Journal of Machine Learning Research 12 (2011) 1655-1696 Submitted 10/08; Revised 11/10; Published 5/11

A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes

Stéphane Ross STEPHANEROSS@CMU.EDU
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA 15213

Joelle Pineau JPINEAU@CS.MCGILL.CA
School of Computer Science
McGill University
Montréal, PQ, Canada H3A 2A7

Brahim Chaib-draa CHAIB@IFT.ULAVAL.CA
Computer Science & Software Engineering Dept
Laval University
Québec, PQ, Canada G1K 7P4

Pierre Kreitmann PIERRE.KREITMANN@GMAIL.COM
Department of Computer Science
Stanford University
Stanford, CA, USA 94305

Editor: Satinder Baveja

Abstract

Bayesian learning methods have recently been shown to provide an elegant solution to the exploration-exploitation trade-off in reinforcement learning. However most investigations of Bayesian reinforcement learning to date focus on the standard Markov Decision Processes (MDPs). The primary focus of this paper is to extend these ideas to the case of partially observable domains, by introducing the Bayes-Adaptive Partially Observable Markov Decision Processes. This new framework can be used to simultaneously (1) learn a model of the POMDP domain through interaction with the environment, (2) track the state of the system under partial observability, and (3) plan (near-)optimal sequences of actions. An important contribution of this paper is to provide theoretical results showing how the model can be finitely approximated while preserving good learning performance. We present approximate algorithms for belief tracking and planning in this model, as well as empirical results that illustrate how the model estimate and agent's return improve as a function of experience.

Keywords: reinforcement learning, Bayesian inference, partially observable Markov decision processes

1. Introduction

Robust decision-making is a core component of many autonomous agents. This generally requires that an agent evaluate a set of possible actions, and choose the best one for its current situation. In many problems, actions have long-term consequences that must be

considered by the agent; it is not useful to simply choose the action that looks the best in the immediate situation. Instead, the agent must choose its actions by carefully trading off their short-term and long-term costs and benefits. To do so, the agent must be able to predict the consequences of its actions, in so far as it is useful to determine future actions. In applications where it is not possible to predict exactly the outcomes of an action, the agent must also consider the uncertainty over possible future events.

Probabilistic models of sequential decision-making take into account such uncertainty by specifying the chance (probability) that any future outcome will occur, given any current configuration (state) of the system, and action taken by the agent. However, if the model used does not perfectly fit the real problem, the agent risks making poor decisions. This is currently an important limitation in practical deployment of autonomous decision-making agents, since available models are often crude and incomplete approximations of reality. Clearly, learning methods can play an important role in improving the model as experience is acquired, such that the agent's future decisions are also improved.

In the past few decades, Reinforcement Learning (RL) has emerged as an elegant and popular technique to handle sequential decision problems when the model is unknown (Sutton and Barto, 1998). Reinforcement learning is a general technique that allows an agent to learn the best way to behave, that is, such as to maximize expected return, from repeated interactions in the environment. A fundamental problem in RL is that of exploration-exploitation: namely, how should the agent chooses actions during the learning phase, in order to both maximize its knowledge of the model as needed to better achieve the objective later (i.e., *explore*), and maximize current achievement of the objective based on what is already known about the domain (i.e., *exploit*). Under some (reasonably general) conditions on the exploratory behavior, it has been shown that RL eventually learns the optimal action-select behavior. However, these conditions do not specify how to choose actions such as to maximize utilities throughout the life of the agent, including during the learning phase, as well as beyond.

Model-based Bayesian RL is an extension of RL that has gained significant interest from the AI community recently as it provides a principled approach to tackle the problem of exploration-exploitation during learning and beyond, within the standard Bayesian inference paradigm. In this framework, prior information about the problem (including uncertainty) is represented in parametric form, and Bayesian inference is used to incorporate any new information about the model. Thus the exploration-exploitation problem can be handled as an explicit sequential decision problem, where the agent seeks to maximize future expected return with respect to its current uncertainty on the model. An important limitation of this approach is that the decision-making process is significantly more complex since it involves reasoning about all possible models *and* courses of action. In addition, most work to date on this framework has been limited to cases where full knowledge of the agent's state is available at every time step (Dearden et al., 1999; Strens, 2000; Duff, 2002; Wang et al., 2005; Poupart et al., 2006; Castro and Precup, 2007; Delage and Mannor, 2007).

The primary contribution of this paper is an extension of the model-based Bayesian reinforcement learning to partially observable domains with discrete representations.[1] In support of this, we introduce a new mathematical model, called the *Bayes-Adaptive POMDP*

1. A preliminary version of this model was described by Ross et al. (2008a). The current paper provides an in-depth development of this model, as well as novel theoretical analysis and new empirical results.

(BAPOMDP). This is a model-based Bayesian RL approach, meaning that the framework maintains a posterior over the parameters of the underlying POMDP domain.[2] We derive optimal algorithms for belief tracking and finite-horizon planning in this model. However, because the size of the state space in a BAPOMD can be countably infinite, these are, for all practical purposes, intractable. We therefore dedicate substantial attention to the problem of approximating the BAPOMDP model. We provide theoretical results for bounding the state space while preserving the value function. These results are leveraged to derive a novel belief monitoring algorithm, which is used to maintain a posterior over both model parameters, and state of the system. Finally, we describe an online planning algorithm which provides the core sequential decision-making component of the model. Both the belief tracking and planning algorithms are parameterized so as to allow a trade-off between computational time and accuracy, such that the algorithms can be applied in real-time settings.

An in-depth empirical validation of the algorithms on challenging real-world scenarios is outside the scope of this paper, since our focus here is on the theoretical properties of the exact and approximative approaches. Nonetheless we elaborate a tractable approach and characterize its performance in three contrasting problem domains. Empirical results show that the BAPOMDP agent is able to learn good POMDP models and improve its return as it learns better model estimates. Experiments on the two smaller domains illustrate performance of the novel belief tracking algorithm, in comparison to the well-known Monte-Carlo approximation methods. Experiments on the third domain confirm good planning and learning performance on a larger domain; we also analyze the impact of the choice of prior on the results.

The paper is organized as follows. Section 2 presents the models and methods necessary for Bayesian reinforcement learning in the fully observable case. Section 3 extends these ideas to the case of partially observable domains, focusing on the definition of the BAPOMDP model and exact algorithms. Section 4 defines a finite approximation of the BAPOMDP model that could be used to be solved by finite offline POMDP solvers. Section 5 presents a more tractable approach to solving the BAPOMDP model based on online POMDP solvers. Section 6 illustrates the empirical performance of the latter approach on sample domains. Finally, Section 7 discusses related Bayesian approaches for simultaneous planning and learning in partially observable domains.

2. Background and Notation

In this section we discuss the problem of model-based Bayesian reinforcement learning in the fully observable case, in preparation for the extension of these ideas to the partially observable case which is presented in Section 3. We begin with a quick review of Markov Decision Processes. We then present the models and methods necessary for Bayesian RL in MDPs. This literature has been developing over the last decade, and we aim to provide a brief but comprehensive survey of the models and algorithms in this area. Readers interested in a more detailed presentation of the material should seek additional references (Sutton and Barto, 1998; Duff, 2002).

2. This is in contrast to model-free Bayesian RL approaches, which maintain a posterior over the value function, for example, Engel et al. (2003, 2005); Ghavamzadeh and Engel (2007b).

2.1 Markov Decision Processes

We consider finite MDPs as defined by the following n-tuple (S, A, T, R, γ):

States: S is a finite set of states, which represents all possible configurations of the system. A state is essentially a sufficient statistic of what occurred in the past, such that what will occur in the future only depends on the current state. For example, in a navigation task, the state is usually the current position of the agent, since its next position usually only depends on the current position, and not on previous positions.

Actions: A is a finite set of actions the agent can make in the system. These actions may influence the next state of the system and have different costs/payoffs.

Transition Probabilities: $T : S \times A \times S \to [0, 1]$ is called the transition function. It models the uncertainty on the future state of the system. Given the current state s, and an action a executed by the agent, $T^{sas'}$ specifies the probability $\Pr(s'|s, a)$ of moving to state s'. For a fixed current state s and action a, $T^{sa\cdot}$ defines a probability distribution over the next state s', such that $\sum_{s' \in S} T^{sas'} = 1$, for all (s, a). The definition of T is based on the *Markov assumption*, which states that the transition probabilities only depend on the current state and action, that is, $\Pr(s_{t+1} = s'|a_t, s_t, \ldots, a_0, s_0) = \Pr(s_{t+1} = s'|a_t, s_t)$, where a_t and s_t denote respectively the action and state at time t. It is also assumed that T is time-homogenous, that is, the transition probabilities do not depend on the current time: $\Pr(s_{t+1} = s'|a_t = a, s_t = s) = \Pr(s_t = s'|a_{t-1} = a, s_{t-1} = s)$ for all t.

Rewards: $R : S \times A \to \mathbb{R}$ is the function which specifies the reward $R(s, a)$ obtained by the agent for doing a particular action a in current state s. This models the immediate costs (negative rewards) and payoffs (positive rewards) incurred by performing different actions in the system.

Discount Factor: $\gamma \in [0, 1)$ is a discount rate which allows a trade-off between short-term and long-term rewards. A reward obtained t-steps in the future is discounted by the factor γ^t. Intuitively, this indicates that it is better to obtain a given reward now, rather than later in the future.

Initially, the agent starts in some initial state, $s_0 \in S$. Then at any time t, the agent chooses an action $a_t \in A$, performs it in the current state s_t, receives the reward $R(s_t, a_t)$ and moves to the next state s_{t+1} with probability $T^{s_t a_t s_{t+1}}$. This process is iterated until termination; the task horizon can be specified *a priori*, or determined by the discount factor.

We define a **policy**, $\pi : S \to A$, to be a mapping from states to actions. The optimal policy, denoted π^*, corresponds to the mapping which maximizes the expected sum of discounted rewards over a trajectory. The **value** of the optimal policy is defined by Bellman's equation:

$$V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} T^{sas'} V^*(s') \right].$$

The optimal policy at a given state, $\pi^*(s)$, is defined to be the action that maximizes the value at that state, $V^*(s)$. Thus the main objective of the MDP framework is to accurately

estimate this value function, so as to then obtain the optimal policy. There is a large literature on the computational techniques that can be leveraged to solve this problem. A good starting point is the recent text by Szepesvari (2010).

A key aspect of reinforcement learning is the issue of *exploration*. This corresponds to the question of determining how the agent should choose actions while learning about the task. This is in contrast to the phase called *exploitation*, through which actions are selected so as to maximize expected reward with respect to the current value function estimate. The issues of value function estimation and exploration are assumed to be orthogonal in much of the MDP literature. However in many applications, where data is expensive or difficult to acquire, it is important to consider the rewards accumulated during the learning phase, and to try to take this cost-of-learning into account in the optimization of the policy.

In RL, most practical work uses a variety of heuristics to balance the exploration and exploitation, including for example the well-known ϵ-greedy and Boltzmann strategies. The main problem with such heuristic methods is that the exploration occurs randomly and is not focused on what needs to be learned.

More recently, it has been shown that it is possible for an agent to reach near-optimal performance with high probability using only a polynomial number of steps (Kearns and Singh, 1998; Brafman and Tennenholtz, 2003; Strehl and Littman, 2005), or alternately to have small regret with respect to the optimal policy (Auer and Ortner, 2006; Tewari and Bartlett, 2008; Auer et al., 2009). Such theoretical results are highly encouraging, and in some cases lead to algorithms which exhibit reasonably good empirical performance.

2.2 Bayesian Learning

Bayesian Learning (or Bayesian Inference) is a general technique for learning the unknown parameters of a probability model from observations generated by this model. In Bayesian learning, a probability distribution is maintained over all possible values of the unknown parameters. As observations are made, this probability distribution is updated via Bayes' rule, and probability density increases around the most likely parameter values.

Formally, consider a random variable X with probability density $f_{X|\Theta}$ over its domain $\mathcal{X}$ parameterized by the unknown vector of parameters Θ in some parameter space $\mathcal{P}$. Let $X_1, X_2, \cdots, X_n$ be a random i.i.d. sample from $f_{X|\Theta}$. Then by Bayes' rule, the posterior probability density $g_{\Theta|X_1,X_2,\ldots,X_n}(\theta|x_1, x_2, \ldots, x_n)$ of the parameters $\Theta = \theta$, after the observations of $X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n$, is:

$$g_{\Theta|X_1,X_2,\ldots,X_n}(\theta|x_1, x_2, \ldots, x_n) = \frac{g_\Theta(\theta) \prod_{i=1}^n f_{X|\Theta}(x_i|\theta)}{\int_{\mathcal{P}} g_\Theta(\theta') \prod_{i=1}^n f_{X|\Theta}(x_i|\theta') d\theta'},$$

where $g_\Theta(\theta)$ is the prior probability density of $\Theta = \theta$, that is, g_Θ over the parameter space $\mathcal{P}$ is a distribution that represents the initial belief (or uncertainty) on the values of Θ. Note that the posterior can be defined recursively as follows:

$$g_{\Theta|X_1,X_2,\ldots,X_n}(\theta|x_1, x_2, \ldots, x_n) = \frac{g_{\Theta|X_1,X_2,\ldots,X_{n-1}}(\theta|x_1, x_2, \ldots, x_{n-1}) f_{X|\Theta}(x_n|\theta)}{\int_{\mathcal{P}} g_{\Theta|X_1,X_2,\ldots,X_{n-1}}(\theta'|x_1, x_2, \ldots, x_{n-1}) f_{X|\Theta}(x_n|\theta') d\theta'},$$

so that whenever we get the n^{th} observation of X, denoted x_n, we can compute the new posterior distribution $g_{\Theta|X_1,X_2,\ldots,X_n}$ from the previous posterior $g_{\Theta|X_1,X_2,\ldots,X_{n-1}}$.

In general, updating the posterior distribution $g_{\Theta|X_1,X_2,\ldots,X_n}$ is difficult due to the need to compute the normalization constant $\int_{\mathcal{P}} g_\Theta(\theta) \prod_{i=1}^{n} f_{X|\Theta}(x_i|\theta) d\theta$. However, for conjugate family distributions, updating the posterior can be achieved very efficiently with a simple update of the parameters defining the posterior distribution (Casella and Berger, 2001).

Formally, consider a particular class $\mathcal{G}$ of prior distributions over the parameter space $\mathcal{P}$, and a class $\mathcal{F}$ of likelihood functions $f_{X|\Theta}$ over $\mathcal{X}$ parameterized by parameters $\Theta \in \mathcal{P}$, then $\mathcal{F}$ and $\mathcal{G}$ are said to be conjugate if for any choice of prior $g_\Theta \in \mathcal{G}$, likelihood $f_{X|\Theta} \in \mathcal{F}$ and observation $X = x$, the posterior distribution $g_{\Theta|X}$ after observation of $X = x$ is also in $\mathcal{G}$.

For example, the Beta distribution[3] is conjugate to the Binomial distribution.[4] Consider $X \sim Binomial(n, p)$ with unknown probability parameter p, and consider a prior $Beta(\alpha, \beta)$ over the unknown value of p. Then following an observation $X = x$, the posterior over p is also Beta distributed and is defined by $Beta(\alpha + x, \beta + n - x)$.

Another important issue with Bayesian methods is the need to specify a prior. While the influence of the prior tends to be negligible when provided with a large amount of data, its choice is particularly important for any inference and decision-making performed when only a small amount of data has been observed. In many practical problems, informative priors can be obtained from domain knowledge. For example many sensors and actuators used in engineering applications have specified confidence intervals on their accuracy provided by the manufacturer. In other applications, such as medical treatment design or portfolio management, data about the problem may have been collected for other tasks, which can guide the construction of the prior.

In the absence of any knowledge, uninformative priors can be specified. Under such priors, any inference done *a posteriori* is dominated by the data, that is, the influence of the prior is minimal. A common uninformative prior consists of using a distribution that is constant over the whole parameter space, such that every possible parameter has equal probability density. From an information theoretic point of view, such priors have maximum entropy and thus contain the least amount of information about the true parameter (Jaynes, 1968). However, one problem with such uniform priors is that typically, under different re-parameterization, one has different amounts of information about the unknown parameters. A preferred uninformative prior, which is invariant under reparameterization, is Jeffreys' prior (Jeffreys, 1961).

The third issue of concern with Bayesian methods concerns the convergence of the posterior towards the true parameter of the system. In general, the posterior density concentrates around the parameters that have highest likelihood of generating the observed data in the limit. For finite parameter spaces, and for smooth families with continuous finite dimensional parameter spaces, the posterior converges towards the true parameter as long as the prior assigns non-zero probability to every neighborhood of the true parameter. Hence in practice, it is often desirable to assign non-zero prior density over the full parameter space.

3. $Beta(\alpha, \beta)$ is defined by the density function $f(p|\alpha, \beta) \propto p^{\alpha-1}(1-p)^{\beta-1}$ for $p \in [0, 1]$ and parameters $\alpha, \beta \geq 0$.
4. $Binomial(n, p)$ is defined by the density function $f(k|n, p) \propto p^k(1-p)^{n-k}$ for $k \in \{0, 1, \ldots, n\}$ and parameters $p \in [0, 1]$,$n \in \mathbb{N}$.

It should also be noted that if multiple parameters within the parameter space can generate the observed data with equal likelihood, then the posterior distribution will usually be multimodal, with one mode surrounding each equally likely parameter. In such cases, it may be impossible to identify the true underlying parameter. However for practical purposes, such as making predictions about future observations, it is sufficient to identify any of the equally likely parameters.

Lastly, another concern is how fast the posterior converges towards the true parameters. This is mostly influenced by how far the prior is from the true parameter. If the prior is poor, that is, it assigns most probability density to parameters far from the true parameters, then it will take much more data to learn the correct parameter than if the prior assigns most probability density around the true parameter. For such reasons, a safe choice is to use an uninformative prior, unless some data is already available for the problem at hand.

2.3 Bayesian Reinforcement Learning in Markov Decision Processes

Work on model-based Bayesian reinforcement learning dates back to the days of Bellman, who studied this problem under the name of Adaptive control processes (Bellman, 1961). An excellent review of the literature on model-based Bayesian RL is provided in Duff (2002). This paper outlines, where appropriate, more recent contributions in this area.

As a side note, model-free BRL methods also exist (Engel et al., 2003, 2005; Ghavamzadeh and Engel, 2007a,b). Instead of representing the uncertainty on the model, these methods explicitly model the uncertainty on the value function or optimal policy. These methods must often rely on heuristics to handle the exploration-exploitation trade-off, but may be useful in cases where it is easier to express prior knowledge about initial uncertainty on the value function or policy, rather than on the model.

The main idea behind model-based BRL is to use Bayesian learning methods to learn the unknown model parameters of the system, based on what is observed by the agent in the environment. Starting from a prior distribution over the unknown model parameters, the agent updates a posterior distribution over these parameters as it performs actions and gets observations from the environment. Under such a Bayesian approach, the agent can compute the best action-selection strategy by finding the one that maximizes its future expected return under the current posterior distribution, but also considering how this distribution will evolve in the future under different possible sequences of actions and observations.

To formalize these ideas, consider an MDP (S, A, T, R, γ), where S, A and R are known, and T is unknown. Furthermore, assume that S and A are finite. The unknown parameters in this case are the transition probabilities, $T^{sas'}$, for all $s, s' \in S$, $a \in A$. The model-based BRL approach to this problem is to start off with a prior, g, over the space of transition functions, T. Now let $\bar{s}_t = (s_0, s_1, \ldots, s_t)$ and $\bar{a}_{t-1} = (a_0, a_1, \ldots, a_{t-1})$ denote the agent's history of visited states and actions up to time t. Then the posterior over transition functions after this sequence is defined by:

$$\begin{aligned} g(T|\bar{s}_t, \bar{a}_{t-1}) &\propto g(T) \prod_{i=0}^{t-1} T^{s_i a_i s_{i+1}} \\ &\propto g(T) \prod_{s \in S, a \in A} \prod_{s' \in S} (T^{sas'})^{N^a_{s,s'}(\bar{s}_t, \bar{a}_{t-1})}, \end{aligned}$$

where $N^a_{s,s'}(\bar{s}_t, \bar{a}_{t-1}) = \sum_{i=0}^{t-1} I_{\{(s,a,s')\}}(s_i, a_i, s_{i+1})$ is the number of times[5] the transition (s, a, s') occurred in the history $(\bar{s}_t, \bar{a}_{t-1})$. As we can see from this equation, the likelihood $\prod_{s\in S, a\in A} \prod_{s'\in S} (T^{sas'})^{N^a_{s,s'}(\bar{s}_t, \bar{a}_{t-1})}$ is a product of $|S||A|$ independent Multinomial[6] distributions over S. Hence, if we define the prior g as a product of $|S||A|$ independent priors over each distribution over next states $T^{sa\cdot}$, that is, $g(T) = \prod_{s\in S, a\in A} g_{s,a}(T^{sa\cdot})$, then the posterior is also defined as a product of $|S||A|$ independent posterior distributions: $g(T|\bar{s}_t, \bar{a}_{t-1}) = \prod_{s\in S, a\in A} g_{s,a}(T^{sa\cdot}|\bar{s}_t, \bar{a}_{t-1})$, where $g_{s,a}(T^{sa\cdot}|\bar{s}_t, \bar{a}_{t-1})$ is defined as:

$$g_{s,a}(T^{sa\cdot}|\bar{s}_t, \bar{a}_{t-1}) \propto g_{s,a}(T^{sa\cdot}) \prod_{s'\in S} (T^{sas'})^{N^a_{s,s'}(\bar{s}_t, \bar{a}_{t-1})}.$$

Furthermore, since the Dirichlet distribution is the conjugate of the Multinomial, it follows that if the priors $g_{s,a}(T^{sa\cdot})$ are Dirichlet distributions for all s, a, then the posteriors $g_{s,a}(T^{sa\cdot}|\bar{s}_t, \bar{a}_{t-1})$ will also be Dirichlet distributions for all s, a. The Dirichlet distribution is the multivariate extension of the Beta distribution and defines a probability distribution over discrete distributions. It is parameterized by a count vector, $\phi = (\phi_1, \ldots, \phi_k)$, where $\phi_i \geq 0$, such that the density of probability distribution $p = (p_1, \ldots, p_k)$ is defined as $f(p|\phi) \propto \prod_{i=1}^k p_i^{\phi_i - 1}$. If $X \sim Multinomial_k(p, N)$ is a random variable with unknown probability distribution $p = (p_1, \ldots, p_k)$, and $Dirichlet(\phi_1, \ldots, \phi_k)$ is a prior over p, then after the observation of $X = \mathbf{n}$, the posterior over p is $Dirichlet(\phi_1 + n_1, \ldots, \phi_k + n_k)$. Hence, if the prior $g_{s,a}(T^{sa\cdot})$ is $Dirichlet(\phi^a_{s,s_1}, \ldots, \phi^a_{s,s_{|S|}})$, then after the observation of history $(\bar{s}_t, \bar{a}_{t-1})$, the posterior $g_{s,a}(T^{sa\cdot}|\bar{s}_t, \bar{a}_{t-1})$ is $Dirichlet(\phi^a_{s,s_1} + N^a_{s,s_1}(\bar{s}_t, \bar{a}_{t-1}), \ldots, \phi^a_{s,s_{|S|}} + N^a_{s,s_{|S|}}(\bar{s}_t, \bar{a}_{t-1}))$. It follows that if $\phi = \{\phi^a_{s,s'} | a \in A, s, s' \in S\}$ represents the set of all Dirichlet parameters defining the current prior/posterior over T, then if the agent performs a transition (s, a, s'), the posterior Dirichlet parameters ϕ' after this transition are simply defined as:

$$\begin{aligned} \phi'^a_{s,s'} &= \phi^a_{s,s'} + 1, \\ \phi'^{a'}_{s'',s'''} &= \phi^{a'}_{s'',s'''}, \forall (s'', a', s''') \neq (s, a, s'). \end{aligned}$$

We denote this update by the function $\mathcal{U}$, where $\mathcal{U}(\phi, s, a, s')$ returns the set ϕ' as updated in the previous equation.

Because of this convenience, most authors assume that the prior over the transition function T follows the previous independence and Dirichlet assumptions (Duff, 2002; Dearden et al., 1999; Wang et al., 2005; Castro and Precup, 2007). We also make such assumptions throughout this paper.

2.3.1 Bayes-Adaptive MDP Model

The core sequential decision-making problem of model-based Bayesian RL can be cast as the problem of finding a policy that maps extended states of the form (s, ϕ) to actions $a \in A$, such as to maximize the long-term rewards of the agent. If this decision problem

5. We use $I()$ to denote the indicator function.
6. $Multinomial_k(p, N)$ is defined by the density function $f(\mathbf{n}|p, N) \propto \prod_{i=1}^k p_i^{n_i}$ for $n_i \in \{0, 1, \ldots, N\}$ such that $\sum_{i=1}^k n_i = N$, parameters $N \in \mathbb{N}$, and p is a discrete distribution over k outcomes.

can be modeled as an MDP over extended states (s, ϕ), then by solving this new MDP, we would find such an optimal policy. We now explain how to construct this MDP.

Consider a new MDP defined by the tuple (S', A, T', R', γ). We define the new set of states $S' = S \times \mathcal{T}$, where $\mathcal{T} = \{\phi \in \mathbb{N}^{|S|^2|A|} | \forall (s,a) \in S \times A, \sum_{s' \in S} \phi^a_{ss'} > 0\}$, and A is the original action space. Here, the constraints on the set $\mathcal{T}$ of possible count parameters ϕ are only needed to ensure that the transition probabilities are well defined. To avoid confusion, we refer to the extended states $(s, \phi) \in S'$ as hyperstates. Also note that the next information state ϕ' only depends on the previous information state ϕ and the transition (s, a, s') that occurred in the physical system, so that transitions between hyperstates also exhibit the Markov property. Since we want the agent to maximize the rewards it obtains in the physical system, the reward function R' should return the same reward as in the physical system, as defined in R. Thus we define $R'(s, \phi, a) = R(s, a)$. The only remaining issue is to define the transition probabilities between hyperstates. The new transition function T' must specify the transition probabilities $T'(s, \phi, a, s', \phi') = \Pr(s', \phi' | s, a, \phi)$. By the chain rule, $\Pr(s', \phi' | s, a, \phi) = \Pr(s' | s, a, \phi) \Pr(\phi' | s, a, s', \phi)$. Since the update of the information state ϕ to ϕ' is deterministic, then $\Pr(\phi' | s, a, s', \phi)$ is either 0 or 1, depending on whether $\phi' = \mathcal{U}(\phi, s, a, s')$ or not. Hence $\Pr(\phi' | s, a, s', \phi) = I_{\{\phi'\}}(\mathcal{U}(\phi, s, a, s'))$. By the law of total probability, $\Pr(s' | s, a, \phi) = \int \Pr(s' | s, a, T, \phi) f(T | \phi) dT = \int T^{sas'} f(T | \phi) dT$, where the integral is carried over transition function T, and $f(T|\phi)$ is the probability density of transition function T under the posterior defined by ϕ. The term $\int T^{sas'} f(T|\phi) dT$ is the expectation of $T^{sas'}$ for the Dirichlet posterior defined by the parameters $\phi^a_{s,s_1}, \ldots, \phi^a_{s,s_{|S|}}$, which corresponds to $\frac{\phi^a_{s,s'}}{\sum_{s'' \in S} \phi^a_{s,s''}}$. Thus it follows that:

$$T'(s, \phi, a, s', \phi') = \frac{\phi^a_{s,s'}}{\sum_{s'' \in S} \phi^a_{s,s''}} I_{\{\phi'\}}(\mathcal{U}(\phi, s, a, s')).$$

We now have a new MDP with a known model. By solving this MDP, we can find the optimal action-selection strategy, given *a posteriori* knowledge of the environment. This new MDP has been called the Bayes-Adaptive MDP (Duff, 2002) or the HyperMDP (Castro and Precup, 2007).

Notice that while we have assumed that the reward function R is known, this BRL framework can easily be extended to the case where R is unknown. In such a case, one can proceed similarly by using a Bayesian learning method to learn the reward function R and add the posterior parameters for R in the hyperstate. The new reward function R' then becomes the expected reward under the current posterior over R, and the transition function T' would also model how to update the posterior over R, upon observation of any reward r. For brevity of presentation, it is assumed that the reward function is known throughout this paper. However, the frameworks we present in the following sections can also be extended to handle cases where the rewards are unknown, by following a similar reasoning.

2.3.2 Optimality and Value Function

The Bayes-Adaptive MDP (BAMDP) is just a conventional MDP with a countably infinite number of states. Fortunately, many theoretical results derived for standard MDPs carry

over to the Bayes-Adaptive MDP model (Duff, 2002). Hence, we know there exists an optimal deterministic policy $\pi^* : S' \rightarrow A$, and that its value function is defined by:

$$\begin{aligned} V^*(s,\phi) &= \max_{a\in A}\left[R'(s,\phi,a) + \gamma \sum_{(s',\phi')\in S'} T'(s,\phi,a,s',\phi')V^*(s',\phi')\right] \\ &= \max_{a\in A}\left[R(s,a) + \gamma \sum_{s'\in S} \frac{\phi^a_{s,s'}}{\sum_{s''\in S}\phi^a_{s,s''}} V^*(s', \mathcal{U}(\phi,s,a,s'))\right]. \end{aligned} \quad (1)$$

This value function is defined over an infinite number of hyperstates, therefore, in practice, computing V^* exactly for all hyperstates is unfeasible. However, since the summation over S is finite, we observe that from one given hyperstate, the agent can transit only to a finite number of hyperstates in one step. It follows that for any finite planning horizon t, one can compute exactly the optimal value function for a particular starting hyperstate. However the number of reachable hyperstates grows exponentially with the planning horizon.

2.3.3 Planning Algorithms

We now review existing approximate algorithms for estimating the value function in the BAMDP. Dearden et al. (1999) proposed one of the first Bayesian model-based exploration methods for RL. Instead of solving the BAMDP directly via Equation 1, the Dirichlet distributions are used to compute a distribution over the state-action values $Q^*(s,a)$, in order to select the action that has the highest expected return and value of information (Dearden et al., 1998). The distribution over Q-values is estimated by sampling MDPs from the posterior Dirichlet distribution, and then solving each sampled MDP to obtain different sampled Q-values. Re-sampling and importance sampling techniques are proposed to update the estimated Q-value distribution as the Dirichlet posteriors are updated.

Rather than using a maximum likelihood estimate for the underlying process, Strens (2000) proposes to fully represent the posterior distribution over process parameters. He then uses a greedy behavior with respect to a sample from this posterior. By doing so, he retains each hypothesis over a period of time, ensuring goal-directed exploratory behavior without the need to use approximate measures or heuristic exploration as other approaches did. The number of steps for which each hypothesis is retained limits the length of exploration sequences. The results of this method is then an automatic way of obtaining behavior which moves gradually from exploration to exploitation, without using heuristics.

Duff (2001) suggests using Finite-State Controllers (FSC) to represent compactly the optimal policy π^* of the BAMDP and then finding the best FSC in the space of FSCs of some bounded size. A gradient descent algorithm is presented to optimize the FSC and a Monte-Carlo gradient estimation is proposed to make it more tractable. This approach presupposes the existence of a good FSC representation for the policy.

For their part, Wang et al. (2005) present an online planning algorithm that estimates the optimal value function of the BAMDP (Equation 1) using Monte-Carlo sampling. This algorithm is essentially an adaptation of the Sparse Sampling algorithm (Kearns et al., 1999) to BAMDPs. However instead of growing the tree evenly by looking at all actions at each level of the tree, the tree is grown stochastically. Actions are sampled according to their likelihood of being optimal, according to their Q-value distributions (as defined by

the Dirichlet posteriors); next states are sampled according to the Dirichlet posterior on the model. This approach requires multiple sampling and solving of MDPs from the Dirichlet distributions to find which action has highest Q-value at each state node in the tree. This can be very time consuming, and so far the approach has only been applied to small MDPs.

Castro and Precup (2007) present a similar approach to Wang et al. However their approach differs on two main points. First, instead of maintaining only the posterior over models, they also maintain Q-value estimates using a standard Q-Learning method. Planning is done by growing a stochastic tree as in Wang et al. (but sampling actions uniformly instead) and solving for the value estimates in that tree using Linear Programming (LP), instead of dynamic programming. In this case, the stochastic tree represents sampled constraints, which the value estimates in the tree must satisfy. The Q-value estimates maintained by Q-Learning are used as value estimates for the fringe nodes (thus as value constraints on the fringe nodes in the LP).

Finally, Poupart et al. (2006) proposed an approximate offline algorithm to solve the BAMDP. Their algorithm, called Beetle, is an extension of the Perseus algorithm (Spaan and Vlassis, 2005) to the BAMDP model. Essentially, at the beginning, hyperstates (s, ϕ) are sampled from random interactions with the BAMDP model. An equivalent continuous POMDP (over the space of states and transition functions) is solved instead of the BAMDP (assuming (s, ϕ) is a belief state in that POMDP). The value function is represented by a set of *α-functions* over the continuous space of transition functions. Each α-function is constructed as a linear combination of basis functions; the sampled hyperstates can serve as the set of basis functions. Dynamic programming is used to incrementally construct the set of α-functions. At each iteration, updates are only performed at the sampled hyperstates, similarly to Perseus (Spaan and Vlassis, 2005) and other Point-Based POMDP algorithms (Pineau et al., 2003).

3. Bayes-Adaptive POMDPs

Despite the sustained interest in model-based BRL, the deployment to real-world applications is limited both by scalability and representation issues. In terms of representation, an important challenge for many practical problems is in handling cases where the state of the system is only partially observable. Our goal here is to show that the model-based BRL framework can be extended to handle partially observable domains. Section 3.1 provides a brief overview of the Partially Observable Markov Decision Process framework. In order to apply Bayesian RL methods in this context, we draw inspiration from the Bayes-Adaptive MDP framework presented in Section 2.3, and propose an extension of this model, called the Bayes-Adaptive POMDP (BAPOMDP). One of the main challenges that arises when considering such an extension is how to update the Dirichlet count parameters when the state is a hidden variable. As will be explained in Section 3.2, this can be achieved by including the Dirichlet parameters in the state space, and maintaining a belief state over these parameters. The BAPOMDP model thus allows an agent to improve its knowledge of an unknown POMDP domain through interaction with the environment, but also allows the decision-making aspect to be contingent on uncertainty over the model parameters. As a result, it is possible to define an action-selection strategy which can directly trade-off between (1) learning the model of the POMDP, (2) identifying the unknown state, and

(3) gathering rewards, such as to maximize its future expected return. This model offers an alternative framework for reinforcement learning in POMDPs, compared to previous history-based approaches (McCallum, 1996; Littman et al., 2002).

3.1 Background on POMDPs

While an MDP is able to capture uncertainty on future outcomes, and the BAMDP is able to capture uncertainty over the model parameters, both fail to capture uncertainty that can exist on the current state of the system at a given time step. For example, consider a medical diagnosis problem where the doctor must prescribe the best treatment to an ill patient. In this problem the state (illness) of the patient is unknown, and only its symptoms can be observed. Given the observed symptoms the doctor may believe that some illnesses are more likely, however he may still have some uncertainty about the exact illness of the patient. The doctor must take this uncertainty into account when deciding which treatment is best for the patient. When the uncertainty is high, the best action may be to order additional medical tests in order to get a better diagnosis of the patient's illness.

To address such problems, the Partially Observable Markov Decision Process (POMDP) was proposed as a generalization of the standard MDP model. POMDPs are able to model and reason about the uncertainty on the current state of the system in sequential decision problems (Sondik, 1971).

A POMDP is defined by a finite set of states S, a finite set of actions A, as well as a finite set of observations Z. These observations capture the aspects of the state which can be perceived by the agent. The POMDP is also defined by transition probabilities $\{T^{sas'}\}_{s,s'\in S,a\in A}$, where $T^{sas'} = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$, as well as observation probabilities $\{O^{saz}\}_{s\in S,a\in A,z\in Z}$ where $O^{saz} = \Pr(z_t = z | s_t = s, a_{t-1} = a)$. The reward function, $R : S \times A \to \mathbb{R}$, and discount factor, γ, are as in the MDP model.

Since the state is not directly observed, the agent must rely on the observation and action at each time step to maintain a belief state $b \in \Delta S$, where ΔS is the space of probability distributions over S. The belief state specifies the probability of being in each state given the history of action and observation experienced so far, starting from an initial belief b_0. It can be updated at each time step using the following Bayes rule:

$$b_{t+1}(s') = \frac{O^{s'a_t z_{t+1}} \sum_{s\in S} T^{sa_t s'} b_t(s)}{\sum_{s''\in s} O^{s''a_t z_{t+1}} \sum_{s\in S} T^{sa_t s''} b_t(s)}.$$

A policy $\pi : \Delta S \to A$ indicates how the agent should select actions as a function of the current belief. Solving a POMDP involves finding the optimal policy π^* that maximizes the expected discounted return over the infinite horizon. The return obtained by following π^* from a belief b is defined by Bellman's equation:

$$V^*(b) = \max_{a\in A} \left[\sum_{s\in S} b(s) R(s,a) + \gamma \sum_{z\in Z} \Pr(z|b,a) V^*(\tau(b,a,z)) \right],$$

where $\tau(b,a,z)$ is the new belief after performing action a and observation z,and $\gamma \in [0,1)$ is the discount factor.

A key result by Smallwood and Sondik (1973) shows that the optimal value function for a finite-horizon POMDP is piecewise-linear and convex. It means that the value function

V_t at any finite horizon t can be represented by a finite set of $|S|$-dimensional hyperplanes: $\Gamma_t = \{\alpha_0, \alpha_1, \ldots, \alpha_m\}$. These hyperplanes are often called α-vectors. Each defines a linear value function over the belief state space, associated with some action, $a \in A$. The value of a belief state is the maximum value returned by one of the α-vectors for this belief state:

$$V_t(b) = \max_{\alpha \in \Gamma_t} \sum_{s \in S} \alpha(s) b(s).$$

The best action is the one associated with the α-vector that returns the best value.

The Enumeration algorithm by Sondik (1971) shows how the finite set of α-vectors, Γ_t, can be built incrementally via dynamic programming. The idea is that any t-step contingency plan can be expressed by an immediate action and a mapping associating a (t-1)-step contingency plan to every observation the agent could get after this immediate action. The value of the 1-step plans corresponds directly to the immediate rewards:

$$\begin{aligned} \Gamma_1^a &= \{\alpha^a | \alpha^a(s) = R(s,a)\}, \\ \Gamma_1 &= \textstyle\bigcup_{a \in A} \Gamma_1^a. \end{aligned}$$

Then to build the α-vectors at time t, we consider all possible immediate actions the agent could take, every observation that could follow, and every combination of (t-1)-step plans to pursue subsequently:

$$\begin{aligned} \Gamma_t^{a,z} &= \{\alpha^{a,z} | \alpha^{a,z}(s) = \gamma \textstyle\sum_{s' \in S} T^{sas'} O^{s'az} \alpha'(s'), \alpha' \in \Gamma_{t-1}\}, \\ \Gamma_t^a &= \Gamma_1^a \oplus \Gamma_t^{a,z_1} \oplus \Gamma_t^{a,z_2} \oplus \ldots \oplus \Gamma_t^{a,z_{|Z|}}, \\ \Gamma_t &= \textstyle\bigcup_{a \in A} \Gamma_t^a, \end{aligned}$$

where $\oplus$ is the cross-sum operator.[7]

Exactly solving the POMDP is usually intractable, except on small domains with only a few states, actions and observations (Kaelbling et al., 1998). Various approximate algorithms, both offline (Pineau et al., 2003; Spaan and Vlassis, 2005; Smith and Simmons, 2004) and online (Paquet et al., 2005; Ross et al., 2008c), have been proposed to tackle increasingly large domains. However, all these methods require full knowledge of the POMDP model, which is a strong assumption in practice. Some approaches do not require knowledge of the model, as in Baxter and Bartlett (2001), but these approaches generally require some knowledge of a good (and preferably compact) policy class, as well as needing substantial amounts of data.

3.2 Bayesian Learning of a POMDP model

Before we introduce the full BAPOMDP model for sequential decision-making under model uncertainty in a POMDP, we first show how a POMDP model can be learned via a Bayesian approach.

Consider an agent in a POMDP $(S, A, Z, T, O, R, \gamma)$, where the transition function T and observation function O are the only unknown components of the POMDP model. Let $\bar{z}_t = (z_1, z_2, \ldots, z_t)$ be the history of observations of the agent up to time t. Recall also that we denote $\bar{s}_t = (s_0, s_1, \ldots, s_t)$ and $\bar{a}_{t-1} = (a_0, a_1, \ldots, a_{t-1})$ the history of visited states and

7. Let A and B be sets of vectors, then $A \oplus B = \{a + b | a \in A, b \in B\}$.

actions respectively. The Bayesian approach to learning T and O involves starting with a prior distribution over T and O, and maintaining the posterior distribution over T and O after observing the history $(\bar{a}_{t-1}, \bar{z}_t)$. Since the current state s_t of the agent at time t is unknown in the POMDP, we consider a joint posterior $g(s_t, T, O|\bar{a}_{t-1}, \bar{z}_t)$ over s_t, T, and O. By the laws of probability and Markovian assumption of the POMDP, we have:

$$\begin{aligned} g(s_t, T, O|\bar{a}_{t-1}, \bar{z}_t) &\propto \Pr(\bar{z}_t, s_t|T, O, \bar{a}_{t-1}) g(T, O, \bar{a}_{t-1}) \\ &\propto \textstyle\sum_{\bar{s}_{t-1} \in S^t} \Pr(\bar{z}_t, \bar{s}_t|T, O, \bar{a}_{t-1}) g(T, O) \\ &\propto \textstyle\sum_{\bar{s}_{t-1} \in S^t} g(s_0, T, O) \prod_{i=1}^{t} T^{s_{i-1} a_{i-1} s_i} O^{s_i a_{i-1} z_i} \\ &\propto \textstyle\sum_{\bar{s}_{t-1} \in S^t} g(s_0, T, O) \left[\prod_{s,a,s'} (T^{sas'})^{N^a_{ss'}(\bar{s}_t, \bar{a}_{t-1})} \right] \times \\ &\quad \left[\prod_{s,a,z} (O^{saz})^{N^a_{sz}(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t)} \right], \end{aligned}$$

where $g(s_0, T, O)$ is the joint prior over the initial state s_0, transition function T, and observation function O; $N^a_{ss'}(\bar{s}_t, \bar{a}_{t-1}) = \sum_{i=0}^{t-1} I_{\{(s,a,s')\}}(s_i, a_i, s_{i+1})$ is the number of times the transition (s, a, s') appears in the history of state-action $(\bar{s}_t, \bar{a}_{t-1})$; and $N^a_{sz}(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t) = \sum_{i=1}^{t} I_{\{(s,a,z)\}}(s_i, a_{i-1}, z_i)$ is the number of times the observation (s, a, z) appears in the history of state-action-observations $(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t)$. We use proportionality rather than equality in the expressions above because we have not included the normalization constant.

Under the assumption that the prior $g(s_0, T, O)$ is defined by a product of independent priors of the form:

$$g(s_0, T, O) = g(s_0) \prod_{s,a} g_{sa}(T^{sa\cdot}) g_{sa}(O^{sa\cdot}),$$

and that $g_{sa}(T^{sa\cdot})$ and $g_{sa}(O^{sa\cdot})$ are Dirichlet priors defined $\forall s, a$, then we observe that the posterior is a mixture of joint Dirichlets, where each joint Dirichlet component is parameterized by the counts corresponding to one specific possible state sequence:

$$\begin{aligned} g(s_t, T, O|\bar{a}_{t-1}, \bar{z}_t) \propto & \textstyle\sum_{\bar{s}_{t-1} \in S^t} g(s_0) c(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t) \times \\ & \left[\prod_{s,a,s'} (T^{sas'})^{N^a_{ss'}(\bar{s}_t, \bar{a}_{t-1}) + \phi^a_{ss'} - 1} \right] \times \\ & \left[\prod_{s,a,z} (O^{saz})^{N^a_{sz}(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t) + \psi^a_{sz} - 1} \right]. \end{aligned} \tag{2}$$

Here, $\phi^a_{s\cdot}$ are the prior Dirichlet count parameters for $g_{sa}(T^{sa\cdot})$, $\psi^a_{s\cdot}$ are the prior Dirichlet count parameters for $g_{sa}(O^{sa\cdot})$, and $c(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t)$ is a constant which corresponds to the normalization constant of the joint Dirichlet component for the state-action-observation history $(\bar{s}_t, \bar{a}_{t-1}, \bar{z}_t)$. Intuitively, Bayes' rule tells us that given a particular state sequence, it is possible to compute the proper posterior counts of the Dirichlets, but since the state sequence that actually occurred is unknown, all state sequences (and their corresponding Dirichlet posteriors) must be considered, with some weight proportional to the likelihood of each state sequence.

In order to update the posterior online, each time the agent performs an action and gets an observation, it is more useful to express the posterior in recursive form:

$$g(s_t, T, O|\bar{a}_{t-1}, \bar{z}_t) \propto \sum_{s_{t-1} \in S} T^{s_{t-1} a_{t-1} s_t} O^{s_t a_{t-1} z_t} g(s_{t-1}, T, O|\bar{a}_{t-2}, \bar{z}_{t-1}).$$

Hence if $g(s_{t-1}, T, O|\bar{a}_{t-2}, \bar{z}_{t-1}) = \sum_{(\phi,\psi) \in \mathcal{C}(s_{t-1})} w(s_{t-1}, \phi, \psi) f(T, O|\phi, \psi)$ is a mixture of $|\mathcal{C}(s_{t-1})|$ joint Dirichlet components, where each component (ϕ, ψ) is parameterized by

the set of transition counts $\phi = \{\phi^a_{ss'} \in \mathbb{N} | s, s' \in S, a \in A\}$ and the set observation counts $\psi = \{\psi^a_{sz} \in \mathbb{N} | s \in S, a \in A, z \in Z\}$, then $g(s_t, T, O | \bar{a}_{t-1}, \bar{z}_t)$ is a mixture of $\prod_{s \in S} |\mathcal{C}(s)|$ joint Dirichlet components, given by:

$$\begin{aligned} g(s_t, T, O | \bar{a}_{t-1}, \bar{z}_t) \quad \propto \quad & \sum_{s_{t-1} \in S} \sum_{(\phi,\psi) \in \mathcal{C}(s_{t-1})} w(s_{t-1}, \phi, \psi) c(s_{t-1}, a_{t-1}, s_t, z_{t-1}, \phi, \psi) \\ & f(T, O | \mathcal{U}(\phi, s_{t-1}, a_{t-1}, s_t), \mathcal{U}(\psi, s_t, a_{t-1}, z_t)), \end{aligned}$$

where $\mathcal{U}(\phi, s, a, s')$ increments the count $\phi^a_{ss'}$ by one in the set of counts ϕ, $\mathcal{U}(\psi, s, a, z)$ increments the count ψ^a_{sz} by one in the set of counts ψ, and $c(s_{t-1}, a_{t-1}, s_t, z_{t-1}, \phi, \psi)$ is a constant corresponding to the ratio of the normalization constants of the joint Dirichlet component (ϕ, ψ) before and after the update with $(s_{t-1}, a_{t-1}, s_t, z_{t-1})$. This last equation gives us an online algorithm to maintain the posterior over (s, T, O), and thus allows the agent to learn about the unknown T and O via Bayesian inference.

Now that we have a simple method of maintaining the uncertainty over both the state and model parameters, we would like to address the more interesting question of how to optimally behave in the environment under such uncertainty, in order to maximize future expected return. Here we proceed similarly to the Bayes-Adaptive MDP framework defined in Section 2.3.

First, notice that the posterior $g(s_t, T, O | \bar{a}_{t-1}, \bar{z}_t)$ can be seen as a probability distribution (belief) b over tuples (s, ϕ, ψ), where each tuple represents a particular joint Dirichlet component parameterized by the counts (ϕ, ψ) for a state sequence ending in state s (i.e., the current state is s), and the probabilities in the belief b correspond to the mixture weights. Now we would like to find a policy π for the agent which maps such beliefs over (s, ϕ, ψ) to actions $a \in A$. This suggests that the sequential decision problem of optimally behaving under state and model uncertainty can be modeled as a POMDP over hyperstates of the form (s, ϕ, ψ).

Consider a new POMDP $(S', A, Z, P', R', \gamma)$, where the set of states (hyperstates) is formally defined as $S' = S \times \mathcal{T} \times \mathcal{O}$, with $\mathcal{T} = \{\phi \in \mathbb{N}^{|S|^2|A|} | \forall (s, a) \in S \times A, \ \sum_{s' \in S} \phi^a_{ss'} > 0\}$ and $\mathcal{O} = \{\psi \in \mathbb{N}^{|S||A||Z|} | \forall (s, a) \in S \times A, \ \sum_{z \in Z} \psi^a_{sz} > 0\}$. As in the definition of the BAMDP, the constraints on the count parameters ϕ and ψ are only to ensure that the transition-observation probabilities, as defined below, are well defined. The action and observation sets are the same as in the original POMDP. The rewards depend only on the state $s \in S$ and action $a \in A$ (but not the counts ϕ and ψ), thus we have $R'(s, \phi, \psi, a) = R(s, a)$. The transition and observations probabilities in the BAPOMDP are defined by a joint transition-observation function $P' : S' \times A \times S' \times Z \to [0, 1]$, such that $P'(s, \phi, \psi, a, s', \phi', \psi', z) = \Pr(s', \phi', \psi', z | s, \phi, \psi, a)$. This joint probability can be factorized by using the laws of probability and standard independence assumptions:

$$\begin{aligned} & \Pr(s', \phi', \psi', z | s, \phi, \psi, a) \\ = \ & \Pr(s' | s, \phi, \psi, a) \Pr(z | s, \phi, \psi, a, s') \Pr(\phi' | s, \phi, \psi, a, s', z) \Pr(\psi' | s, \phi, \psi, a, s', \phi', z) \\ = \ & \Pr(s' | s, a, \phi) \Pr(z | a, s', \psi) \Pr(\phi' | \phi, s, a, s') \Pr(\psi' | \psi, a, s', z). \end{aligned}$$

As in the Bayes-Adaptive MDP case, $\Pr(s' | s, a, \phi)$ corresponds to the expectation of $\Pr(s' | s, a)$ under the joint Dirichlet posterior defined by ϕ, and $\Pr(\phi' | \phi, s, a, s')$ is either 0 or 1, depending on whether ϕ' corresponds to the posterior after observing transition (s, a, s')

from prior ϕ. Hence $\Pr(s'|s,a,\phi) = \frac{\phi^a_{ss'}}{\sum_{s''\in S}\phi^a_{ss''}}$, and $\Pr(\phi'|\phi,s,a,s') = I_{\{\phi'\}}(\mathcal{U}(\phi,s,a,s'))$. Similarly, $\Pr(z|a,s',\psi) = \int O^{s'az} f(O|\psi)dO$, which is the expectation of the Dirichlet posterior for $\Pr(z|s',a)$, and $\Pr(\psi'|\psi,a,s',z)$, is either 0 or 1, depending on whether ψ' corresponds to the posterior after observing observation (s',a,z) from prior ψ. Thus $\Pr(z|a,s',\psi) = \frac{\psi^a_{s'z}}{\sum_{z'\in Z}\psi^a_{s'z'}}$, and $\Pr(\psi'|\psi,a,s',z) = I_{\{\psi'\}}(\mathcal{U}(\psi,s',a,z))$. To simplify notation, we denote $T^{sas'}_{\phi} = \frac{\phi^a_{ss'}}{\sum_{s''\in S}\phi^a_{ss''}}$ and $O^{s'az}_{\psi} = \frac{\psi^a_{s'z}}{\sum_{z'\in Z}\psi^a_{s'z'}}$. It follows that the joint transition-observation probabilities in the BAPOMDP are defined by:

$$\Pr(s',\phi',\psi',z|s,\phi,\psi,a) = T^{sas'}_{\phi} O^{s'az}_{\psi} I_{\{\phi'\}}(\mathcal{U}(\phi,s,a,s')) I_{\{\psi'\}}(\mathcal{U}(\psi,s',a,z)).$$

Hence, the BAPOMDP defined by the POMDP (S',A,Z,P',R',γ) has a known model and characterizes the problem of optimal sequential decision-making in the original POMDP (S,A,Z,T,O,R,γ) with uncertainty on the transition T and observation functions O described by Dirichlet distributions.

An alternative interpretation of the BAPOMDP is as follows: given the unknown state sequence that occurred since the beginning, one can compute exactly the posterior counts ϕ and ψ. Thus there exists a unique (ϕ,ψ) reflecting the correct posterior counts according to the state sequence that occurred, but these correct posterior counts are only partially observable through the observations $z \in Z$ obtained by the agent. Thus (ϕ,ψ) can simply be thought of as other hidden state variables that the agent tracks via the belief state, based on its observations. The BAPOMDP formulates the decision problem of optimal sequential decision-making under partial observability of both the state $s \in S$, and posterior counts (ϕ,ψ).

The belief state in the BAPOMDP corresponds exactly to the posterior defined in the previous section (Equation 2). By maintaining this belief, the agent maintains its uncertainty on the POMDP model and learns about the unknown transition and observations functions. Initially, if ϕ_0 and ψ_0 represent the prior Dirichlet count parameters (i.e., the agent's prior knowledge of T and O), and b_0 the initial state distribution of the unknown POMDP, then the initial belief b'_0 of the BAPOMDP is defined as $b'_0(s,\phi,\psi) = b_0(s) I_{\{\phi_0\}}(\phi) I_{\{\psi_0\}}(\psi)$. Since the BAPOMDP is just a POMDP with an infinite number of states, the belief update and value function equations presented in Section 3.1 can be applied directly to the BAPOMDP model. However, since there is an infinite number of hyperstates, these calculations can be performed exactly in practice only if the number of possible hyperstates in the belief is finite. The following theorem shows that this is the case at any finite time t:

Theorem 1 *Let (S',A,Z,P',R',γ) be a BAPOMDP constructed from the POMDP (S,A,Z,T,O,R,γ). If S is finite, then at any time t, the set $S'_{b'_t} = \{\sigma \in S' | b'_t(\sigma) > 0\}$ has size $|S'_{b'_t}| \leq |S|^{t+1}$.*

Proof Proof available in Appendix A. ■

function $\tau(b, a, z)$
Initialize b' as a 0 vector.
for all $(s, \phi, \psi) \in S'_b$ **do**
  for all $s' \in S$ **do**
    $\phi' \leftarrow \mathcal{U}(\phi, s, a, s')$
    $\psi' \leftarrow \mathcal{U}(\psi, s', a, z)$
    $b'(s', \phi', \psi') \leftarrow b'(s', \phi', \psi') + b(s, \phi, \psi) T_\phi^{sas'} O_\psi^{s'az}$
  end for
end for
return normalized b'

Algorithm 1: Exact Belief Update in BAPOMDP.

The proof of Theorem 1 suggests that it is sufficient to iterate over S and $S'_{b'_{t-1}}$ in order to compute the belief state b'_t when an action and observation are taken in the environment. Hence, we can update the belief state in closed-form, as outlined in Algorithm 1. Of course this algorithm is not tractable for large domains with long action-observation sequences. Section 5 provides a number of approximate tracking algorithms which tackle this problem.

3.3 Exact Solution for the BAPOMDP in Finite Horizons

The value function of a BAPOMDP for finite horizons can be represented by a finite set Γ of functions $\alpha : S' \rightarrow \mathbb{R}$, as in standard POMDPs. This is shown formally in the following theorem:

Theorem 2 *For any horizon t, there exists a finite set Γ_t of functions $S' \rightarrow \mathbb{R}$, such that $V_t^*(b) = \max_{\alpha \in \Gamma_t} \sum_{\sigma \in S'} \alpha(\sigma) b(\sigma)$.*

Proof Proof available in the appendix. ■

The proof of this theorem shows that as in any POMDP, an exact solution of the BAPOMDP can be computed using dynamic programming, by incrementally constructing the set of α-functions that defines the value function as follows:

$$
\begin{aligned}
\Gamma_1^a &= \{\alpha^a | \alpha^a(s, \phi, \psi) = R(s, a)\}, \\
\Gamma_t^{a,z} &= \{\alpha^{a,z} | \alpha^{a,z}(s, \phi, \psi) = \gamma \textstyle\sum_{s' \in S} T_\phi^{sas'} O_\psi^{s'az} \alpha'(s', \mathcal{U}(\phi, s, a, s'), \mathcal{U}(\psi, s', a, z)), \\
&\qquad\qquad \alpha' \in \Gamma_{t-1}\}, \\
\Gamma_t^a &= \Gamma_1^a \oplus \Gamma_t^{a,z_1} \oplus \Gamma_t^{a,z_2} \oplus \cdots \oplus \Gamma_t^{a,z_{|Z|}}, \quad \text{(where } \oplus \text{ is the cross sum operator)}, \\
\Gamma_t &= \textstyle\bigcup_{a \in A} \Gamma_t^a.
\end{aligned}
$$

However in practice, it will be impossible to compute $\alpha_i^{a,z}(s, \phi, \psi)$ for all $(s, \phi, \psi) \in S'$, unless a particular finite parametric form for the α-functions is used. Poupart and Vlassis (2008) showed that these α-functions can be represented as a linear combination of product of Dirichlets and can thus be represented by a finite number of parameters. Further discussion of their work is included in Section 7. We present an alternate approach in Section 5.

4. Approximating the BAPOMDP by a Finite POMDP

Solving the BAPOMDP exactly for all belief states is often impossible due to the dimensionality of the state space, in particular because the count vectors can grow unbounded. The first proposed method to address this problem is to reduce this infinite state space to a finite state space, while preserving the value function of the BAPOMDP to arbitrary precision. This allows us to compute an ϵ-optimal value function over the resulting finite-dimensional belief space using standard finite POMDP solvers. This can then be used to obtain an ϵ-optimal policy to the BAPOMDP.

The main intuition behind the compression of the state space presented here is that, as the Dirichlet counts grow larger and larger, the transition and observation probabilities defined by these counts do not change much when the counts are incremented by one. Hence, there should exist a point where if we simply stop incrementing the counts, the value function of that approximate BAPOMDP (where the counts are bounded) approximates the value function of the BAPOMDP within some $\epsilon > 0$. If we can bound above the counts in such a way, this ensures that the state space will be finite.

In order to find such a bound on the counts, we begin by deriving an upper bound on the value difference between two hyperstates that differ only by their model estimates ϕ and ψ. This bound uses the following definitions: given $\phi, \phi' \in \mathcal{T}$, and $\psi, \psi' \in \mathcal{O}$, define $D_S^{sa}(\phi, \phi') = \sum_{s' \in S} \left| T_\phi^{sas'} - T_{\phi'}^{sas'} \right|$, $D_Z^{sa}(\psi, \psi') = \sum_{z \in Z} \left| O_\psi^{saz} - O_{\psi'}^{saz} \right|$, $\mathcal{N}_\phi^{sa} = \sum_{s' \in S} \phi_{ss'}^a$, and $\mathcal{N}_\psi^{sa} = \sum_{z \in Z} \psi_{sz}^a$.

Theorem 3 *Given any $\phi, \phi' \in \mathcal{T}$, $\psi, \psi' \in \mathcal{O}$, and $\gamma \in (0,1)$, then for all t:*

$$\sup_{\alpha_t \in \Gamma_t, s \in S} |\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| \leq \frac{2\gamma ||R||_\infty}{(1-\gamma)^2} \sup_{s,s' \in S, a \in A} \Bigg[D_S^{sa}(\phi,\phi') + D_Z^{s'a}(\psi,\psi') + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s'' \in S} \left| \phi_{ss''}^a - \phi_{ss''}'^a \right|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z \in Z} \left| \psi_{s'z}^a - \psi_{s'z}'^a \right|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Bigg]$$

Proof Proof available in the appendix. ■

We now use this bound on the α-vector values to approximate the space of Dirichlet parameters within a finite subspace. We use the following definitions: given any $\epsilon > 0$, define $\epsilon' = \frac{\epsilon(1-\gamma)^2}{8\gamma ||R||_\infty}$, $\epsilon'' = \frac{\epsilon(1-\gamma)^2 \ln(\gamma^{-e})}{32\gamma ||R||_\infty}$, $N_S^\epsilon = \max\left(\frac{|S|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1 \right)$ and $N_Z^\epsilon = \max\left(\frac{|Z|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1 \right)$.

Theorem 4 *Given any $\epsilon > 0$ and $(s, \phi, \psi) \in S'$ such that $\exists a \in A, \exists s' \in S$, $\mathcal{N}_\phi^{s'a} > N_S^\epsilon$ or $\mathcal{N}_\psi^{s'a} > N_Z^\epsilon$, then $\exists (s, \phi', \psi') \in S'$ such that $\forall a \in A, \forall s' \in S$, $\mathcal{N}_{\phi'}^{s'a} \leq N_S^\epsilon$, $\mathcal{N}_{\psi'}^{s'a} \leq N_Z^\epsilon$ and $|\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| < \epsilon$ holds for all t and $\alpha_t \in \Gamma_t$.*

Proof Proof available in the appendix. ■

Theorem 4 suggests that if we want a precision of ϵ on the value function, we just need to restrict the space of Dirichlet parameters to count vectors $\phi \in \tilde{\mathcal{T}}_\epsilon = \{\phi \in \mathbb{N}^{|S|^2|A|} | \forall a \in A, s \in S, 0 < \mathcal{N}_\phi^{sa} \leq N_S^\epsilon\}$, and $\psi \in \tilde{\mathcal{O}}_\epsilon = \{\psi \in \mathbb{N}^{|S||A||Z|} | \forall a \in A, s \in S, 0 < \mathcal{N}_\psi^{sa} \leq N_Z^\epsilon\}$. Since $\tilde{\mathcal{T}}_\epsilon$ and $\tilde{\mathcal{O}}_\epsilon$ are finite, we can define a finite approximate BAPOMDP as the tuple

$(\tilde{S}_\epsilon, A, Z, \tilde{P}_\epsilon, \tilde{R}_\epsilon, \gamma)$, where $\tilde{S}_\epsilon = S \times \tilde{\mathcal{T}}_\epsilon \times \tilde{\mathcal{O}}_\epsilon$ is the finite state space, and $\tilde{P}_\epsilon$ is the joint transition-observation function over this finite state space. To define this function, we need to ensure that whenever the count vectors are incremented, they stay within the finite space. To achieve this, we define a projection operator $\mathcal{P}_\epsilon : S' \to \tilde{S}_\epsilon$ that simply projects every state in S' to their closest state in $\tilde{S}_\epsilon$.

Definition 1 *Let $d : S' \times S' \to \mathbb{R}$ be defined such that:*

$$d(s,\phi,\psi,s',\phi',\psi') = \begin{cases} \frac{2\gamma||R||_\infty}{(1-\gamma)^2} \sup_{s,s' \in S, a \in A} \Big[D_S^{sa}(\phi,\phi') + D_Z^{s'a}(\psi,\psi') \\ \quad + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s'' \in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{as}+1)(\mathcal{N}_{\phi'}^{as}+1)} + \frac{\sum_{z \in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{as'}+1)(\mathcal{N}_{\psi'}^{as'}+1)} \right) \Big], & \text{if } s = s' \\ \frac{8\gamma||R||_\infty}{(1-\gamma)^2} \left(1 + \frac{4}{\ln(\gamma^{-e})}\right) + \frac{2||R||_\infty}{(1-\gamma)}, & \text{otherwise.} \end{cases}$$

Definition 2 *Let $\mathcal{P}_\epsilon : S' \to \tilde{S}_\epsilon$ be defined as $\mathcal{P}_\epsilon(s) = \arg\min_{s' \in \tilde{S}_\epsilon} d(s,s')$.*

The function d uses the bound defined in Theorem 3 as a distance between states that only differ in their ϕ and ψ vectors, and uses an upper bound on that value when the states differ. Thus $\mathcal{P}_\epsilon$ always maps states $(s,\phi,\psi) \in S'$ to some state $(s,\phi',\psi') \in \tilde{S}_\epsilon$. Note that if $\sigma \in \tilde{S}_\epsilon$, then $\mathcal{P}_\epsilon(\sigma) = \sigma$. Using $\mathcal{P}_\epsilon$, the joint transition-observation function can then be defined as follows:

$$\tilde{P}_\epsilon(s,\phi,\psi,a,s',\phi',\psi',z) = T_\phi^{sas'} O_\psi^{s'az} I_{\{(s',\phi',\psi')\}}(\mathcal{P}_\epsilon(s', \mathcal{U}(\phi,s,a,s'), \mathcal{U}(\psi,s',a,z))).$$

This definition is the same as the one in the BAPOMDP, except that now an extra projection is added to make sure that the incremented count vectors stay in $\tilde{S}_\epsilon$. Finally, the reward function $\tilde{R}_\epsilon : \tilde{S}_\epsilon \times A \to \mathbb{R}$ is defined as $\tilde{R}_\epsilon((s,\phi,\psi),a) = R(s,a)$. This defines a proper finite POMDP $(\tilde{S}_\epsilon, A, Z, \tilde{P}_\epsilon, \tilde{R}_\epsilon, \gamma)$, which can be used to approximate the original BAPOMDP model.

Next, we are interested in characterizing the quality of solutions that can be obtained with this finite model. Theorem 5 bounds the value difference between α-vectors computed with this finite model and the α-vector computed with the original model.

Theorem 5 *Given any $\epsilon > 0$, $(s,\phi,\psi) \in S'$ and $\alpha_t \in \Gamma_t$ computed from the infinite BAPOMDP. Let $\tilde{\alpha}_t$ be the α-vector representing the same conditional plan as α_t but computed with the finite POMDP $(\tilde{S}_\epsilon, A, Z, \tilde{T}_\epsilon, \tilde{O}_\epsilon, \tilde{R}_\epsilon, \gamma)$, then $|\tilde{\alpha}_t(\mathcal{P}_\epsilon(s,\phi,\psi)) - \alpha_t(s,\phi,\psi)| < \frac{\epsilon}{1-\gamma}$.*

Proof Proof available in the appendix. To summarize, it solves a recurrence over the 1-step approximation in Theorem 4. ■

Such bounded approximation over the α-functions of the BAPOMDP implies that the optimal policy obtained from the finite POMDP approximation has an expected value close to the value of the optimal policy of the full (non-projected) BAPOMDP:

Theorem 6 *Given any $\epsilon > 0$, and any horizon t, let $\tilde{\pi}_t$ be the optimal t-step policy computed from the finite POMDP $(\tilde{S}_\epsilon, A, Z, \tilde{T}_\epsilon, \tilde{O}_\epsilon, \tilde{R}_\epsilon, \gamma)$, then for any initial belief b the value of executing policy $\tilde{\pi}_t$ in the BAPOMDP $V_{\tilde{\pi}_t}(b) \geq V^*(b) - 2\frac{\epsilon}{1-\gamma}$.*

Proof Proof available in the appendix, and follows from Theorem 5. ■

We note that the last two theorems hold even if we construct the finite POMDP with the following approximate state projection $\tilde{\mathcal{P}}_\epsilon$, which is more easy to use in practice:

Definition 3 *Let* $\tilde{\mathcal{P}}_\epsilon : S' \to \tilde{S}_\epsilon$ *be defined as* $\tilde{\mathcal{P}}_\epsilon(s, \phi, \psi) = (s, \hat{\phi}, \hat{\psi})$ *where:*

$$\hat{\phi}^a_{s',s''} = \begin{cases} \phi^a_{s',s''} & \text{if } \mathcal{N}^{s'a}_\phi \leq N^\epsilon_S \\ \lfloor N^\epsilon_S T^{s'as''}_\phi \rfloor & \text{if } \mathcal{N}^{s'a}_\phi > N^\epsilon_S \end{cases}$$

$$\hat{\psi}^a_{s',z} = \begin{cases} \psi^a_{s',z} & \text{if } \mathcal{N}^{s'a}_\psi \leq N^\epsilon_Z \\ \lfloor N^\epsilon_Z O^{s'az}_\psi \rfloor & \text{if } \mathcal{N}^{s'a}_\psi > N^\epsilon_Z \end{cases}$$

This follows from the proof of Theorem 5, which only relies on such a projection, and not on the projection that minimizes d (as done by $\mathcal{P}_\epsilon$).

Given that the state space is now finite, offline solution methods from the literature on finite POMDPs could potentially be applied to obtain an ϵ-optimal policy to the BAPOMDP. Note however that even though the state space is finite, it will generally be very large for small ϵ, such that the resulting finite POMDP may still be intractable to solve offline, even for small domains.

An alternative approach is to solve the BAPOMDP online, by focusing on finding the best immediate action to perform in the current belief of the agent, as in online POMDP solution methods (Ross et al., 2008c). In fact, provided we have an efficient way of updating the belief, online POMDP solvers can be applied directly in the infinite BAPOMDP without requiring a finite approximation of the state space. In practice, maintaining the exact belief in the BAPOMDP quickly becomes intractable (exponential in the history length, as shown in Theorem 1). The next section proposes several practical efficient approximations for both belief updating and online planning in the BAPOMDP.

5. Towards a Tractable Approach to BAPOMDPs

Having fully specified the BAPOMDP framework and its finite approximation, we now turn our attention to the problem of scalable belief tracking and planning in this framework. This section is intentionally briefer, as many of the results in the probabilistic reasoning literature can be applied to the BAPOMDP framework. We outline those methods which have proven effective in our empirical evaluations.

5.1 Approximate Belief Monitoring

As shown in Theorem 1, the number of states with non-zero probability grows exponentially in the planning horizon, thus exact belief monitoring can quickly become intractable. This problem is not unique to the Bayes-optimal POMDP framework, and was observed in the context of Bayes nets with missing data (Heckerman et al., 1995). We now discuss different particle-based approximations that allow polynomial-time belief tracking.

Monte-Carlo Filtering: Monte-Carlo filtering algorithms have been widely used for sequential state estimation (Doucet et al., 2001). Given a prior belief b, followed by action

function $WD(b, a, z, K)$
$b' \leftarrow \tau(b, a, z)$
Initialize b'' as a 0 vector.
$(s, \phi, \psi) \leftarrow \text{argmax}_{(s',\phi',\psi') \in S'_{b'}} b'(s', \phi', \psi')$
$b''(s, \phi, \psi) \leftarrow b'(s, \phi, \psi)$
for $i = 2$ to K **do**
  $(s, \phi, \psi) \leftarrow \text{argmax}_{(s',\phi',\psi') \in S'_{b'}} b'(s', \phi', \psi') \min_{(s'',\phi'',\psi'') \in S'_{b''}} d(s', \phi', \psi', s'', \phi'', \psi'')$
  $b''(s, \phi, \psi) \leftarrow b'(s, \phi, \psi)$
end for
return normalized b''

Algorithm 2: Weighted Distance Belief Update in BAPOMDP.

a and observation z, the new belief b' is obtained by first sampling K states from the distribution b, then for each sampled s a new state s' is sampled from $T^{sa\cdot}$. Finally, the probability $O^{s'az}$ is added to $b'(s')$ and the belief b' is re-normalized. This will capture at most K states with non-zero probabilities. In the context of BAPOMDPs, we use a slight variation of this method, where (s, ϕ, ψ) are first sampled from b, and then a next state $s' \in S$ is sampled from the normalized distribution $T^{sa\cdot}_{\phi} O^{\cdot az}_{\psi}$. The probability $1/K$ is added directly to $b'(s', \mathcal{U}(\phi, s, a, s'), \mathcal{U}(\psi, s, a, s'))$.

Most Probable: Another possibility is to perform the exact belief update at a given time step, but then only keep the K most probable states in the new belief b', and re-normalize b'. This minimizes the L_1 distance between the exact belief b' and the approximate belief maintained with K particles.[8] While keeping only the K most probable particles biases the belief of the agent, this can still be a good approach in practice, as minimizing the L_1 distance bounds the difference between the values of the exact and approximate belief: that is, $|V^*(b) - V^*(b')| \leq \frac{||R||_\infty}{1-\gamma} ||b - b'||_1$.

Weighted Distance Minimization: Finally, we consider an belief approximation technique which aims to directly minimize the difference in value function between the approximate and exact belief state by exploiting the upper bound on the value difference defined in Section 4. Hence, in order to keep the K particles which best approximate the exact belief's value, an exact belief update is performed and then the K particles which minimize the weighted sum of distance measures, where distance is defined as in Definition 1, are kept to approximate the exact belief. This procedure is described in Algorithm 2.

5.2 Online Planning

As discussed above, standard offline or online POMDP solvers can be used to optimize the choice of action in the BAPOMDP model. Online POMDP solvers (Paquet et al., 2005; Ross et al., 2008c) have a clear advantage over offline finite POMDP solvers (Pineau et al., 2003; Spaan and Vlassis, 2005; Smith and Simmons, 2004) in the context of the BAPOMDP as they can be applied directly in infinite POMDPs, provided we have an efficient way to compute beliefs. Hence online POMDP solvers can be applied directly to solve the BAPOMDP without using the finite POMDP representation presented in Section 4. Another advantage of the online approach is that by planning from the current belief,

8. The L_1 distance between two beliefs b and b', denoted $||b - b'||_1$, is defined as $\sum_{\sigma \in S'} |b(\sigma) - b'(\sigma)|$.

for any finite planning horizon t, one can compute exactly the optimal value function, as only a finite number of beliefs can be reached over that finite planning horizon. While the number of reachable beliefs is exponential in the horizon, often only a small subset is most relevant for obtaining a good estimate of the value function. Recent online algorithms (Ross et al., 2008c) have leveraged this by developing several heuristics for focusing computations on only the most important reachable beliefs to obtain a good estimate quickly.

Since our focus is not on developing new online planning algorithms, we hereby simply present a simple online lookahead search algorithm that performs dynamic programming over all the beliefs reachable within some fixed finite planning horizon from the current belief. The action with highest return over that finite horizon is executed and then planning is conducted again on the next belief.

To further limit the complexity of the online planning algorithm, we used the approximate belief monitoring methods detailed above. The detailed procedure is provided in Algorithm 3. This algorithm takes as input: b is the current belief of the agent, D the desired depth of the search, and K the number of particles to use to compute the next belief states. At the end of this procedure, the agent executes action $bestA$ in the environment and restarts this procedure with its next belief. Note here that an approximate value function $\hat{V}$ can be used to approximate the long term return obtained by the optimal policy from the fringe beliefs. For efficiency reasons, we simply defined $\hat{V}(b)$ to be the maximum immediate reward in belief b throughout our experiments. The overall complexity of this planning approach is $O((|A||Z|)^D C_b)$, where C_b is the complexity of updating the belief.

function $\mathrm{V}(b, d, K)$
if $d = 0$ **then**
  return $\hat{V}(b)$
end if
$maxQ \leftarrow -\infty$
for all $a \in A$ **do**
  $q \leftarrow \sum_{(s,\phi,\psi)\in S'_b} b(s,\phi,\psi)R(s,a)$
  for all $z \in Z$ **do**
    $b' \leftarrow \hat{\tau}(b, a, z, K)$
    $q \leftarrow q + \gamma \Pr(z|b,a)\mathrm{V}(b', d-1, K)$
  end for
  if $q > maxQ$ **then**
    $maxQ \leftarrow q$
    $maxA \leftarrow a$
  end if
end for
if $d = D$ **then**
  $bestA \leftarrow maxA$
end if
return $maxQ$

Algorithm 3: Online Planning in the BAPOMDP.

In general, planning via forward search can be improved by using an accurate simulator, a good exploration policy, and a good heuristic function. For example, any offline POMDP solution can be used at the leaves of the lookahead search to improve search quality (Ross

et al., 2008c). Additionally, more efficient online planning algorithms presented in Ross et al. (2008c) could be used provided one can compute an informative upper bound and lower bound on the value function of the BAPOMDP.

6. Empirical Evaluation

The main focus of this paper is on the definition of the Bayes-Adaptive POMDP model, and examination of its theoretical properties. Nonetheless it is useful to consider experiments on a few sample domains to verify that the algorithms outlined in Section 5 produce reasonable results. We begin by comparing the three different belief approximations introduced above. To do so, we use a simple online d-step lookahead search, and compare the overall expected return and model accuracy in three different problems: the well-known Tiger domain (Kaelbling et al., 1998), a new domain called Follow which simulates simple human-robot interactions and finally a standard robot planning domain known as RockSample (Smith and Simmons, 2004).

Given $T^{sas'}$ and $O^{s'az}$ the exact probabilities of the (unknown) POMDP, the model accuracy is measured in terms of the weighted sum of L1-distance, denoted $WL1$, between the exact model and the probable models in a belief state b:

$$\begin{aligned} WL1(b) &= \sum_{(s,\phi,\psi)\in S'_b} b(s,\phi,\psi) L1(\phi,\psi) \\ L1(\phi,\psi) &= \sum_{a\in A}\sum_{s'\in S}\left[\sum_{s\in S}|T^{sas'}_{\phi} - T^{sas'}| + \sum_{z\in Z}|O^{s'az}_{\psi} - O^{s'az}|\right] \end{aligned}$$

6.1 Tiger

The Tiger problem (Kaelbling et al., 1998) is a 2-state POMDP, $S = \{tiger\_left, tiger\_right\}$, describing the position of the tiger. The tiger is assumed to be behind a door; its location is inferred through a noisy observation, $Z = \{hear\_right, hear\_left\}$. The agent has to select whether to open a door (preferably such as to avoid the tiger), or listen for further information,
$A = \{open\_left, open\_right, listen\}$. We consider the case where the transition and reward parameters are known, but the observation probabilities are not. Hence, there are four unknown parameters: O_{Ll}, O_{Lr}, O_{Rl}, O_{Rr} (O_{Lr} stands for $\Pr(z = hear\_right | s = tiger\_left, a = listen)$). We define the observation count vector, $\psi = (\psi_{Ll}, \psi_{Lr}, \psi_{Rl}, \psi_{Rr})$, and consider a prior of $\psi_0 = (5, 3, 3, 5)$, which specifies an expected sensor accuracy of 62.5% (instead of the correct 85%) in both states. Each simulation consists of 100 episodes. Episodes terminate when the agent opens a door, at which point the POMDP state (i.e., tiger's position) is reset, but the distribution over count vectors is carried over to the next episode.

Figure 1 shows how the average return and model accuracy evolve over the 100 episodes (results are averaged over 1000 simulations), using an online 3-step lookahead search with varying belief approximations and parameters. Returns obtained by planning directly with the prior and exact model (without learning) are shown for comparison. Model accuracy is measured on the initial belief of each episode. Figure 1 also compares the average planning time per action taken by each approach. We observe from these figures that the results for the Most Probable and Weighted Distance approximations are similar and perform well even with few particles. On the other hand, the performance of the Monte-Carlo belief tracking

is much weaker, even using many more particles (64). The Most Probable approach yields slightly more efficient planning times than the Weighted Distance approximation.

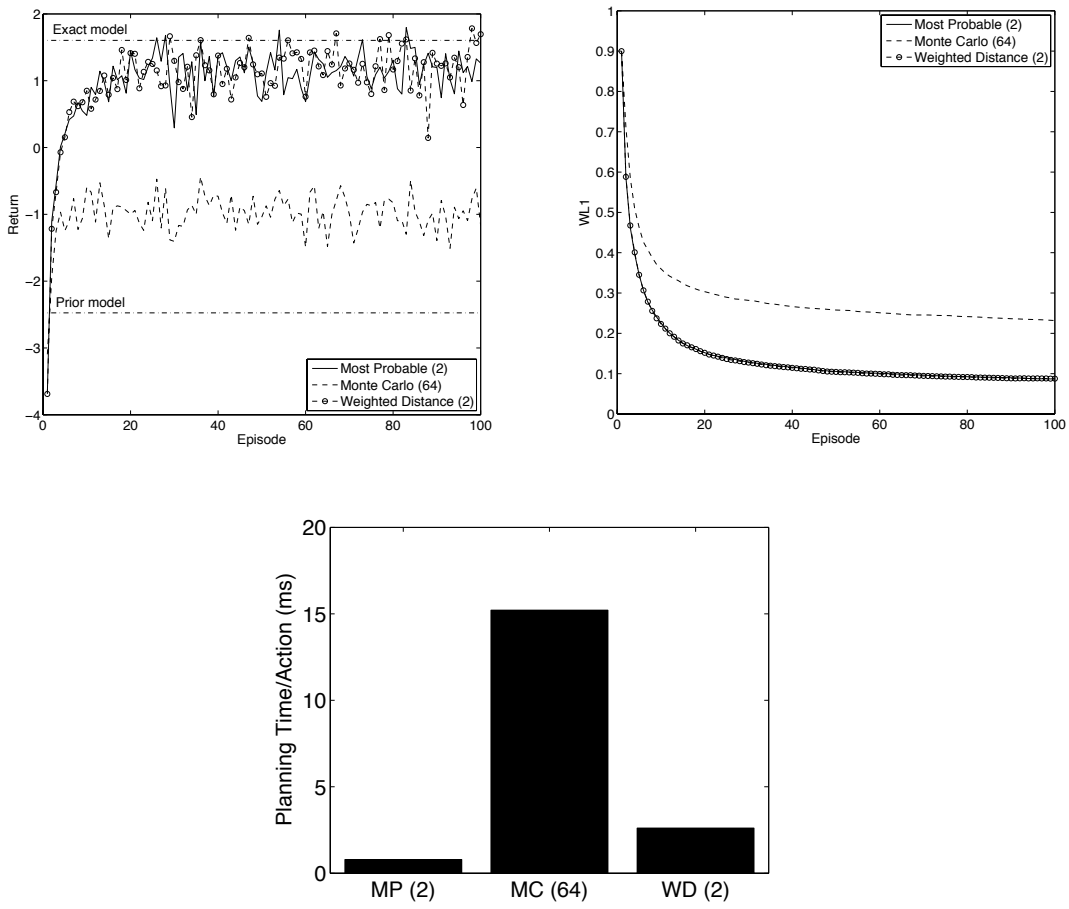

Figure 1: Tiger: Empirical return (top left), belief estimation error (top right), and planning time (bottom), for different belief tracking approximation.

6.2 Follow

We also consider a new POMDP domain, called Follow, inspired by an interactive human-robot task. It is often the case that such domains are particularly subject to parameter uncertainty (due to the difficulty in modeling human behavior), thus this environment motivates the utility of Bayes-Adaptive POMDP in a very practical way. The goal of the Follow task is for a robot to continuously follow one of two individuals in a 2D open area. The two subjects have different motion behavior, requiring the robot to use a different policy for each. At every episode, the target person is selected randomly with $Pr = 0.5$ (and the other is not present). The person's identity is not observable (except through their motion). The state space has two features: a binary variable indicating which per-

son is being followed, and a position variable indicating the person's position relative to the robot (5×5 square grid with the robot always at the center). Initially, the robot and person are at the same position. Both the robot and the person can perform five motion actions $\{NoAction, North, East, South, West\}$. The person follows a fixed stochastic policy (stationary over space and time), but the parameters of this behavior are unknown. The robot perceives observations indicating the person's position relative to the robot: $\{Same, North, East, South, West, Unseen\}$. The robot perceives the correct observation $Pr = 0.8$ and $Unseen$ with $Pr = 0.2$. The reward $R = +1$ if the robot and person are at the same position (central grid cell), $R = 0$ if the person is one cell away from the robot, and $R = -1$ if the person is two cells away. The task terminates if the person reaches a distance of 3 cells away from the robot, also causing a reward of -20. We use a discount factor of 0.9.

When formulating the BAPOMDP, the robot's motion model (deterministic), the observation probabilities, and the rewards are all assumed to be known. However we consider the case where each person's motion model is unknown. We maintain a separate count vector for each person, representing the number of times they move in each direction, that is, $\phi^1 = (\phi^1_{NA}, \phi^1_N, \phi^1_E, \phi^1_S, \phi^1_W)$, $\phi^2 = (\phi^2_{NA}, \phi^2_N, \phi^2_E, \phi^2_S, \phi^2_W)$. We assume a prior $\phi^1_0 = (2, 3, 1, 2, 2)$ for person 1 and $\phi^2_0 = (2, 1, 3, 2, 2)$ for person 2, while in reality person 1 moves with probabilities $Pr = (0.3, 0.4, 0.2, 0.05, 0.05)$ and person 2 with $Pr = (0.1, 0.05, 0.8, 0.03, 0.02)$. We run 200 simulations, each consisting of 100 episodes (of at most 10 time steps). The count vectors' distributions are reset after every simulation, and the target person is reset after every episode. We use a 2-step lookahead search for planning in the BAPOMDP.

Figure 2 shows how the average return and model accuracy evolve over the 100 episodes (averaged over the 200 simulations) with different belief approximations. Figure 2 also compares the planning time taken by each approach. We observe from these figures that the results for the Weighted Distance approximations are much better both in terms of return and model accuracy, even with fewer particles (16). Monte-Carlo fails at providing any improvement over the prior model, which indicates it would require much more particles. Running Weighted Distance with 16 particles require less time than both Monte-Carlo and Most Probable with 64 particles, showing that it can be more time efficient for the performance it provides in complex environment.

6.3 RockSample

To test our algorithm against problems with a larger number of states, we consider the RockSample problem (Smith and Simmons, 2004). In this domain, a robot is on an $n \times n$ square board, with rocks on some of the cells. Each rock has an unknown binary quality (good or bad). The goal of the robot is to gather samples of the good rocks. Sampling a good rock yields high reward (+10), in contrast to sampling a bad rock (-10). However a sample can only be acquired when the robot is in the same cell as the rock. The number of rocks and their respective positions are fixed and known, while their qualities are fixed but unknown. A state is defined by the position of the robot on the board and the quality of all the rocks. With an $n \times n$ board and k rocks, the number of states is then $n^2 2^k$. Most results below assume $n = 3$ and $k = 2$, which makes 36 states. The robot can choose

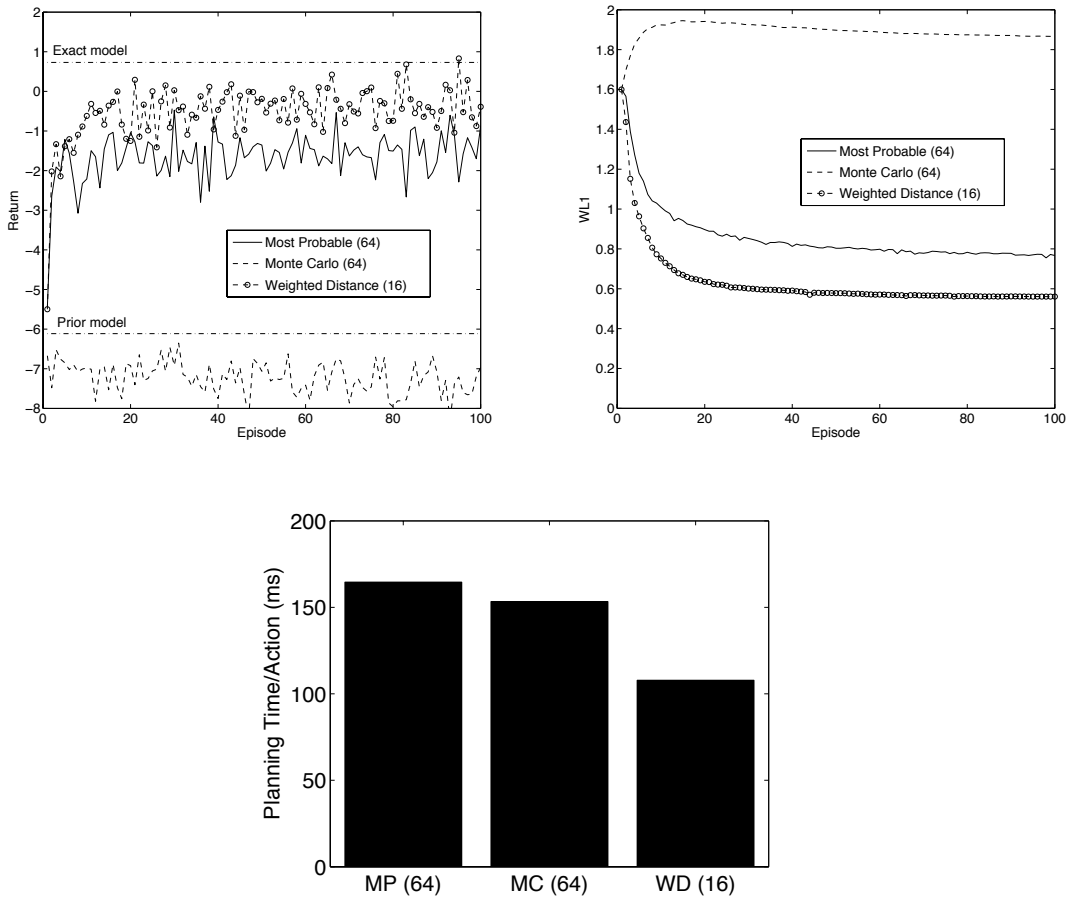

Figure 2: Follow: Empirical return (top left), belief estimation error (top right), and planning time (bottom), for different belief tracking approximation.

between 4 (deterministic) motion actions to move to neighboring cells, the Sample action, and a Sensor action for each rock, so there are $k+5$ actions in general. The robot is able to acquire information on the quality of each rock by using the corresponding sensor action. The sensor returns either GOOD or BAD, according to the quality of the rock. The sensor can be used when the robot is away from the rock, but the accuracy depends on the distance d between the robot and the rock. As in the original problem, the accuracy η of the sensor is given by $\eta = 2^{-d/d_0}$.

6.3.1 Influence of Large Number of States

We consider the case where transition probabilities are known, and the agent must learn its observation function. The prior knowledge over the structure of the observation function is as follows:

- the probability distribution over observations after performing action CHECK$_i$ in state s depends only on the distance between the robot and the rock i;

- at a given distance d, the probability of observing GOOD when the rock is a good one is equal to the probability of observing BAD when the rock is a bad one. This means that for each distance d, the robot's sensor has a probability to give incorrect observations, which doesn't depend of the quality of the rock.

These two assumptions seem reasonable in practice, and allow the robot to learn a model efficiently without having to try all CHECK actions in all states.

We begin by comparing performance of the BAPOMDP framework with different belief approximations. For the belief tracking, we focus on the Most Probable and Weighted Distance Minimization approximations, knowing that the Monte Carlo has given poor results in the two smaller domains. Each simulation consists of 100 episodes, and the results are averaged over 100 simulations.

As we can see in Figure 3(left), the Most Probable approximation outperforms Weighted Distance Minimization; in fact, after only 50 iterations, it reaches the same level of performance as a robot that knows the true model. Figure 3(right) sheds further light on this issue, by showing, for each episode, the maximum L_1 distance between the estimated belief $\hat{b}(s) = \sum_{\psi,\phi} b(s,\phi,\psi)$, and the correct belief $b(s)$ (assuming the model is known *a priori*). We see that this distance decreases for both approximations, and that it reaches values close to 0 after 50 episodes for the Most Probable approximation. This suggests that the robot has reached a point where it knows its model well enough to have the same belief over the physical states as a robot who would know the true model. Note that the error in belief estimate is calculated over the trajectories; it is possible that the estimated model is wrong in parts of the beliefs which are not visited under the current (learned) policy.

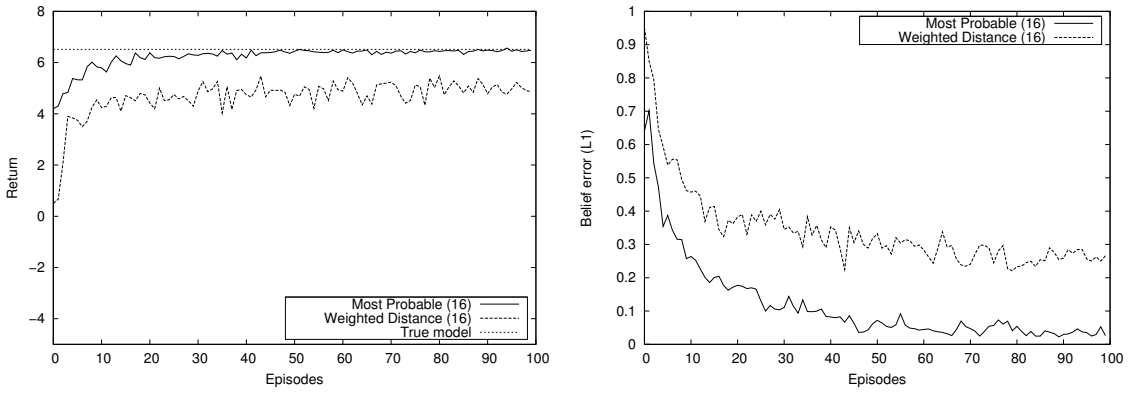

Figure 3: RockSample: Empirical return (left) and belief estimation error (right) for different belief tracking approximation.

To further verify the scalability of our approach, we consider larger versions of the RockSample domain in Figure 4. Recall that for k rocks and an $n \times n$ board, the domain has state space $|S| = n^2 2^k$ and action space $|A| = 5+k$. For this experiment, and all subsequent

ones, belief tracking in the BAPOMDP is done with the Most Probable approximation (with $K = 16$). As expected, the computational time for planning grows quickly with n and k. Better solutions could likely be obtained with appropriate use of heuristics in the forward search planner (Ross et al., 2008c).

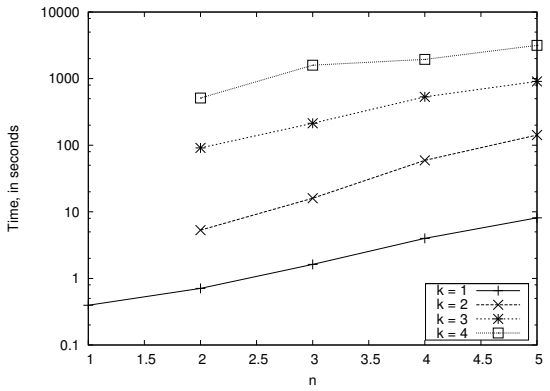

Figure 4: RockSample: Computational time for different values of k and n. All results are computed with $K = 16$ and a depth=3 planning horizon.

6.3.2 Influence of the Priors

The choice of prior plays an important role in Bayesian Learning. As explained above, in the RockSample domain we have constrained the structure of the observation probability model structural assumptions in the prior. For all results presented above, we used a prior made of 4 ϕ-vectors with probability $\frac{1}{4}$ each. Each of those vectors ϕ_i is made of coefficients (ϕ_{ij}), where ϕ_{ij} is the probability that the sensor will give a correct observation at distance j. For each of the 4 vectors ϕ_i, we sample the coefficients ϕ_{ij} from an uniform distribution between 0.45 and 0.95. We adopt this approach for a number of reasons. First, this prior is very general, in assuming that the sensor's probability to make a mistake is uniformly distributed between 0.05 and 0.55, at every distance d. Second, by sampling a new prior for every simulation, we ensure that the results do not depend closely on inadvertent similarities between our prior and the correct model.

We now consider two other forms of prior. First, we consider the case where the coefficients ϕ_{ij} are not sampled uniformly from $\mathcal{U}_{[0.45,0.95]}$, but rather from $\mathcal{U}_{[\phi^*_j \pm \varepsilon]}$, where ϕ^*_j is the value of the true model (that is, the probability that the true sensor gives a true observation at distance j). We consider performance for various levels of noise, $0 \leq \varepsilon \leq 0.25$. This experiment allows us to measure the influence of prior uncertainty on the performances of our algorithm. The results in Figure 5 show that the BAPOMDP agent performs well for various levels of initial uncertainty over the model. As expected, the fact that all the priors have ϕ_{ij} coefficients centered around the true value ϕ^*_j carries in itself substantial information, in many cases enough for the robot to perform very well from the first episode (note that the y-axis in Fig. 5 is different than that in Fig. 3). Furthermore, we observe

that the noise has very little influence on the performances of the robot: for all values of ε, the empirical return is above 6.3 after only 30 episodes.

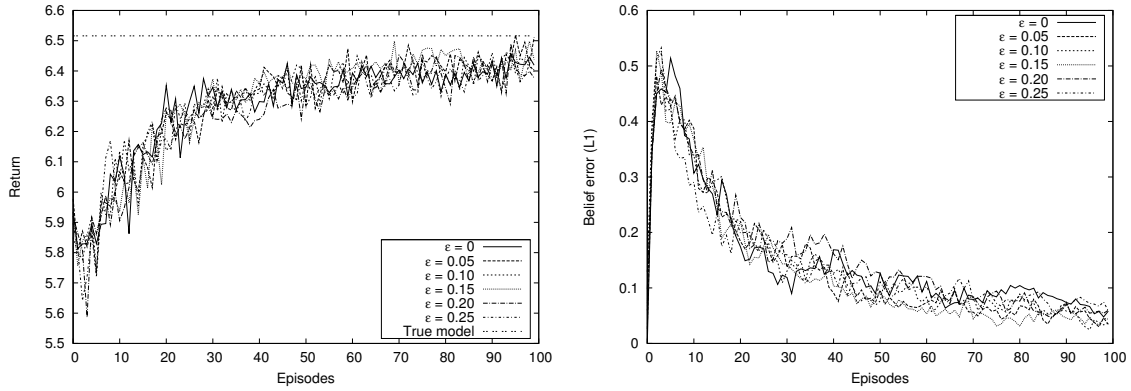

Figure 5: Performance of BAPOMDP with centered uniform priors in RockSample domain, using the Most Probably (K=16) belief tracking approximation. Empirical return (left). Belief state tracking error (right).

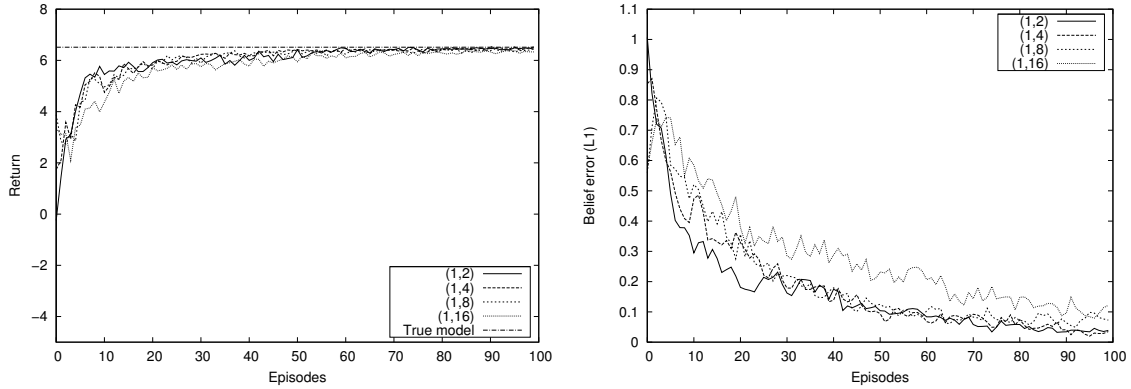

Figure 6: Performance of BAPOMDP with Beta priors in RockSample domain, using the Most Probable (K=16) belief tracking approximation. Empirical return (left). Belief state tracking error (right).

Second, we consider the case where there is only one ϕ-vector, which has probability one. This vector has coefficients ϕ_j, such that for all j, $\phi_j = \frac{k-1}{k}$, for different values of k. This represents a beta distribution of parameters $(1, k)$, where 1 is the count of wrong observations, and k the count of correct observations. The results presented in Figure 6 show that for all values of k, the rewards converge towards the optimal value within 100 episodes. We see that for high values of k, the robot needs more time to converge towards optimal rewards. Indeed, those priors have a large total count (k+1), which means their variance is small. Thus, they need more time to correct themselves towards the true model.

In particular, the $(1, 16)$ is very optimistic (it considers that the sensor only makes an error with probability $\frac{1}{17}$), which causes the robot to make mistakes during the first experiments, thus earning poor rewards at the beginning, and needing about 80 episodes to learn a sufficiently good model to achieve near-optimal performance. The right-side graph clearly shows how the magnitude of the initial k impacts the error in the belief over physical states (indicating that the robot doesn't know the quality of the rocks as well as if it knew the correct model). The error in belief state tracking is significantly reduced after about 80 iterations, confirming that our algorithm is able to overcome poor priors, even those with high initial confidence.

Finally, we consider the case where the true underlying POMDP model is changed such that the sensor has a constant probability ε of making mistakes for all distances; the prior is sampled as for the results of Figure 3. This makes the situation harder for the robot, because it increases its sensor's overall probability of making mistakes, including at distance zero (i.e., when the robot is on the same cell as the rock). The empirical results presented in Figure 7 show a decrease in the empirical return as ε increases. Similarly, as shown in the right graph, the learning performance suffers with higher values of ε. This is not surprising since a higher ε indicates that the robot's CHECKs are more prone to error, which makes it more difficult for the robot to improve its knowledge about its physical states, and thus about its model. In fact, it is easy to verify that the optimal return (assuming a fully known model) is lower for the noisier model. In general, in domains where the observations are noisy or aliased, it is difficult for the agent to learn a good model, as well as perform well (unless the observations are not necessary for good performance).

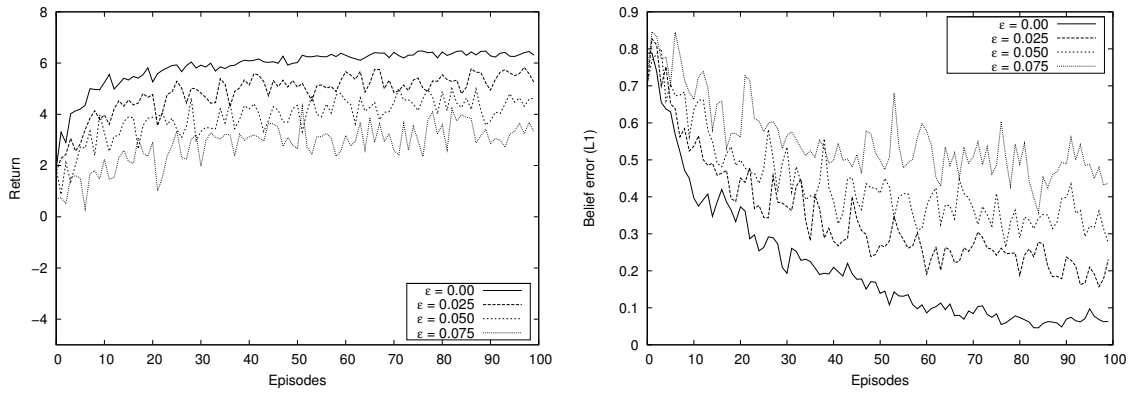

Figure 7: Performance of BAPOMDP with varying observation models in RockSample domain. Empirical return (left). Belief error (right).

7. Related Work

A few recent approaches have tackled the problem of joint planning and learning under partial (state and model) observability using a Bayesian framework. The work of Poupart and Vlassis (2008) is probably closest to the BAPOMDP outlined here. Using a similar Bayesian representation of model uncertainty, they proposed an extension of the Beetle al-

gorithm (Poupart et al., 2006) (original designed for fully observable domains) to compute an approximate solution for BAPOMDP-type problems. Their work is presented in the context of factored representations, but the model learning is done using similar Bayesian mechanisms, namely by updating a posterior represented by a mixture of Dirichlet distributions. They outline approximation methods to maintain a compact belief set that are similar to the Most Probably and Monte-Carlo methods outlined in Section 5.1 above. Presumably our Weighted Distance minimization technique could be extended to their factored representation, assuming one can compute the distance metric. Finally, they propose an offline planning algorithm, similar to the literature on point-based POMDP solvers, to find a policy. However we are not aware of any empirical validation with this approach, thus scalability and expressivity in experimental domains remains to be determined.

Jaulmes et al. (2005) have for their part considered active learning in partially observable domains where information gathering actions are provided by oracles that reveal the underlying state. The key assumption of this approach, which is not used in other model-free approaches, concerns the existence of this oracle (or human) which is able to correctly identify the state following each transition. This makes it much easier to know how to update the prior. In the same vein than Jaulmes and colleagues, Doshi et al. (2008) developed an approach for active learning in POMDPs that can robustly determine a near-optimal policy. To achieve that, they introduced meta-queries (questions about action) and a risk-averse action selection criterion that allows agents to behave robustly even with uncertain knowledge of the POMDP model. Finally, Doshi-Velez (2010) proposed a Bayesian learning framework for the case of POMDPs where the number of states is not known *a priori*, thus allowing the number of states to grow gradually as the agent explores the world, while simultaneously updating a posterior over the parameters.

The work on Universal Artificial Intelligence (Hutter, 2005) presents an interesting alternative to the framework of BAPOMDPs. It tackles a similar problem, namely sequential decision-making under (general) uncertainty. But Hutter's AIXI framework is more general, in that it allows the model to be sampled from any computable distribution. The learning problem is constrained by placing an Occam's razor prior (measured by Kolmogorov complexity) over the space of models. The main drawback is that inference in this framework is incomputable, though an algorithm is presented for computing time/space-bounded solutions. Further development of a general purpose AIXI learning/planning algorithm would be necessary to allow a direct comparison between AIXI and BAPOMDPs on practical problems. Recent results in Monte-Carlo Planning provide a good basis for this (Silver and Veness, 2010; Veness et al., 2011).

A number of useful theoretical results have also been published recently. For the specific case of exploration in reinforcement learning, Asmuth et al. (2009) presented a fully Bayesian analysis of the performance of a sampling approach. Subsequently, Kolter and Ng (2009) clarified the relation between Bayesian and PAC-MDP approaches and presented a simple algorithm for efficiently achieving near-Bayesian exploration.

Finally, it is worth emphasizing that Bayesian approaches have also been investigated in the control literature. The problem of optimal control under uncertain model parameters was originally introduced by Feldbaum (1961), as the theory of dual control, also sometimes referred to as adaptive dual control. Extensions of this theory have been developed for time-varying systems (Filatov and Unbehauen, 2000). Several authors have studied this problem

for different kinds of dynamical systems : linear time invariant systems under partial observability (Rusnak, 1995), linear time varying Gaussian models under partial observability (Ravikanth et al., 1992), nonlinear systems with full observability (Zane, 1992), and more recently a non linear systems under partial observability (Greenfield and Brockwell, 2003). All this work is targeted towards specific classes of continuous systems, and we are not aware of similar work in the control literature for discrete (or hybrid) systems.

8. Conclusion

The problem of sequential decision-making under model uncertainty arises in many practical applications of AI and decision systems. Developing effective models and algorithms to handle these problems under realistic conditions—including stochasticity, partial state observability, and model inaccuracy—is imperative if we hope to deploy robots and other autonomous agents in real-world situations.

This paper focuses in particular on the problem of simultaneous learning and decision-making in dynamic environments under partial model and state uncertainty. We adopt a model-based Bayesian reinforcement learning framework, which allows us to explicitly target the exploration-exploitation problem in a coherent mathematical framework. Our work is a direct extension of previous results on model-based Bayesian reinforcement learning in fully observable domains.

The main contributions of the paper pertains to the development of the Bayes-Adaptive POMDP model, and analysis of its theoretical properties. This work addresses a number of interesting questions, including:

1. defining an appropriate model for POMDP parameter uncertainty,
2. approximating this model while maintaining performance guarantees,
3. performing tractable belief updating, and
4. optimizing action sequences given a posterior over state and model uncertainty.

From the theoretical analysis, we are able to derive simple algorithms for belief tracking and (near-)optimal decision-making in this model. We illustrate performance of these algorithms in a collection of synthetic POMDP domains. Results in the Follow problem showed that our approach is able to learn the motion patterns of two (simulated) individuals. This suggests interesting applications in human-robot interaction, where we often lack good models of human behavior and where it is imperative that an agent be able to learn quickly, lest the human user lose interest (this is in contrast to robot navigation tasks, for which we often have access to more precise dynamical models and/or high-fidelity simulators). For their part, results of RockSample problem shows how one should take into account prior knowledge on agent's sensors when this knowledge is available.

While the BAPOMDP model provides a rich model for sequential decision-making under uncertainty, it has a number of important limitations. First, the model and theoretical analysis are limited to discrete domains. It is worth noting however that the approximate algorithms extend quite easily to the continuous case (Ross et al., 2008b), at least for some

families of dynamical systems. Other related references for the continuous case are available in the control literature, as described in Section 7.

Another limitation is the fact that the model requires specification of a prior. This is standard in the Bayesian RL framework. The main concern is to ensure that the prior assigns some weight to the correct model. Our empirical evaluation shows good performance for a range of priors; though the issue of choosing good priors in large domains remains a challenge in general. Our empirical results also confirm standard Bayesian intuition, whereby the influence of the prior is particularly important for any inference and decision-making performed when only a small amount of data has been observed, but the influence becomes negligible as large amounts of data are acquired.

As a word of caution, problems may arise in cases where Bayesian RL is used to infer both transition and observation probabilities simultaneously, while the rewards are not explicitly perceived through the observations (even if the rewards are known *a priori*). In this challenging setting, the Bayes-Adaptive POMDP framework as outlined above might converge to an incorrect model if the initial priors on the transition and observation model are non-informative. This is mainly due to the fact that many possible parameters may correctly explain the observed action-observation sequences. While the agent is able to predict observations correctly, this leads to poor prediction of rewards and thus possibly sub-optimal long-term rewards. However if the rewards are observable, and their probabilities taken into account in the belief update, such problems do not arise, in the sense that the agent learns an equivalent model that correctly explains the observed action-observation-reward sequence and recovers a good policy for the unknown POMDP model. In the latter case, where rewards are observable, the framework presented in this paper can be used with only minor modifications to also learn the reward function.

Finally, it is worth pointing out that Bayesian RL methods in general have not been deployed in real-world domains yet. We hope that the work presented here will motivate further investigation of practical issues pertaining to the application and deployment of this class of learning approaches.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT), and the National Institutes of Health (grant R21DA019800). We would also like to thank Michael Bowling, Patrick Dallaire, Mohammad Ghavamzadeh, Doina Precup, Prakash Panangaden, as well as the anonymous reviewers, for offering thoughtful comments on earlier versions of this work.

Appendix A. Theorems and Proofs

This appendix presents the proofs of the theorems presented throughout this paper. Theorems 1 and 2 are presented first, then some useful lemmas, followed by the proofs of the remaining Theorems.

Theorem 1 *Let $(S', A, Z, T', O', R', \gamma)$ be a BAPOMDP constructed from the POMDP $(S, A, Z, T, O, R, \gamma)$. If S is finite, then at any time t, the set $S'_{b'_t} = \{\sigma \in S' | b'_t(\sigma) > 0\}$ has size $|S'_{b'_t}| \leq |S|^{t+1}$.*

Proof Proof by induction. When $t = 0$, $b'_0(s, \phi, \psi) > 0$ only if $\phi = \phi_0$ and $\psi = \psi_0$. Hence $|S'_{b'_0}| \leq |S|$. For the general case, assume that $|S'_{b'_{t-1}}| \leq |S|^t$. From the definitions of the belief update function, $b'_t(s', \phi', \psi') > 0$ iff $\exists (s, \phi, \psi)$ such that $b'_{t-1}(s, \phi, \psi) > 0$, $\phi' = \phi + \delta^a_{ss'}$ and $\psi' = \psi + \delta^a_{s'z}$. Hence, a particular (s, ϕ, ψ) such that $b'_{t-1}(s, \phi, \psi) > 0$ yields non-zero probabilities to at most $|S|$ different states in b'_t. Since $|S'_{b'_{t-1}}| \leq |S|^t$ by assumption, then if we generate $|S|$ different probable states in b'_t, for each probable state in $S'_{b_{t-1}}$ it follows that $|S'_{b'_t}| \leq |S|^{t+1}$. ■

Theorem 2 *For any horizon t, there exists a finite set Γ_t of functions $S' \to \mathbb{R}$, such that $V^*_t(b) = \max_{\alpha \in \Gamma_t} \sum_{\sigma \in S'} \alpha(\sigma) b(\sigma)$.*

Proof Proof by induction. This holds true for horizon $t = 1$, since $V^*_1(b) = \max_{a \in A} \sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a)$. Hence by defining $\Gamma_1 = \{\alpha_a | \alpha_a(s, \phi, \psi) = R(s, a), a \in A\}$, $V^*_1(b) = \max_{\alpha \in \Gamma_1} \sum_{\sigma \in S'} b(\sigma)\alpha(\sigma)$. By induction, we assume that there exists a set Γ_t such that $V^*_t(b) = \max_{\alpha \in \Gamma_t} \sum_{\sigma \in S'} b(\sigma)\alpha(\sigma)$.
Now $V^*_{t+1}(b) = \max_{a \in A} \left[\sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a) + \sum_{z \in Z} \Pr(z|b, a) V^*_t(b^{az}) \right]$. Hence:

$$\begin{aligned}
V^*_{t+1}(b) &= \max_{a \in A} \left[\sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a) + \sum_{z \in Z} \Pr(z|b, a) \max_{\alpha \in \Gamma_t} \sum_{\sigma \in S'} b^{az}(\sigma)\alpha(\sigma) \right] \\
&= \max_{a \in A} \left[\sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a) + \sum_{z \in Z} \max_{\alpha \in \Gamma_t} \sum_{\sigma \in S'} \Pr(z|b, a) b^{az}(\sigma)\alpha(\sigma) \right] \\
&= \max_{a \in A} \Big[\sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a) + \\
&\quad \sum_{z \in Z} \max_{\alpha \in \Gamma_t} \sum_{(s,\phi,\psi) \in S'} \sum_{s' \in S} b(s, \phi, \psi) T^{sas'}_{\phi} O^{s'az}_{\psi} \alpha(s', \mathcal{U}(\phi, s, a, s'), \mathcal{U}(\psi, s', a, z)) \Big].
\end{aligned}$$

Thus if we define:

$$\begin{aligned}
\Gamma_{t+1} = \{ &\alpha_{a,f} | \alpha_{a,f}(s, \phi, \psi) = R(s, a) + \\
&\sum_{z \in Z} \sum_{s' \in S} T^{sas'}_{\phi} O^{s'az}_{\psi} f(z)(s', \mathcal{U}(\phi, s, a, s'), \mathcal{U}(\psi, s', a, z)), a \in A, f \in [Z \to \Gamma_t] \},
\end{aligned}$$

then $V^*_{t+1}(b) = \max_{\alpha \in \Gamma_{t+1}} \sum_{\sigma \in S'} b(\sigma)\alpha(\sigma)$ and Γ_{t+1} is finite since $|\Gamma_{t+1}| = |A||\Gamma_t|^{|Z|}$, which is finite by assumptions that A, Z and Γ_t are all finite. ■

For some of the following theorems, lemmas and proofs, we will sometime denote the Dirichlet count update operator $\mathcal{U}$, as defined for the BAPOMDP, as a vector addition: $\phi' = \phi + \delta^a_{ss'} = \mathcal{U}(\phi, s, a, s')$, that is, $\delta^a_{ss'}$ is a vector full of zeros, with a 1 for the element $\phi^a_{ss'}$.

Lemma 1 *For any $t \geq 2$, any α-vector $\alpha_t \in \Gamma_t$ can be expressed as $\alpha^{a,\alpha'}_t(s, \phi, \psi) = R(s, a) + \gamma \sum_{z \in Z} \sum_{s \in S'} T^{sas'}_{\phi} O^{s'az}_{\psi} \alpha'(z)(s', \phi + \delta^a_{ss'}, \psi + \delta^a_{s'z})$ for some $a \in A$, and α' defining a mapping $Z \to \Gamma_{t-1}$.*

Proof Follows from proof of theorem 2. ■

Lemma 2 *Given any* $a, b, c, d \in \mathbb{R}$, $ab - cd = \frac{(a-c)(b+d)+(a+c)(b-d)}{2}$.

Proof Follows from direct computation. ■

Lemma 3 *Given any* $\phi, \phi' \in \mathcal{T}$, $\psi, \psi' \in \mathcal{O}$, *then for all* $s \in S$, $a \in A$, *we have that* $\sum_{s' \in S} \sum_{z \in Z} \left| \frac{\phi'^a_{ss'} \psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'} \mathcal{N}^{s'a}_{\psi'}} - \frac{\phi^a_{ss'} \psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi} \mathcal{N}^{s'a}_{\psi}} \right| \leq D^{sa}_S(\phi', \phi) + \sup_{s' \in S} D^{s'a}_Z(\psi', \psi)$.

Proof Using lemma 2, we have that:

$$
\begin{aligned}
& \sum_{s' \in S} \sum_{z \in Z} \left| \frac{\phi'^a_{ss'} \psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'} \mathcal{N}^{s'a}_{\psi'}} - \frac{\phi^a_{ss'} \psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi} \mathcal{N}^{s'a}_{\psi}} \right| \\
= \;& \tfrac{1}{2} \sum_{s' \in S} \sum_{z \in Z} \left| \left(\frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} - \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right) \left(\frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} + \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right) + \left(\frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} + \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right) \left(\frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} - \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right) \right| \\
\leq \;& \tfrac{1}{2} \sum_{s' \in S} \left| \frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} - \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right| \sum_{z \in Z} \left| \frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} + \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right| + \tfrac{1}{2} \sum_{s' \in S} \left| \frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} + \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right| \sum_{z \in Z} \left| \frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} - \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right| \\
\leq \;& \sum_{s' \in S} \left| \frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} - \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right| + \tfrac{1}{2} \left[\sup_{s' \in S} \sum_{z \in Z} \left| \frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} - \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right| \right] \left[\sum_{s' \in S} \left| \frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} + \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right| \right] \\
= \;& \sum_{s' \in S} \left| \frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} - \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right| + \sup_{s' \in S} \sum_{z \in Z} \left| \frac{\psi'^a_{s'z}}{\mathcal{N}^{s'a}_{\psi'}} - \frac{\psi^a_{s'z}}{\mathcal{N}^{s'a}_{\psi}} \right| \\
= \;& D^{sa}_S(\phi', \phi) + \sup_{s' \in S} D^{s'a}_Z(\psi', \psi).
\end{aligned}
$$

■

Lemma 4 *Given any* $\phi, \phi', \Delta \in \mathcal{T}$, *then for all* $s \in S$, $a \in A$, $D^{sa}_S(\phi + \Delta, \phi' + \Delta) \leq D^{sa}_S(\phi, \phi') + \frac{2\mathcal{N}^{sa}_\Delta \sum_{s' \in S} |\phi^a_{ss'} - \phi'^a_{ss'}|}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)}$.

Proof We have that:

$$
\begin{aligned}
& D^{sa}_S(\phi + \Delta, \phi' + \Delta) \\
= \;& \sum_{s' \in S} \left| \frac{\phi^a_{ss'} + \Delta^a_{ss'}}{\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta} - \frac{\phi'^a_{ss'} + \Delta^a_{ss'}}{\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta} \right| \\
= \;& \sum_{s' \in S} \left| \frac{(\phi^a_{ss'} + \Delta^a_{ss'})(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta) - (\phi'^a_{ss'} + \Delta^a_{ss'})(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \right| \\
= \;& \sum_{s' \in S} \left| \frac{\phi^a_{ss'} \mathcal{N}^{sa}_{\phi'} + \phi^a_{ss'} \mathcal{N}^{sa}_\Delta + \Delta^a_{ss'} \mathcal{N}^{sa}_{\phi'} - \phi'^a_{ss'} \mathcal{N}^{sa}_\phi - \phi'^a_{ss'} \mathcal{N}^{sa}_\Delta - \Delta^a_{ss'} \mathcal{N}^{sa}_\phi}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \right| \\
\leq \;& \sum_{s' \in S} \left| \frac{\phi^a_{ss'} \mathcal{N}^{sa}_{\phi'} - \phi'^a_{ss'} \mathcal{N}^{sa}_\phi}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \right| + \sum_{s' \in S} \left| \frac{\mathcal{N}^{sa}_\Delta (\phi^a_{ss'} - \phi'^a_{ss'}) + \Delta^a_{ss'} (\mathcal{N}^{sa}_{\phi'} - \mathcal{N}^{sa}_\phi)}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \right| \\
\leq \;& \sum_{s' \in S} \left| \frac{\phi^a_{ss'} \mathcal{N}^{sa}_{\phi'} - \phi'^a_{ss'} \mathcal{N}^{sa}_\phi}{\mathcal{N}^{sa}_\phi \mathcal{N}^{sa}_{\phi'}} \right| + \frac{\mathcal{N}^{sa}_\Delta \left[\sum_{s' \in S} \left| \phi^a_{ss'} - \phi'^a_{ss'} \right| \right] + \left| \mathcal{N}^{sa}_{\phi'} - \mathcal{N}^{sa}_\phi \right| \sum_{s' \in S} \Delta^a_{ss'}}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \\
= \;& D^{sa}_S(\phi, \phi') + \frac{\mathcal{N}^{sa}_\Delta \left[\sum_{s' \in S} \left| \phi^a_{ss'} - \phi'^a_{ss'} \right| \right] + \mathcal{N}^{sa}_\Delta \left| \mathcal{N}^{sa}_{\phi'} - \mathcal{N}^{sa}_\phi \right|}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)} \\
\leq \;& D^{sa}_S(\phi, \phi') + \frac{2\mathcal{N}^{sa}_\Delta \sum_{s' \in S} \left| \phi^a_{ss'} - \phi'^a_{ss'} \right|}{(\mathcal{N}^{sa}_\phi + \mathcal{N}^{sa}_\Delta)(\mathcal{N}^{sa}_{\phi'} + \mathcal{N}^{sa}_\Delta)}.
\end{aligned}
$$

■

Lemma 5 *Given any $\psi, \psi', \Delta \in \mathcal{O}$, then for all $s \in S$, $a \in A$,*
$D_Z^{sa}(\psi + \Delta, \psi' + \Delta) \leq D_Z^{sa}(\psi, \psi') + \frac{2\mathcal{N}_\Delta^{sa} \sum_{z \in Z} |\psi_{sz}^a - \psi_{sz}'^a|}{(\mathcal{N}_\psi^{sa} + \mathcal{N}_\Delta^{sa})(\mathcal{N}_{\psi'}^{sa} + \mathcal{N}_\Delta^{sa})}$.

Proof Same proof as for lemma 4, except that we sum over $z \in Z$ in this case. ■

Lemma 6 *Given any $\gamma \in (0,1)$, then $\sup_x \gamma^{x/2} x = \frac{2}{\ln(\gamma^{-e})}$.*

Proof We observe that when $x = 0$, $\gamma^{x/2} x = 0$ and $\lim_{x \to \infty} \gamma^{x/2} x = 0$. Furthermore, $\gamma^{x/2}$ is monotonically decreasing exponentially as x increases, while x is monotonically increasing linearly as x increases. Thus it is clear that $\gamma^{x/2} x$ will have a unique global maximum in $(0, \infty)$. We can find this maximum by taking the derivative:

$$\begin{aligned} & \frac{\partial}{\partial x}(\gamma^{x/2} x) \\ = \; & \frac{(\ln \gamma)\gamma^{x/2} x}{2} + \gamma^{x/2} \\ = \; & \gamma^{x/2}\left(\frac{(\ln \gamma) x}{2} + 1\right). \end{aligned}$$

Hence by solving when this is equal 0, we have:

$$\begin{aligned} & \gamma^{x/2}\left(\frac{(\ln \gamma) x}{2} + 1\right) = 0 \\ \Leftrightarrow \; & \frac{(\ln \gamma) x}{2} + 1 = 0 \\ \Leftrightarrow \; & x = \frac{-2}{\ln \gamma} = -2 \log_\gamma(e). \end{aligned}$$

Hence we have that:

$$\begin{aligned} & \gamma^{x/2} x \\ \leq \; & -2\gamma^{-\log_\gamma(e)} \log_\gamma(e) \\ = \; & -2e^{-1} \log_\gamma(e) \\ = \; & \frac{2}{\ln(\gamma^{-e})}. \end{aligned}$$

■

Lemma 7 $\sup_{\alpha_1 \in \Gamma_1, s \in S} |\alpha_1(s, \phi, \psi) - \alpha_1(s, \phi', \psi')| = 0$ *for any ϕ, ϕ', ψ, ψ'.*

Proof For any $a \in A$, $s \in S$, $|\alpha_1^a(s, \phi, \psi) - \alpha_1^a(s, \phi', \psi')| = |R(s,a) - R(s,a)| = 0$. ■

Theorem 3 *Given any $\phi, \phi' \in \mathcal{T}$, $\psi, \psi' \in \mathcal{O}$ and $\gamma \in (0,1)$, then $\forall t$:*

$$\sup_{\alpha_t \in \Gamma_t, s \in S} |\alpha_t(s, \phi, \psi) - \alpha_t(s, \phi', \psi')| \leq \frac{2\gamma ||R||_\infty}{(1-\gamma)^2} \sup_{s, s' \in S, a \in A} \Big[D_S^{sa}(\phi, \phi') + D_Z^{s'a}(\psi, \psi') + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s'' \in S} \left| \phi_{ss''}^a - \phi_{ss''}'^a \right|}{(\mathcal{N}_\phi^{sa} + 1)(\mathcal{N}_{\phi'}^{sa} + 1)} + \frac{\sum_{z \in Z} \left| \psi_{s'z}^a - \psi_{s'z}'^a \right|}{(\mathcal{N}_\psi^{s'a} + 1)(\mathcal{N}_{\psi'}^{s'a} + 1)} \right) \Big].$$

Proof Using lemma 1, we have that:

$$
\begin{aligned}
&|\alpha_t^{a,\alpha'}(s,\phi,\psi) - \alpha_t^{a,\alpha'}(s,\phi',\psi')| \\
&= \Big| R(s,a) + \gamma \textstyle\sum_{s'\in S}\sum_{z\in Z} \frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}}\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) \\
&\quad -R(s,a) - \gamma \textstyle\sum_{s'\in S}\sum_{z\in Z} \frac{\phi'^a_{ss'}\psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'}\mathcal{N}^{s'a}_{\psi'}}\alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\Big| \\
&= \gamma \Bigg| \textstyle\sum_{s'\in S}\sum_{z\in Z} \left[\frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}}\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \frac{\phi'^a_{ss'}\psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'}\mathcal{N}^{s'a}_{\psi'}}\alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z}) \right] \Bigg| \\
&= \gamma \Bigg| \textstyle\sum_{s'\in S}\sum_{z\in Z} \Bigg[\frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}}(\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})) \\
&\quad - \left(\frac{\phi'^a_{ss'}\psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'}\mathcal{N}^{s'a}_{\psi'}} - \frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}} \right) \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z}) \Bigg] \Bigg| \\
&\leq \gamma \textstyle\sum_{s'\in S}\sum_{z\in Z} \frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}} \left|\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| \\
&\quad + \gamma \textstyle\sum_{s'\in S}\sum_{z\in Z} \left| \frac{\phi'^a_{ss'}\psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'}\mathcal{N}^{s'a}_{\psi'}} - \frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}} \right| \left|\alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| \\
&\leq \gamma \sup_{s'\in S, z\in Z} \left|\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| \\
&\quad + \tfrac{\gamma||R||_\infty}{1-\gamma} \textstyle\sum_{s'\in S}\sum_{z\in Z} \left| \frac{\phi'^a_{ss'}\psi'^a_{s'z}}{\mathcal{N}^{sa}_{\phi'}\mathcal{N}^{s'a}_{\psi'}} - \frac{\phi^a_{ss'}\psi^a_{s'z}}{\mathcal{N}^{sa}_{\phi}\mathcal{N}^{s'a}_{\psi}} \right| \\
&\leq \gamma \sup_{s'\in S, z\in Z} \left|\alpha'(z)(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| \\
&\quad + \tfrac{\gamma||R||_\infty}{1-\gamma} \left(D_S^{sa}(\phi',\phi) + \sup_{s'\in S} D_Z^{s'a}(\psi',\psi) \right).
\end{aligned}
$$

The last inequality follows from lemma 3. Hence by taking the sup we get:

$$
\begin{aligned}
&\sup_{\alpha_t\in\Gamma_t, s\in S} |\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| \\
&\leq \gamma \sup_{s,s'\in S, a\in A, z\in Z, \alpha_{t-1}\in\Gamma_{t-1}} \left|\alpha_{t-1}(s',\phi+\delta^a_{ss'},\psi+\delta^a_{s'z}) - \alpha_{t-1}(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| \\
&\quad + \tfrac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \left(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \right).
\end{aligned}
$$

We notice that this inequality defines a recurrence. By unfolding it up to $t=1$ we get that:

$$
\begin{aligned}
&\sup_{\alpha_t\in\Gamma_t, s\in S} |\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| \\
&\leq \gamma^{t-1} \sup_{\alpha_1\in\Gamma_1, s'\in S, \Delta\in\mathcal{T}, \Delta'\in\mathcal{O} \mid ||\Delta||_1=||\Delta'||_1=(t-1)} |\alpha_1(s',\phi+\Delta,\psi+\Delta') - \alpha_1(s',\phi'+\Delta,\psi'+\Delta')| \\
&\quad + \tfrac{\gamma||R||_\infty}{1-\gamma} \textstyle\sum_{i=1}^{t-2} \gamma^i \sup_{s,s'\in S, a\in A, \Delta\in\mathcal{T}, \Delta'\in\mathcal{O} \mid ||\Delta||_1=||\Delta'||_1=i} \left(D_S^{sa}(\phi'+\Delta,\phi+\Delta) + D_Z^{s'a}(\psi'+\Delta',\psi+\Delta') \right) \\
&\quad + \tfrac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \left(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \right).
\end{aligned}
$$

Applying lemmas 7, 4 and 5 to the last term, we get that:

$$
\begin{aligned}
&\sup_{\alpha_t\in\Gamma_t, s\in S} |\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| \\
&\leq \frac{\gamma||R||_\infty}{1-\gamma} \sum_{i=1}^{t-2} \gamma^i \sup_{s,s'\in S, a\in A, \Delta\in\mathcal{T}, \Delta'\in\mathcal{O}|\ ||\Delta||_1=||\Delta'||_1=i} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \\
&\quad + \frac{2\mathcal{N}_\Delta^{sa} \sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+\mathcal{N}_\Delta^{sa})(\mathcal{N}_{\phi'}^{sa}+\mathcal{N}_\Delta^{sa})} + \frac{2\mathcal{N}_{\Delta'}^{s'a} \sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+\mathcal{N}_{\Delta'}^{s'a})(\mathcal{N}_{\psi'}^{s'a}+\mathcal{N}_{\Delta'}^{s'a})} \Big) \\
&\quad + \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \Big) \\
&= \frac{\gamma||R||_\infty}{1-\gamma} \sum_{i=1}^{t-2} \gamma^{i/2} \sup_{s,s'\in S, a\in A, \Delta\in\mathcal{T}, \Delta'\in\mathcal{O}|\ ||\Delta||_1=||\Delta'||_1=i} \Big(\gamma^{i/2} D_S^{sa}(\phi',\phi) + \gamma^{i/2} D_Z^{s'a}(\psi',\psi) \\
&\quad + \frac{2\gamma^{i/2}\mathcal{N}_\Delta^{sa} \sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+\mathcal{N}_\Delta^{sa})(\mathcal{N}_{\phi'}^{sa}+\mathcal{N}_\Delta^{sa})} + \frac{2\gamma^{i/2}\mathcal{N}_{\Delta'}^{s'a} \sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+\mathcal{N}_{\Delta'}^{s'a})(\mathcal{N}_{\psi'}^{s'a}+\mathcal{N}_{\Delta'}^{s'a})} \Big) \\
&\quad + \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \Big).
\end{aligned}
$$

Now we notice that $\gamma^{i/2} \leq \gamma^{\mathcal{N}_\Delta^{sa}/2}$ since $||\Delta||_1 = i$, and similarly $\gamma^{i/2} \leq \gamma^{\mathcal{N}_{\Delta'}^{sa}/2}$. Hence by applying lemma 6, we get that:

$$
\begin{aligned}
&\sup_{\alpha_t\in\Gamma_t, s\in S} |\alpha_t(s,\phi,\psi) - \alpha_t(s,\phi',\psi')| \\
&\leq \frac{\gamma||R||_\infty}{1-\gamma} \sum_{i=1}^{t-2} \gamma^{i/2} \sup_{s,s'\in S, a\in A, \Delta\in\mathcal{T}, \Delta'\in\mathcal{O}|\ ||\Delta||_1=||\Delta'||_1=i} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \\
&\quad + \frac{4 \sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{\ln(\gamma^{-e})(\mathcal{N}_\phi^{sa}+\mathcal{N}_\Delta^{sa})(\mathcal{N}_{\phi'}^{sa}+\mathcal{N}_\Delta^{sa})} + \frac{4 \sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{\ln(\gamma^{-e})(\mathcal{N}_\psi^{s'a}+\mathcal{N}_{\Delta'}^{s'a})(\mathcal{N}_{\psi'}^{s'a}+\mathcal{N}_{\Delta'}^{s'a})} \Big) \\
&\quad + \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \Big) \\
&\leq \frac{\gamma||R||_\infty}{1-\gamma} \sum_{i=1}^{t-2} \gamma^{i/2} \sup_{s,s'\in S, a\in A} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) + \frac{4 \sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{\ln(\gamma^{-e})(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} \\
&\quad + \frac{4 \sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{\ln(\gamma^{-e})(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \Big) + \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big(D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \Big) \\
&\leq \Big(\sum_{i=0}^{t-2} \gamma^{i/2} \Big) \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big[D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \\
&\quad + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Big] \\
&\leq \Big(\sum_{i=0}^{\infty} \gamma^{i/2} \Big) \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \Big[D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) \\
&\quad + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \Big] \\
&= \frac{1+\sqrt{\gamma}}{1-\gamma} \frac{\gamma||R||_\infty}{1-\gamma} \sup_{s,s'\in S, a\in A} \left[D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \right] \\
&\leq \frac{2\gamma||R||_\infty}{(1-\gamma)^2} \sup_{s,s'\in S, a\in A} \left[D_S^{sa}(\phi',\phi) + D_Z^{s'a}(\psi',\psi) + \frac{4}{\ln(\gamma^{-e})} \left(\frac{\sum_{s''\in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_{\phi'}^{sa}+1)} + \frac{\sum_{z\in Z} |\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}_\psi^{s'a}+1)(\mathcal{N}_{\psi'}^{s'a}+1)} \right) \right].
\end{aligned}
$$

■

Lemma 8 *Given $\phi \in \mathcal{T}$, $s \in S$, $a \in A$, then for all $\Delta \in \mathcal{T}$, $\frac{\sum_{s'\in S} |\phi^a_{ss'} - (\phi^a_{ss'} + \Delta^a_{ss'})|}{(\mathcal{N}_\phi^{sa}+1)(\mathcal{N}_\phi^{sa}+\mathcal{N}_\Delta^{sa}+1)} \leq \frac{1}{\mathcal{N}_\phi^{sa}+1}$.*

Proof

$$\begin{aligned}
&\frac{\sum_{s' \in S} |\phi^a_{ss'} - (\phi^a_{ss'} + \Delta^a_{ss'})|}{(\mathcal{N}^{sa}_{\phi}+1)(\mathcal{N}^{sa}_{\phi}+\mathcal{N}^{sa}_{\Delta}+1)} \\
= &\frac{\sum_{s' \in S} \Delta^a_{ss'}}{(\mathcal{N}^{sa}_{\phi}+1)(\mathcal{N}^{sa}_{\phi}+\mathcal{N}^{sa}_{\Delta}+1)} \\
= &\frac{1}{\mathcal{N}^{sa}_{\phi}+1}\left(\frac{\mathcal{N}^{sa}_{\Delta}}{\mathcal{N}^{sa}_{\Delta}+\mathcal{N}^{sa}_{\phi}+1}\right).
\end{aligned}$$

The term $\frac{\mathcal{N}^{sa}_{\Delta}}{\mathcal{N}^{sa}_{\Delta}+\mathcal{N}^{sa}_{\phi}+1}$ is monotonically increasing and converge to 1 as $\mathcal{N}^{sa}_{\Delta} \to \infty$. Thus the lemma follows. ■

Corollary 1 *Given $\epsilon > 0$, $\phi \in \mathcal{T}$, $s \in S$, $a \in A$, if $\mathcal{N}^{sa}_{\phi} > \frac{1}{\epsilon} - 1$ then for all $\Delta \in \mathcal{T}$ we have that $\frac{\sum_{s' \in S} |\phi^a_{ss'} - (\phi^a_{ss'} + \Delta^a_{ss'})|}{(\mathcal{N}^{sa}_{\phi}+1)(\mathcal{N}^{sa}_{\phi}+\mathcal{N}^{sa}_{\Delta}+1)} < \epsilon$.*

Proof According to lemma 8, we know that for all $\Delta \in \mathcal{T}$, we have that $\frac{\sum_{s' \in S} |\phi^a_{ss'} - (\phi^a_{ss'} + \Delta^a_{ss'})|}{(\mathcal{N}^{sa}_{\phi}+1)(\mathcal{N}^{sa}_{\phi}+\mathcal{N}^{sa}_{\Delta}+1)} \leq \frac{1}{\mathcal{N}^{sa}_{\phi}+1}$. Hence if $\mathcal{N}^{sa}_{\phi} > \frac{1}{\epsilon} - 1$, then $\frac{1}{\mathcal{N}^{sa}_{\phi}+1} < \epsilon$. ■

Lemma 9 *Given $\psi \in \mathcal{O}$, $s \in S$, $a \in A$, then for all $\Delta \in \mathcal{O}$, $\frac{\sum_{z \in Z} |\psi^a_{sz} - (\psi^a_{sz} + \Delta^a_{sz})|}{(\mathcal{N}^{sa}_{\psi}+1)(\mathcal{N}^{sa}_{\psi}+\mathcal{N}^{sa}_{\Delta}+1)} \leq \frac{1}{\mathcal{N}^{sa}_{\psi}+1}$.*

Proof Same proof as lemma 8. ■

Corollary 2 *Given $\epsilon > 0$, $\psi \in \mathcal{O}$, $s \in S$, $a \in A$, if $\mathcal{N}^{sa}_{\psi} > \frac{1}{\epsilon} - 1$ then for all $\Delta \in \mathcal{O}$ we have that $\frac{\sum_{z \in Z} |\psi^a_{sz} - (\psi^a_{sz} + \Delta^a_{sz})|}{(\mathcal{N}^{sa}_{\psi}+1)(\mathcal{N}^{sa}_{\psi}+\mathcal{N}^{sa}_{\Delta}+1)} < \epsilon$*

Proof Same proof as corollary 1, but using lemma 9 instead. ■

Theorem 4 *Given any $\epsilon > 0$ and $(s, \phi, \psi) \in S'$ such that $\exists a \in A, \exists s' \in S$, $\mathcal{N}^{s'a}_{\phi} > N^{\epsilon}_{S}$ or $\mathcal{N}^{s'a}_{\psi} > N^{\epsilon}_{Z}$, then $\exists (s, \phi', \psi') \in S'$ such that $\forall a \in A, \forall s' \in S$, $\mathcal{N}^{s'a}_{\phi'} \leq N^{\epsilon}_{S}$, $\mathcal{N}^{s'a}_{\psi'} \leq N^{\epsilon}_{Z}$ and $|\alpha_t(s, \phi, \psi) - \alpha_t(s, \phi', \psi')| < \epsilon$ holds for all t and $\alpha_t \in \Gamma_t$.*

Proof Consider an arbitrary $\epsilon > 0$. We first find a bound on $\mathcal{N}^{sa}_{\phi}$ and $\mathcal{N}^{sa}_{\psi}$ such that any vector with higher counts is within ϵ distance of another vector with lower counts. Let's define $\epsilon' = \frac{\epsilon(1-\gamma)^2}{8\gamma||R||_\infty}$ and $\epsilon'' = \frac{\epsilon(1-\gamma)^2 \ln(\gamma^{-e})}{32\gamma||R||_\infty}$. According to corollary 1, we have that for any $\phi \in \mathcal{T}$ such that $\mathcal{N}^{sa}_{\phi} > \frac{1}{\epsilon''} - 1$, then for all $\phi' \in \mathcal{T}$ such that there exists a $\Delta \in \mathcal{T}$ where $\phi' = \phi + \Delta$, then $\frac{\sum_{s'' \in S} |\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}^{sa}_{\phi}+1)(\mathcal{N}^{sa}_{\phi'}+1)} < \epsilon''$. Hence we want to find an N such that given $\phi \in \mathcal{T}$ with $\mathcal{N}^{sa}_{\phi} > N$, there exists a $\phi' \in \mathcal{T}$ such that $\mathcal{N}^{sa}_{\phi'} \leq N$, $D^{sa}_{S}(\phi, \phi') < \epsilon'$ and exists a $\Delta \in \mathcal{T}$ such that $\phi = \phi' + \Delta$. Let's consider an arbitrary ϕ such that $\mathcal{N}^{sa}_{\phi} > N$. We can construct a new vector ϕ' as follows, for all s' define $\phi'^a_{ss'} = \left\lfloor \frac{N\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}} \right\rfloor$ and for all other $a' \neq a, s'' \neq s$, define $\phi'^{a'}_{s''s'} = \phi^{a'}_{s''s'}$ for all s'. Clearly, $\phi' \in \mathcal{T}$, such that $N - |S| \leq \mathcal{N}^{sa}_{\phi'} \leq N$. Moreover, we have that $\phi'^{a'}_{s's''} \leq \phi^{a'}_{s's''}$ for all s', a', s'', and thus there exists a $\Delta \in \mathcal{T}$ such that $\phi = \phi' + \Delta$.

Furthermore, from its construction, we know that $\forall s'$, $\left|\frac{\phi'^a_{ss'}}{\mathcal{N}^{sa}_{\phi'}} - \frac{\phi^a_{ss'}}{\mathcal{N}^{sa}_{\phi}}\right| \leq \frac{1}{\mathcal{N}^{sa}_{\phi'}}$. Hence it is clear from this that $D^{sa}_S(\phi,\phi') \leq \frac{|S|}{N-|S|}$. Thus, if we want $D^{sa}_S(\phi,\phi') < \epsilon'$, we just need to take $N > \frac{|S|(1+\epsilon')}{\epsilon'}$. Since we also want $N > \frac{1}{\epsilon''} - 1$, let's just define $N_S = \max\left(\frac{|S|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$. $N_S = N^\epsilon_S$, as defined in Section 4, will be our bound on $\mathcal{N}^{sa}_\phi$ such that, as we have just showed, for any $\phi \in \mathcal{T}$ such that $\mathcal{N}^{sa}_\phi > N_S$, we can find a $\phi' \in \mathcal{T}$ such that $\mathcal{N}^{sa}_{\phi'} \leq N_S$, $D^{sa}_S(\phi,\phi') < \epsilon'$ and $\frac{\sum_{s''\in S}|\phi^a_{ss''} - \phi'^a_{ss''}|}{(\mathcal{N}^{sa}_\phi+1)(\mathcal{N}^{sa}_{\phi'}+1)} < \epsilon''$. Similarly, since we have a similar corollary (corollary 1) for the observation counts ψ, we can proceed in the same way and define $N_Z = \max\left(\frac{|Z|(1+\epsilon')}{\epsilon'}, \frac{1}{\epsilon''} - 1\right)$, such that for any $\psi \in \mathcal{O}$ such that $\mathcal{N}^{sa}_\psi > N_Z$, we can find a $\psi' \in \mathcal{O}$ such that $\mathcal{N}^{sa}_{\psi'} \leq N_Z$, $D^{sa}_Z(\psi,\psi') < \epsilon'$ and $\frac{\sum_{z\in Z}|\psi^a_{sz} - \psi'^a_{sz}|}{(\mathcal{N}^{sa}_\psi+1)(\mathcal{N}^{sa}_{\psi'}+1)} < \epsilon''$. $N_Z = N^\epsilon_Z$ as we have defined in Section 4.
Now let $\tilde{S} = \{(s,\phi,\psi) \in S' | \forall s' \in S, a \in A, N^{s'a}_\phi \leq N_S \ \&\ N^{s'a}_\psi \leq N_Z\}$ and consider an arbitrary $(s,\phi,\psi) \in S'$. For any $s' \in S$, $a \in A$, such that $\mathcal{N}^{s'a}_\phi > N_S$, there exists a $\phi' \in \mathcal{T}$ such that $\mathcal{N}^{s'a}_{\phi'} \leq N_S$, $D^{s'a}_S(\phi,\phi') < \epsilon'$ and $\frac{\sum_{s''\in S}|\phi^a_{s's''} - \phi'^a_{s's''}|}{(\mathcal{N}^{s'a}_\phi+1)(\mathcal{N}^{s'a}_{\phi'}+1)} < \epsilon''$ (as we have just showed above). Thus let's define $\tilde{\phi}^a_{s's''} = \phi'^a_{s's''}$ for all $s'' \in S$. For any $s' \in S$, $a \in A$, such that $\mathcal{N}^{s'a}_\phi \leq N_S$, just set $\tilde{\phi}^a_{s's''} = \phi^a_{s's''}$, $\forall s'' \in S$. Similarly, for any $s' \in S$, $a \in A$, such that $\mathcal{N}^{s'a}_\psi > N_Z$, there exists a $\psi' \in \mathcal{O}$ such that $\mathcal{N}^{s'a}_{\psi'} \leq N_Z$, $D^{s'a}_Z(\psi,\psi') < \epsilon'$ and $\frac{\sum_{z\in Z}|\psi^a_{s'z} - \psi'^a_{s'z}|}{(\mathcal{N}^{s'a}_\psi+1)(\mathcal{N}^{s'a}_{\psi'}+1)} < \epsilon''$ (as we have just showed above). Thus let's define $\tilde{\psi}^a_{s's''} = \psi'^a_{s's''}$ for all $s'' \in S$. For any $s' \in S$, $a \in A$, such that $\mathcal{N}^{s'a}_\psi \leq N_Z$, just set $\tilde{\psi}^a_{s's''} = \psi^a_{s's''}$ $\forall s'' \in S$. Now it is clear from this construction that $(s,\tilde{\phi},\tilde{\psi}) \in \tilde{S}$. By Theorem 3, for any t, $\sup_{\alpha_t\in\Gamma_t, s\in S}|\alpha_t(s,\phi,\psi) - \alpha_t(s,\tilde{\phi},\tilde{\psi})| \leq \frac{2\gamma||R||_\infty}{(1-\gamma)^2}\sup_{s,s'\in S, a\in A}\left[D^{s,a}_S(\phi,\tilde{\phi}) + D^{s',a}_Z(\psi,\tilde{\psi}) + \frac{4}{\ln(\gamma^{-e})}\left(\frac{\sum_{s''\in S}|\phi^a_{ss''} - \tilde{\phi}^a_{ss''}|}{(\mathcal{N}^{sa}_\phi+1)(\mathcal{N}^{sa}_{\tilde{\phi}}+1)} + \frac{\sum_{z\in Z}|\psi^a_{s'z} - \tilde{\psi}^a_{s'z}|}{(\mathcal{N}^{s'a}_\psi+1)(\mathcal{N}^{s'a}_{\tilde{\psi}}+1)}\right)\right] < \frac{2\gamma||R||_\infty}{(1-\gamma)^2}\left[\epsilon' + \epsilon' + \frac{4}{\ln(\gamma^{-e})}(\epsilon'' + \epsilon'')\right] = \epsilon.$ ■

Theorem 5 *Given any $\epsilon > 0$, $(s,\phi,\psi) \in S'$ and $\alpha_t \in \Gamma_t$ computed from the infinite BAPOMDP. Let $\tilde{\alpha}_t$ be the α-vector representing the same conditional plan as α_t but computed with the finite BAPOMDP $(\tilde{S}_\epsilon, A, Z, \tilde{T}_\epsilon, \tilde{O}_\epsilon, \tilde{R}_\epsilon, \gamma)$, then $|\tilde{\alpha}_t(\mathcal{P}_\epsilon(s,\phi,\psi)) - \alpha_t(s,\phi,\psi)| < \frac{\epsilon}{1-\gamma}$.*

Proof Let $(s,\phi',\psi') = \mathcal{P}_\epsilon(s,\phi,\psi)$.

$$
\begin{aligned}
&|\tilde{\alpha}_t(\mathcal{P}_\epsilon(s,\phi,\psi)) - \alpha_t(s,\phi,\psi)| \\
&\leq |\tilde{\alpha}_t(s,\phi',\psi') - \alpha_t(s,\phi',\psi')| + |\alpha_t(s,\phi',\psi') - \alpha_t(s,\phi,\psi)| \\
&< |\tilde{\alpha}_t(s,\phi',\psi') - \alpha_t(s,\phi',\psi')| + \epsilon \quad \text{(by Theorem 4)} \\
&= |\gamma\textstyle\sum_{z\in Z}\sum_{s'\in S} T^{sas'}_{\phi'} O^{s'az}_{\psi'}\left[\tilde{\alpha}'(z)(\mathcal{P}_\epsilon(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right]| + \epsilon \\
&\leq \gamma\textstyle\sum_{z\in Z}\sum_{s'\in S} T^{sas'}_{\phi'} O^{s'az}_{\psi'}\left|\tilde{\alpha}'(z)(\mathcal{P}_\epsilon(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| + \epsilon \\
&\leq \gamma\sup_{z\in Z, s'\in S}\left|\tilde{\alpha}'(z)(\mathcal{P}_\epsilon(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})) - \alpha'(z)(s',\phi'+\delta^a_{ss'},\psi'+\delta^a_{s'z})\right| + \epsilon \\
&\leq \gamma\sup_{\alpha_{t-1}\in\Gamma_{t-1},(s',\phi'',\psi'')\in S'}|\tilde{\alpha}_{t-1}(\mathcal{P}_\epsilon(s',\phi'',\psi'')) - \alpha_{t-1}(s',\phi'',\psi'')| + \epsilon.
\end{aligned}
$$

Thus, we have that:

$$\begin{aligned}&\sup_{\alpha_t\in\Gamma_t,\sigma\in S'} |\tilde{\alpha}_t(\mathcal{P}_\epsilon(\sigma)) - \alpha_t(\sigma)| \\ <\quad &\gamma \sup_{\alpha_{t-1}\in\Gamma_{t-1},\sigma'\in S'} |\tilde{\alpha}_{t-1}(\mathcal{P}_\epsilon(\sigma')) - \alpha_{t-1}(\sigma')| + \epsilon.\end{aligned}$$

This defines a recurrence. By unfolding it up to $t = 1$, where $\forall\sigma \in S'$, $\tilde{\alpha}_1(\mathcal{P}_\epsilon(\sigma)) = \alpha_1(\sigma)$, we get that $\sup_{\alpha_t\in\Gamma_t,\sigma\in S'} |\tilde{\alpha}_t(\mathcal{P}_\epsilon(\sigma)) - \alpha_t(\sigma)| < \epsilon \sum_{i=0}^{t-2} \gamma^i$. Hence for all t, this is lower than $\frac{\epsilon}{1-\gamma}$. ■

Theorem 6 *Given any $\epsilon > 0$, and any horizon t, let $\tilde{\pi}_t$ be the optimal t-step policy computed from the finite POMDP $(\tilde{S}_\epsilon, A, Z, \tilde{T}_\epsilon, \tilde{O}_\epsilon, \tilde{R}_\epsilon, \gamma)$, then for any initial belief b the value of executing policy $\tilde{\pi}_t$ in the BAPOMDP $V_{\tilde{\pi}_t}(b) \geq V^*(b) - 2\frac{\epsilon}{1-\gamma}$.*

Proof Pick any starting belief b in the BAPOMDP. Let α^* denote the optimal t-step condition plan in the BAPOMDP for b: $\alpha^* = \text{argmax}_{\alpha\in\Gamma_t} \sum_{(s,\phi,\psi)} b(s,\phi,\psi)\alpha(s,\phi,\psi)$, such that the value of this optimal conditional plan is $\sum_{(s,\phi,\psi)} b(s,\phi,\psi)\alpha^*(s,\phi,\psi) = V^*(b)$. Denote $\tilde{\alpha}^*$ the corresponding α-vector representing the same t-step conditional plan in the finite POMDP approximation.

Now let $\tilde{\alpha}' = \text{argmax}_{\tilde{\alpha}\in\tilde{\Gamma}_t} \sum_{(s,\phi,\psi)} b(s,\phi,\psi)\tilde{\alpha}(\mathcal{P}_\epsilon(s,\phi,\psi))$ be the optimal t-step conditional plan in the finite POMDP approximation if we start in belief b. This conditional plan represents exactly what the policy $\tilde{\pi}_t$ would do over t-steps starting in b. Denote α' the corresponding α-function in the BAPOMDP representing the same t-step conditional plan. Then the value of executing $\tilde{\pi}_t$ starting in b in the BAPOMDP is $V_{\tilde{\pi}_t}(b) = \sum_{(s,\phi,\psi)} b(s,\phi,\psi)\alpha'(s,\phi,\psi)$. Using Theorem 5, this value is lower bounded as follows:

$$\begin{aligned}&V_{\tilde{\pi}_t}(b) \\ =\quad &\textstyle\sum_{(s,\phi,\psi)} b(s,\phi,\psi)\alpha'(s,\phi,\psi) \\ \geq\quad &\textstyle\sum_{(s,\phi,\psi)} b(s,\phi,\psi)\tilde{\alpha}'(\mathcal{P}_\epsilon(s,\phi,\psi)) - \frac{\epsilon}{1-\gamma} \\ \geq\quad &\textstyle\sum_{(s,\phi,\psi)} b(s,\phi,\psi)\tilde{\alpha}^*(\mathcal{P}_\epsilon(s,\phi,\psi)) - \frac{\epsilon}{1-\gamma} \\ \geq\quad &\textstyle\sum_{(s,\phi,\psi)} b(s,\phi,\psi)\alpha^*(\mathcal{P}_\epsilon(s,\phi,\psi)) - 2\frac{\epsilon}{1-\gamma} \\ =\quad &V^*(b) - 2\frac{\epsilon}{1-\gamma}.\end{aligned}$$

■

References

J. Asmuth, L. Li, M. Littman, A. Nouri, and D. Wingate. A bayesian sampling approach to exploration in reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Neural Information Processing Systems (NIPS)*, volume 19, pages 49–56, 2006.

P. Auer, T. Jaksch, and R. Ortner. Near-optimal regret bounds for reinforcement learning. In *Neural Information Processing Systems (NIPS)*, volume 21, 2009.

J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research (JAIR)*, 15:319–350, 2001.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

R. I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 3: 213–231, 2003.

G. Casella and R. Berger. *Statistical Inference*. Duxbury Resource Center, 2001.

P. S. Castro and D. Precup. Using linear programming for bayesian exploration in markov decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2437–2442, 2007.

R. Dearden, N. Friedman, and S. J. Russell. Bayesian Q-learning. In *AAAI Conference on Artificial Intelligence*, pages 761–768, 1998.

R. Dearden, N. Friedman, and D. Andre. Model based bayesian exploration. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 150–159, 1999.

E. Delage and S. Mannor. Percentile optimization in uncertain mdps with application to efficient exploration. In *International Conference on Machine Learning (ICML)*, 2007.

F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. In *International Conference on Machine Learning*, pages 256–263. ACM, 2008.

F. Doshi-Velez. The infinite partially observable markov decision process. In *Neural Information Processing Systems (NIPS)*, volume 22, 2010.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods In Practice*. Springer, 2001.

M. Duff. *Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes*. PhD thesis, University of Massachusetts Amherst, Amherst, MA, 2002.

M. Duff. Monte-Carlo algorithms for the improvement of finite-state stochastic controllers: Application to bayes-adaptive Markov decision processes. In *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: The gaussian process approach to temporal difference learning. In *International Conference on Machine Learning (ICML)*, pages 154–161, 2003.

Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *International Conference on Machine learning (ICML)*, pages 201–208, 2005.

A. A. Feldbaum. Dual control theory, parts i and ii. *Automation and Remote Control*, 21: 874–880 and 1033–1039, 1961.

N. M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. In *IEEE Control Theory and Applications*, volume 147, pages 118–128, 2000.

M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithms. In *Neural Information Processing Systems (NIPS)*, volume 19, pages 457–464, 2007a.

M. Ghavamzadeh and Y. Engel. Bayesian actor-critic algorithms. In *International Conference on Machine Learning (ICML)*, pages 297–304, 2007b.

A. Greenfield and A. Brockwell. Adaptive control of nonlinear stochastic systems by particle filtering. In *International Conference on Control and Automation (ICCA)*, pages 887–890, 2003.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

M. Hutter. *Universal Artificial Intelligence*. Springer, 2005.

R. Jaulmes, J. Pineau, and D. Precup. Active learning in partially observable markov decision processes. *European Conference on Machine Learning*, pages 601–608, 2005.

E. T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4:227–241, 1968.

H. Jeffreys. *Theory of Probability*. Oxford University Press, 1961.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. In *International Conference on Machine Learning (ICML)*, pages 260–268, 1998.

M. J. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1324–1331, 1999.

J. Zico Kolter and Andrew Y. Ng. Near-bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, 2009.

M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *Neural Information Processing Systems (NIPS)*, volume 14, pages 1555–1561, 2002.

A. K. McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, University of Rochester, 1996.

S. Paquet, L. Tobin, and B. Chaib-draa. An online POMDP algorithm for complex multiagent environments. In *International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, pages 970–977, 2005.

J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: an anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032, 2003.

P. Poupart and N. Vlassis. Model-based bayesian reinforcement learning in partially observable domains. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2008.

P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *International Conference on Machine learning (ICML)*, pages 697–704, 2006.

R. Ravikanth, S.P. Meyn, and L.J. Brown. Bayesian adaptive control of time varying systems. In *IEEE Conference on Decision and Control*, pages 705–709, 1992.

S. Ross, B. Chaib-draa, and J. Pineau. Bayes-adaptive POMDPs. In *Neural Information Processing Systems (NIPS)*, volume 20, pages 1225–1232, 2008a.

S. Ross, B. Chaib-draa, and J. Pineau. Bayesian reinforcement learning in continuous POMDPs. In *International Conference on Robotics and Automation (ICRA)*, 2008b.

S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 32:663–704, 2008c.

I. Rusnak. Optimal adaptive control of uncertain stochastic discrete linear systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 4521–4526, 1995.

D. Silver and J. Veness. Monte-Carlo planning in large POMDPs. In *Neural Information Processing Systems (NIPS)*, 2010.

R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, Sep/Oct 1973.

T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 520–527, 2004.

E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University, 1971.

M. T. J. Spaan and N. Vlassis. Perseus: randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 24:195–220, 2005.

A. L. Strehl and M. L. Littman. A theoretical analysis of model-based interval estimation. In *International Conference on Machine learning (ICML)*, pages 856–863, 2005.

M. Strens. A bayesian framework for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2000.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

C. Szepesvari. *Algorithms for Reinforcement Learning*. Morgan & Claypool, 2010.

A. Tewari and P. Bartlett. Optimistic linear programming gives logarithmic regret for irreducible MDPs. In *Neural Information Processing Systems (NIPS)*, volume 20, pages 1505–1512, 2008.

J. Veness, K. S. Ng, M. Hutter, W. Uther, and D. Silver. A monte-carlo aixi approximation. *Journal of Artificial Intelligence Research (JAIR)*, 2011.

T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *International Conference on Machine learning (ICML)*, pages 956–963, 2005.

O. Zane. Discrete-time bayesian adaptive control problems with complete information. In *IEEE Conference on Decision and Control*, pages 2748–2749, 1992.