is another interesting approach. Unlike BAMDP, the algorithms don't compute the hyper-state value function. Algorithms in this category either directly estimate the state-value distribution from the data or rely on Bayesian formulation of backward induction. They come in both model-free [Dearden et al., 1998, Engel et al., 2003] and model-based [Dearden et al., 1999, Dimitrakakis, 2011, Deisenroth et al., 2009] flavours. In model-based approach, the uncertainty in state-value is taken care by maintaining a distribution over models, although similarity with BAMDP ends there.

Model-free: Bayesian Q-learning [Dearden et al., 1998] attempts to model the Bayesian Q-value directly with a parametric distribution. They propose online learning using myopic Value of Perfect Information (VPI) as action selection strategy, which essentially gives the expected (belief averaged) advantage of choosing an action over the others. They then propose two ways, both based

[12] In BAMDP, the belief over transition probabilities converge as we plan, while no such analogy exits when considering belief simply over MDP states.

on bootstrapping, to address the crux of the problem: delayed rewards, i.e., no direct access to Q-value samples. In practice, their algorithm changes policy too often to actually get a good estimate by bootstrapping. They mention this problem in their follow-up [Dearden et al., 1998] "to avoid the problem faced by model-free exploration methods, that need to perform repeated actions to propagate values from one state to another". Reader is directed to section (2.4) of [Duff, 2002], for a survey of other similar non Bayesian attempts. Engel et al. [2003] propose a Gaussian Process prior on value function, combined with temporal difference motivated data likelihood. This is discussed further in [Dimitrakakis et al., 2020].

Model-based: Dearden et al. [1999] propose a model-based follow-up to address the problems in Bayesian Q-learning. The main algorithm (sec. 5.1) computes a Monte Carlo upper bound on the Bayesian value function which they take as substitute for optimal Bayesian Q-value[13]. They address the complexity of sampling and solving multiple models by two re-weighting approaches: Importance sampling (sec. 5.2) and Particle filtering (sec. 5.3). Although they do mention Bayesian Bellman update (sec. 5.4), it is not clearly described (it seems like a mean-field approximation), and they do not experimentally investigate it. PILCO [Deisenroth et al., 2013] develops an analytic method similar to [Dearden et al., 1999] in nature, where they use GP prior with an assumption of normal input (sec. 3.2, 4) to predict a closed form Bayesian estimate of the objective function (multi-step cost, eq. 11). They use analytical gradient for policy optimization (sec. 3.3). Deisenroth et al. [2009] take a more direct approach by developing backward induction for Gaussian Process(GP) priors over models.

In [Dimitrakakis et al., 2020] we develop a Bayesian backward induction framework for joint estimation of model and value function distributions.

[13]The state-action value function is commonly known as Q-value.

Chapter 3

Efficient Bayesian Reinforcement Learning

The following sections will outline the contributions of this thesis.

In Section (3.1), we develop a sampling based approach to belief tree approximation. We go a step further than previous works by sampling policies instead of actions to curb the branching factor and reduce complexity. We analyze it, giving a performance bound, and experimentally validate it on different discrete environments. My individual contribution is the development of the initial idea with my supervisor, implementation and experiments, and contributing to its theoretical analysis.

In Section (3.2), we propose a fully Bayesian, backward induction approach for joint estimation of model and value function distributions. My individual contribution to this work is the baseline implementation, verifying correct functioning of algorithms and drawing comparison to BAMDP techniques.

3.1 Deep, Sparse Sampling

We propose a novel BAMDP algorithm, called Deep, Sparse Sampling (DSS), with the help of insights developed in section (2.2.2). During planning, it focuses on reducing the branching factor by considering K-step policies instead of primitive actions. These policies are generated through (possibly approximate) Thompson sampling[1] over MDP models. This approach is rounded by using Sparse sampling [Kearns et al., 1999]. The reduced branching factor allows us to build a deeper tree. Figure 3.1 shows a planning tree expanded by DSS[2]. The intuition why this might be desirable is that if the belief changes slowly enough, an adaptive policy that is constructed out of K-step stationary policies will still be approximately optimal. This intuition is supported by the theoretical analysis: we prove that our algorithm results in nearly-optimal planning under certain mild assumptions regarding the belief. The freedom to choose a policy generator allows the algorithm scale smoothly: we choose Policy Iteration (PI) and a variant of Real Time Dynamic Programming (RTDP) depending on size of environments.

Algorithm:

The core idea of the DSS algorithm is to plan in the belief tree, not at the individual action level, but at the level of K-step policies. Figure 3.1 illustrates this concept graphically. Algorithm 2 is called with the current state s and belief β as input, with additional parameters controlling how the tree is approximated. The algorithm then generates the tree and calculates the value of each policy candidate recursively (for H stages or episodes), in the following manner:

1. Line 6: Generate N MDPs from the current belief β_t, and for each MDP μ_i use the policy generator $\mathcal{P} : \mu \to \pi$ to generate a policy π_i. This gives a policy set Π_β with $|\Pi_\beta| = N$.

2. Line 10-18: Run each policy for K steps, collecting total K-step discounted reward R in BAMDP. Note that we sample the reward and next-

[1] We refer to the optimal policy of the sampled model as Thompson sample(TS) policy.

[2] Subscripts denote the planning depth. Superscripts on hyper-state iterate policy and belief respectively.

Algorithm 2 DSS

1: **Parameters:** Number of stages H, steps K, no. of policies N, no. of samples per policy M, policy generator $\mathcal{P}$
2: **Input:** hyper-state $\omega_h = (s_h, \beta_h)$, depth h.
3: **if** $h = KH$ **then**
4: return $V(\omega_h) = 0$
5: **end if**
6: $\Pi_{\beta_h} = \{\mathcal{P}(\mu_i) | \mu_i \backsim \omega_h, i \in \mathbb{Z}, i \leq N\}$
7: **for** all $\pi \in \Pi_{\beta_h}$ **do**
8: $Q(\omega_h, \pi) = 0$
9: **for** 1 to M **do**
10: $R = 0, c = \gamma^h, k = 0$
11: $\omega_k = \omega_h, s_k = s_h, \beta_k = \beta_h, a_k = \pi(s_h)$
12: **for** $k = 1, \ldots, K$ **do**
13: $s_{k+1} \backsim \nu(\omega_{k+1} | \omega_k, a_k)$
14: $r_{k+1} \backsim \tau(r_{k+1} | \omega_k, a_k)$
15: $R \mathrel{+}= c \times r_{k+1}; c = c \times \gamma$
16: $\beta_{k+1} =$ UpdatePosterior(ω_k, s_{k+1}, a_k) (from eq. 2.2)
17: **end for**
18: $Q(\omega_h, \pi) \mathrel{+}= R +$ DSS$(\omega_K, h + K)$
19: **end for**
20: $Q(\omega_h, \pi) /= M$
21: **end for**
22: **return** $\arg\max_\pi Q(\omega_h, \pi)$

state from the marginal (Line 13-14), and also update the posterior (Line 16).

3. Line 19-21: Make recursive call to DSS at the end of K steps. Repeat the process just described for M times. This gives an M-sample estimate of that policy's utility v^π_β.

Note that the fundamental control unit that we are trying to find here is a policy, hence Q-values are defined over (ω_t,π) tuples. Since we now have policies at any given tree node, we re-branch only after running those policies for K steps.

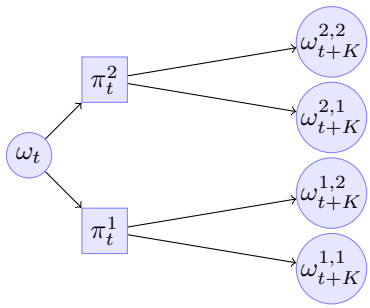

Figure 3.1: Deeper & Sparser tree expansion.

Hence we can increase the effective depth of the belief tree upto HK for the same computational budget. This allows for deeper lookahead and ensures that the approximation error propagated is also smaller as the error is discounted by γ^{HK} instead of γ^H. We elaborate this effect in the analysis below.

Analysis:

Since our goal is to prove the near-Bayes optimality of DSS, we focus on the effects of DSS parameters for planning in belief tree. DSS eliminates the necessity to try all actions at every node by making certain assumptions about the belief:

Assumption 1. *The belief β_h in the planning tree is such that $\epsilon_h \leq \epsilon_0/h$, where $h \geq 1$, $\epsilon_h = ||\hat{\beta}_h - \beta_h||_1$ and $\hat{\beta}_h$ is the constant belief approximation at the start of episode h.*

The first assumption states that as we go deeper in the planning tree, the belief error reduces. The intuition is that if the belief concentrates at a certain rate, then so does error of Bayes utility for any Markov policy, by the virtue of its definition.

Assumption 2. *Bounded correlation: Given some constant $C \in \mathbb{R}^+$, $D(\mu, \mu') \triangleq \max_{s,a} \|P_\mu(\cdot \mid s, a) - P_{\mu'}(\cdot \mid s, a)\|_1$. We have:*

$$\beta_t(\mu)\beta_t(\mu') \leq \frac{C}{D(\mu, \mu')}$$

The second assumption states that belief correlation across similar MDPs is

higher than across dissimilar ones, due to inverse dependence on the distance between MDPs. This helps us proving bound for Bayesian value of DSS policy compared to K-step optimal policy.

Then, our algorithm finds a near-Bayes-optimal policy as stated in Theorem 1:

Theorem 1. *Under Assumptions 1 and 2,* $\forall s \in S$

$$v_\beta^{DS}(s) \geq v_\beta^*(s) - \left(2\epsilon_0 K \ln \frac{1}{1-\gamma^K} + \frac{2(KC+\gamma^K)}{(1-\gamma)}\right) - \sqrt{\frac{\ln M/\delta}{2N(1-\gamma)^2}}$$

with probability $1-\delta$*. Here,* T *is the horizon, divided by parameter* K *into* H *stages, i.e,* $T = KH$*. In addition, at each node of the sparse tree, we evaluate* N *policies for* M *times.*

At the same time, DSS is significantly less expensive than basic Sparse sampling [Kearns et al., 1999] which would take $O((|A|M)^T)$ calls to the generative BAMDP model, while it requires only $O((NM)^{T/K})$ calls for a T-horizon problem.

Experiments:

We refer the reader to the original publication [Grover et al., 2019] for comparisons to the current state-of-the-art.

3.2 Bayesian backward induction (BBI)

Simple BI:

Since this work is based on backward induction, we first consider the case of estimating the mean value function under MDP uncertainty. We show here how the most common approximation of Mean MDP is related to the BAMDP value function.

Consider backward induction equation for the Bayesian value:

$$\begin{aligned} & V_t^\pi(s_t, \beta_t) \\ &= \int_{\mathcal{M}} V_\mu^\pi(s_t)\beta_t(\mu)d\mu \\ &= \int_{\mathcal{M}} \mathbb{E}_\mu[r_{t+1}]\beta_t(\mu)d\mu + \gamma \int_{\mathcal{M}} \sum_{s' \in s_{t+1}} \mathbb{P}_\mu(s'|s_t, a_t) V_\mu^\pi(s')\beta_t(\mu)d\mu \\ &= \int_{\mathcal{M}} \mathbb{E}_\mu[r_{t+1}]\beta_t(\mu)d\mu + \gamma \sum_{s' \in s_{t+1}} \int_{\mathcal{M}} V_\mu^\pi(s')\, \mathbb{P}_\mu(s'|s_t, a_t)\beta_t(\mu)d\mu \end{aligned} \tag{3.1}$$

Mean MDP approximation for eq. (3.1) is obtained by taking mean-field approximation of the inner part of second term:

$$\int_{\mathcal{M}} V_\mu^\pi(s')\, \mathbb{P}_\mu(s'|s_t, a_t)\, \mathbb{P}(\mu)d\mu \approx \int_{\mathcal{M}} V_\mu^\pi(s')\, \mathbb{P}(\mu)d\mu \int_{\mathcal{M}} \mathbb{P}_\mu(s'|s_t, a_t)\, \mathbb{P}(\mu)d\mu$$

Define $\bar{V}(s') \triangleq \int_{\mathcal{M}} V_\mu^\pi(s')\, \mathbb{P}(\mu)$ and $\nu(s'|s, a) \triangleq \int_{\mathcal{M}} \mathbb{P}_\mu(s'|s, a)\, \mathbb{P}(\mu)d\mu$. Substituting above expression back in eq. (3.1):

$$\begin{aligned} & V_t^\pi(s_t, \beta_t) \\ &= \int_{\mathcal{M}} \mathbb{E}_\mu[r_{t+1}]\beta_t(\mu)d\mu + \gamma \sum_{s' \in s_{t+1}} \nu(s'|s_t, \pi_t)\bar{V}_t^\pi(s') \end{aligned} \tag{3.2}$$

Equation (3.2) gives the mean MDP approximation under constant belief assumption, since then we only keep track of $|S|$ values at each iteration. MMBI [Dimitrakakis, 2011] directly works on eq. (3.1) giving much better results. These approximations only give the mean of the state-value distribution, although it would be more useful to get the full distribution.

Distribution over value functions:

Consider the value function V, with $V = (V_1, \ldots, V_T)$ for finite-horizon problems, and some prior belief β over MDPs, and some previously collected data $D = (s_1, a_1, r_1, \ldots, s_t, a_t, r_t)$ from some policy π. Then the posterior value function distribution can be written in terms of the MDP posterior:

$$\mathbb{P}_\beta(V \mid D) = \int_{\mathcal{M}} \mathbb{P}_\mu(V)\, \mathrm{d}\beta(\mu \mid D). \tag{3.3}$$

Note that this is different from eq. (2.3) which gives the expected state-value under the model distribution, while here we get the full state-value distribution for the given belief. The empirical measure $\hat{P}^E_{MC}$ defined below corresponds to the standard Monte-Carlo estimate

$$\hat{P}^E_{MC}(B) \triangleq N_\mu^{-1} \sum_{k=1}^{K} \mathbb{1}\left\{\boldsymbol{v}^{(k)} \in B\right\}, \tag{3.4}$$

where $\mathbb{1}\{\}$ is the indicator function. In practice, this can be implemented via Algorithm 3. The problem with this approach is the computational cost associated with it.

Algorithm 3 Monte-Carlo Estimation of Value Function Distributions

1: Select a policy π.
2: **for** $k = 1, \ldots, N_\mu$ **do**
3: Sample an MDP $\mu^{(k)} \sim \beta$.
4: Calculate $\boldsymbol{v}^{(k)} = V^\pi_{\mu^{(k)},} \sim \beta$.
5: **end for**
6: **return** $\hat{P}_{MC}(\{\boldsymbol{v}^{(k)}\})$

Bayesian Backward Induction (BBI):

We propose a framework that inductively calculates $\mathbb{P}^\pi_\beta(V_{i+1} \mid D)$ from $\mathbb{P}^\pi_\beta(V_i \mid D)$ for $i \geq t$:

$$\mathbb{P}^\pi_\beta(V_i \mid D) = \int_{\mathcal{V}} \mathbb{P}^\pi_\beta(V_i \mid V_{i+1}, D) \,\mathrm{d}\, \mathbb{P}^\pi_\beta(V_{i+1} \mid D). \tag{3.5}$$

Let ψ_{i+1} be a (possibly approximate) representation of $\mathbb{P}^\pi_\beta(V_{i+1} \mid D)$. Then the remaining problem is to define the term $\mathbb{P}^\pi_\beta(V_i \mid V_{i+1}, D)$ appropriately and calculate the complete distribution.

Link distribution: $\mathbb{P}(V_i \mid V_{i+1}, D)$

A simple idea for dealing with the term linking the two value functions is to marginalize over the MDP as follows:

$$\mathbb{P}^\pi_\beta(V_i \mid V_{i+1}, D) = \int_{\mathcal{M}} \mathbb{P}^\pi_\mu(V_i \mid V_{i+1}) \,\mathrm{d}\, \mathbb{P}^\pi_\beta(\mu \mid V_{i+1}, D). \tag{3.6}$$

This equality holds because given μ, V_i is uniquely determined by the policy π and V_{i+1} through the Bellman operator. However, it is crucial to note that $\mathbb{P}^\pi_\beta(\mu \mid V_{i+1}, D) \neq \mathbb{P}_\beta(\mu \mid D)$, as knowing the value function gives information about the MDP.[3]

In order to maintain a correct estimate of uncertainty, we must specify an appropriate conditional distribution $\mathbb{P}^\pi_\beta(\mu|V_{i+1}, D)$. We focus on the idea of maintaining an approximation ψ_i of value function distributions and combining this with the MDP posterior through an appropriate kernel, as detailed below.

Conditional MDP distribution: $\mathbb{P}(\mu|V_{i+1}, D)$

The other important design decision concerns the distribution $\mathbb{P}^\pi_\beta(\mu|V^{(k)}_{i+1}, D)$. Expanding this term, we obtain, for any subset of MDPs $A \subseteq \mathcal{M}$:

$$\mathbb{P}^\pi_\beta(\mu \in A \mid V_{i+1}, D) = \frac{\int_A \mathbb{P}^\pi_\mu(V_{i+1}) \,\mathrm{d}\beta(\mu \mid D)}{\int_{\mathcal{M}} \mathbb{P}^\pi_\mu(V_{i+1}) \,\mathrm{d}\beta(\mu \mid D)}, \tag{3.7}$$

since $\mathbb{P}^\pi_\mu(V_{i+1} \mid D) = \mathbb{P}^\pi_\mu(V_{i+1})$, as μ, π are sufficient for calculating V_{i+1}.

Inference and experiments:

We refer the reader to the full paper [Dimitrakakis et al., 2020] for details on inference procedure and experiments.

[3] Assuming otherwise results in a mean-field approximation.

Chapter 4

Concluding Remarks

In this thesis, we introduced a novel BAMDP algorithm, Deep Sparse Sampling (DSS) which is the author's primary contribution. We showed its superior performance experimentally and also analyzed its theoretical properties. We also jointly proposed a Bayesian backward induction approach for estimating the state-value and model distributions.

A natural extension to this thesis would be in the direction of bounded memory (*insufficient statistics*) controllers. We believe it to be a promising avenue to prove stronger results for BAMDP algorithms. While a general regret bound for the Bayes-optimal policy is an open question, counter-intuitively, the analysis with insufficient statistics may be simpler, since there will only be a finite number of beliefs. The main idea for BAMDP would be to optimally tune the branching factor and depth of the planning tree depending on accuracy of the statistics. We can then analyze the effect on optimality relative to an oracle with sufficient statistics. Next, we can analyze how close approximate algorithms (DSS, Sparse sampling etc.) are to the best insufficient statistic policy under different computational constraints. For completeness, we shall also analyze the regret for insufficient-statistic versions of the upper-bound algorithms relying on concentration inequalities. We expect to get qualitatively different bounds compared to standard analysis, due to the use of insufficient statistics.

Bibliography

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

J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *UAI 2009*, 2009.

John Asmuth and Michael L Littman. Approaching bayes-optimalilty using monte-carlo tree search. In *Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany*, 2011.

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10 (1):174–205, 1965.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, Gilles Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

Fu Chang and Tze Leung Lai. Optimal stopping and dynamic allocation. *Advances in Applied Probability*, 19(4):829–853, 1987.

Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998. URL `citeseer.ist.psu.edu/dearden98bayesian.html`.

Richard Dearden, Nir Friedman, and David Andre. Model based Bayesian exploration. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages

150–159, San Francisco, CA, July 30–August 1 1999. Morgan Kaufmann, San Francisco, CA.

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2013.

M.P. Deisenroth, C.E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.

Christos Dimitrakakis. Robust bayesian reinforcement learning through tight lower bounds. In *European Workshop on Reinforcement Learning*, page arXiv:1106.3651v2. Springer, 2011.

Christos Dimitrakakis, Hannes Eriksson, Emilio Jorge, Divya Grover, and Debabrota Basu. Inferential induction: Joint bayesian estimation of mdps and value functions. *arXiv preprint arXiv:2002.03098*, 2020.

Michael O Duff and Andrew G Barto. Local bandit approximation for optimal learning problems. In *Advances in Neural Information Processing Systems*, pages 1019–1025, 1997.

Michael O'Gordon Duff. *Optimal Learning Computational Procedures for Bayes-adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *ICML 2003*, 2003.

Raphael Fonteneau, Lucian Buşoniu, and Rémi Munos. Optimistic planning for belief-augmented markov decision processes. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 77–84. IEEE, 2013.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. 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)*, 41(2):148–164, 1979.

Divya Grover and Christos Dimitrakakis. Deeper and sparser sampling. *Exploration in Reinforcement Learning Workshop, ICML*, 2018.

Divya Grover, Debabrota Basu, and Christos Dimitrakakis. Bayesian reinforcement learning via deep, sparse sampling. *arXiv preprint arXiv:1902.02661*, 2019.

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

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

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Emilie Kaufmann. *Analyse de stratégies Bayésiennes et fréquentistes pour l'allocation séquentielle de ressources*. PhD thesis, Paris, ENST, 2014.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. In *Proc. 15th International Conf. on Machine Learning*, pages 260–268. Morgan Kaufmann, San Francisco, CA, 1998. URL `citeseer.ist.psu.edu/kearns98nearoptimal.html`.

Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In Thomas Dean, editor, *IJCAI*, pages 1324–1231. Morgan Kaufmann, 1999. ISBN 1-55860-613-0.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

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

James John Martin. *Bayesian decision problems and Markov chains*. Wiley, 1967.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

George E Monahan. State of the arta survey of partially observable markov decision processes: theory, models, and algorithms. *Management science*, 28(1):1–16, 1982.

P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *ICML 2006*, pages 697–704. ACM Press New York, NY, USA, 2006.

Marting L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New Jersey, US, 1994.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Edward A Silver. Markovian decision processes with uncertain transition probabilities or rewards. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESEARCH CENTER, 1963.

Edward J Sondik. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations research*, 26 (2):282–304, 1978.

Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, pages 943–950, 2000.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

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.

Aristide Tossou, Debabrota Basu, and Christos Dimitrakakis. Near-optimal optimistic reinforcement learning using empirical bernstein inequalities. *arXiv preprint arXiv:1905.12425*, 2019.

Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *ICML '05*, pages 956–963, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: http://doi.acm.org/10.1145/1102351.1102472.