Journal of Machine Learning Research Submitted 01/2009

Multi-task Reinforcement Learning in Partially Observable Stochastic Environments

Hui Li HL1@EE.DUKE.EDU
Xuejun Liao XJLIAO@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291, USA

Abstract

We consider the problem of multi-task reinforcement learning (MTRL) in multiple partially observable stochastic environments. We introduce the regionalized policy representation (RPR) to characterize the agent's behavior in each environment. The RPR is a parametric model of the conditional distribution over current actions given the history of past actions and observations; the agent's choice of actions is directly based on this conditional distribution, without an intervening model to characterize the environment itself. We propose off-policy batch algorithms to learn the parameters of the RPRs, using episodic data collected when following a behavior policy, and show their linkage to policy iteration. We employ the Dirichlet process as a nonparametric prior over the RPRs across multiple environments. The intrinsic clustering property of the Dirichlet process imposes sharing of episodes among similar environments, which effectively reduces the number of episodes required for learning a good policy in each environment, when data sharing is appropriate. The number of distinct RPRs and the associated clusters (the sharing patterns) are automatically discovered by exploiting the episodic data as well as the nonparametric nature of the Dirichlet process. We demonstrate the effectiveness of the proposed RPR as well as the RPR-based MTRL framework on various problems, including grid-world navigation and multi-aspect target classification. The experimental results show that the RPR is a competitive reinforcement learning algorithm in partially observable domains, and the MTRL consistently achieves better performance than single task reinforcement learning.

1. Introduction

Planning in a partially observable stochastic environment has been studied extensively in the fields of operations research and artificial intelligence. Traditional methods are based on partially observable Markov decision processes (POMDPs) and assume that the POMDP models are given (Sondik 1971; Smallwood and Sondik 1973). Many POMDP planning algorithms (Sondik 1971 1978; Cheng 1988; Lovejoy 1991; Hansen 1997; Kaelbling et al. 1998; Poupart and Boutilier 2003; Pineau et al. 2003; Spaan and Vlassis 2005; Smith and Simmons 2005; Li et al. 2006ab) have been proposed, addressing problems of increasing complexity as the algorithms become progressively more efficient. However, the assumption of knowing the underlying POMDP model is often difficult to meet in practice. In many cases the only knowledge available to the agent are experiences, i.e., the observations and rewards, resulting from interactions with the environment, and the agent must learn the behavior policy based on such experience. This problem is known as reinforcement learning

(RL) (Sutton and Barto 1998). Reinforcement learning methods generally fall into two broad categories: model-based and model-free. In model-based methods, one first builds a POMDP model based on experiences and then exploits the existing planning algorithms to find the POMDP policy. In model-free methods, one directly infers the policy based on experiences. The focus of this paper is on the latter, trying to find the policy for a partially observable stochastic environment without the intervening stage of environment-model learning.

In model-based approaches, when the model is updated based on new experiences gathered from the agent-environment interaction, one has to solve a new POMDP planing problem. Solving a POMDP is computationally expensive, which is particularly true when one takes into account the model uncertainty; in the latter case the POMDP state space grows fast, often making it inefficient to find even an approximate solution (Wang et al. 2005). Recent work (Ross et al. 2008) gives a relatively efficient approximate model-based method, but still the computation time grows exponentially with the planning horizon. By contrast, model-free methods update the policy directly, without the need to update an intervening POMDP model, thus saving time and eliminating the errors introduced by approximations that may be made when solving the POMDP.

Model-based methods suffer particular computational inefficiency in multi-task reinforcement learning (MTRL), the problem being investigated in this paper, because one has to repeatedly solve multiple POMDPs due to frequent experience-updating arising from the communications among different RL tasks. The work in (Wilson et al. 2007) assumes the environment states are perfectly observable, reducing the POMDP in each task to a Markov decision process (MDP); since a MDP is relatively efficient to solve, the computational issue is not serious there. In the present paper, we assume the environment states are partially observable, thus manifesting a POMDP associated with each environment. If model-based methods are pursued, one would have to solve multiple POMDPs for each update of the task clusters, which entails a prohibitive computational burden.

Model-free methods are consequently particularly advantageous for MTRL in partially observable domains. The regionalized policy representation (RPR) proposed in this paper, which yields an efficient parametrization for the policy governing the agent's behavior in each environment, lends itself naturally to a Bayesian formulation and thus furnishes a posterior distribution of the policy. The policy posterior allows the agent to reason and plan under uncertainty about the policy itself. Since the ultimate goal of reinforcement learning is the policy, the policy's uncertainty is more direct and relevant to the learning goal than the POMDP model's uncertainty as considered in (Ross et al. 2008).

The MTRL problem considered in this paper shares similar motivations as the work in (Wilson et al. 2007) – that is, in many real-world settings there may be multiple environments for which policies are desired. For example, a single agent may have collected experiences from previous environments and wishes to borrow from previous experience when learning the policy for a new environment. In another case, multiple agents are distributed in multiple environments, and they wish to communicate with each other and share experiences such that their respective performances are enhanced. In either case the experiences in one environment should be properly exploited to benefit the learning in another (Guestrin et al. 2003). Appropriate experience sharing among multiple environments and joint learning of multiple policies save resources, improve policy quality, and enhance

generalization to new environments, especially when the experiences from each individual environment are scarce (Thrun 1996). Many problems in practice can be formulated as an MTRL problem, with one example given in (Wilson et al. 2007). The application we consider in the experiments (see Section 6.2.3) is another example, in which we make the more realistic assumption that the states of the environments are partially observable.

To date there has been much work addressing the problem of inferring the sharing structure between general learning tasks. Most of the work follows a hierarchical Bayesian approach, which assumes that the parameters (models) for each task are sampled from a common prior distribution, such as a Gaussian distribution specified by unknown hyper-parameters (Lawrence and Platt 2004; Yu et al. 2003). The parameters as well as the hyper-parameters are estimated simultaneously in the learning phase. In (Bakker and Heskes 2003) a single Gaussian prior is extended to a Gaussian mixture; each task is given a corresponding Gaussian prior and related tasks are allowed to share a common Gaussian prior. Such a formulation for information sharing is more flexible than a single common prior, but still has limitations: the form of the prior distribution must be specified *a priori*, and the number of mixture components must also be pre-specified.

In the MTRL framework developed in this paper, we adopt a nonparametric approach by employing the Dirichlet process (DP) (Ferguson 1973) as our prior, extending the work in (Yu et al. 2004; Xue et al. 2007) to model-free policy learning. The nonparametric DP prior does not assume a specific form, therefore it offers a rich representation that captures complicated sharing patterns among various tasks. A nonparametric prior drawn from the DP is almost surely discrete, and therefore a prior distribution that is drawn from a DP encourages task-dependent parameter clustering. The tasks in the same cluster share information and are learned collectively as a group. The resulting MTRL framework automatically learns the number of clusters, the members in each cluster as well as the associated common policy.

The nonparametric DP prior has been used previously in MTRL (Wilson et al. 2007), where each task is a Markov decision process (MDP) assuming perfect state observability. To the authors' knowledge, this paper represents the first attempt to apply the DP prior to reinforcement learning in multiple partially observable stochastic environments. Another distinction is that the method here is model-free, with information sharing performed directly at the policy level, without having to learn a POMDP model first; the method in (Wilson et al. 2007) is based on using MDP models.

This paper contains several technical contributions. We propose the regionalized policy representation (RPR) as an efficient parametrization of stochastic policies in the absence of a POMDP model, and develop techniques of learning the RPR parameters based on maximizing the sum of discounted rewards accrued during episodic interactions with the environment. An analysis of the techniques is provided, and relations are established to the expectation-maximization algorithm and the POMDP policy improvement theorem. We formulate the MTRL framework by placing multiple RPRs in a Bayesian setting and employ a draw from the Dirichlet process as their common nonparametric prior. The Dirichlet process posterior is derived, based on a nonconventional application of Bayes law. Because the DP posterior involves large mixtures, Gibbs sampling analysis is inefficient. This motivates a hybrid Gibbs-variational algorithm to learn the DP posterior. The proposed techniques are evaluated on four problem domains, including the benchmark Hallway2 (Littman et al.

1995), its multi-task variants, and a remote sensing application. The main theoretical results in the paper are summarized in the form of theorems and lemmas, the proofs of which are all given in the Appendix.

The RPR formulation in this paper is an extension of the work in (Li 2006; Liao et al. 2007). All other content in the paper is extended from the work in (Li 2006).

2. Partially Observable Markov Decision Processes

The partially observable Markov decision process (POMDP) (Sondik 1971; Lovejoy 1991; Kaelbling et al. 1998) is a mathematical model for the optimal control of an agent situated in a partially observable stochastic environment. In a POMDP the state dynamics of the agent are governed by a Markov process, and the state of the process is not completely observable but is inferred from observations; the observations are probabilistically related to the state. Formally, the POMDP can be described as a tuple $(\mathcal{S}, \mathcal{A}, T, \mathcal{O}, \Omega, R)$, where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$ respectively denote a finite set of states, actions, and observations; T are state-transition matrices with $T_{ss'}(a)$ the probability of transiting to state s' by taking action a in state s; Ω are observation functions with $\Omega_{s'o}(a)$ the probability of observing o after performing action a and transiting to state s'; and R is a reward function with $R(s, a)$ the expected immediate reward received by taking action a in state s.

The optimal control of a POMDP is represented by a policy for choosing the best action at any time such that the future expected reward is maximized. Since the state in a POMDP is only partially observable, the action choice is based on the belief state, a sufficient statistic defined as the probability distribution of the state s given the history of actions and observations (Sondik 1971). It is important to note that computation of the belief state requires knowing the underlying POMDP model.

The belief state constitutes a continuous-state Markov process (Smallwood and Sondik 1973). Given that at time t the belief state is b and the action a is taken, and the observation received at time $t+1$ is o, then the belief state at time $t+1$ is computed by Bayes rule

$$b_o^a(s') = \frac{\sum_{s\in\mathcal{S}} b(s) T_{ss'}^a \Omega_{s'o}^a}{p(o|b,a)} \tag{1}$$

where the superscript a and the subscript o are used to indicate the dependence of the new belief state on a and o, and

$$p(o|b,a) = \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} b(s) T_{ss'}^a \Omega_{s'o}^a \tag{2}$$

is the probability of transiting from b to b' when taking action a.

Equations (1) and (2) imply that, for any POMDP, there exists a corresponding Markov decision process (MDP), the state of which coincides with the belief state of the POMDP (hence the term "belief-state MDP"). Although the belief state is continuous, their transition probabilities are discrete : from any given b, one can only make a transition to a finite number of new belief states $\{b_o^a : a \in \mathcal{A}, o \in \mathcal{O}\}$, assuming $\mathcal{A}$ and $\mathcal{O}$ are discrete sets with finite alphabets. For any action $a \in \mathcal{A}$, the belief state transition probabilities are given by

$$p(b'|b,a) = \begin{cases} p(o|b,a), & \text{if } b' = b_o^a \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The expected reward of the belief-state MDP is given by

$$R(b,a) = \sum_{s\in\mathcal{S}} b(s)R(s,a) \tag{4}$$

In summary, the belief-state MDP is completely defined by the action set $\mathcal{A}$, the space of belief state

$$\mathcal{B} = \left\{ b \in \mathbb{R}^{|\mathcal{S}|} : b(s) \geq 0, \sum_{s\in\mathcal{S}} b(s) = 1 \right\}$$

along with the belief state transition probabilities in (3) and the reward function in (4).

The optimal control of the POMDP can be found by solving the corresponding belief-state MDP. Assume that at any time there are infinite steps remaining for the POMDP (infinite horizon), the future rewards are discounted exponentially with a factor $0 < \gamma < 1$, and the action is drawn from $p^{\Pi}(a|b)$, then the expected reward accumulated over the infinite horizon satisfies the Bellman equation (Bellman 1957; Smallwood and Sondik 1973)

$$V^{\Pi}(b) = \sum_{a\in\mathcal{A}} p^{\Pi}(a|b) \left[R(b,a) + \gamma \sum_{o\in\mathcal{O}} p(o|b,a) V^{\Pi}(b_o^a) \right] \tag{5}$$

where $V^{\Pi}(b)$ is called the value function. Sondik (1978) showed that, for a finite-transient deterministic policy [1], there exists a Markov partition $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots$ satisfying the following two properties :

(a) There is a unique optimal action a_i associated with subset $\mathcal{B}_i$, $i = 1, 2, \cdots$. This implies that the optimal control is represented by a deterministic mapping from the Markov partition to the set of actions.

(b) Each subset maps completely into another (or itself), i.e., $\{b_o^a : b \in \mathcal{B}_i, a = \Pi(b), o \in \mathcal{O}\} \subseteq \mathcal{B}_j$ (i may equal j).

The Markov partition yields an equivalent representation of the finite-transient deterministic policy. Sondik noted that an arbitrary policy Π is not likely to be finite-transient, and for it one can only construct a partition where one subset maps partially into another (or itself), i.e., there exists $b \in \mathcal{B}_i$ and $o \in \mathcal{O}$ such that $b_o^{\Pi(b)} \notin \mathcal{B}_j$. Nevertheless, the Markov partition provides an approximate representation for non-finite-transient policies and Sondik gave an error bound of the difference between the true value function and approximate value function obtained by the Markov partition. Based on the Markov partition, Sondik also proposed a policy iteration algorithm for POMDPs, which was later improved by Hansen (1997) and the improved algorithm is referred to as finite state controller (the partition is finite).

1. Let Π be a deterministic policy, i.e., $p^{\Pi}(a|b) = \begin{cases} 1, & \text{if } a = \Pi(b) \\ 0, & \text{otherwise} \end{cases}$. Let S_{Π}^n be the set of all possible belief-states when Π has been followed for n consecutive steps by starting from any initial belief-state. The Π is finite transient if and only if there exists $n < \infty$ such that S_{Π}^n is disjoint with $\{b : \Pi(b) \text{ is discontinuous at } b\}$ (Sondik 1978).

3. Regionalized Policy Representation

We are interested in model-free policy learning, i.e., we assume the model of the POMDP is *unknown* and aim to learn the policy directly from the experiences (data) collected from agent-environment interactions. One may argue that we do in fact learn a model, but our model is directly at the policy level, constituting a *probabilistic* mapping from the space of action-observation histories to the action space.

Although the optimal control of a POMDP can be obtained via solving the corresponding belief-state MDP, this is not true when we lack an underlying POMDP model. This is because, as indicated above, the observability of the belief-state depends on the availability of the POMDP model. When the model is unknown, one does not have access to the information required to compute the belief state, making the belief state *unobservable*.

In this paper, we treat the belief-state as a hidden (latent) variable and marginalize it out to yield a stochastic POMDP policy that is purely dependent on the observable history, i.e., the sequence of previous actions and observations. The belief-state dynamics, as well as the optimal control in each state, is learned empirically from experiences, instead of being computed from an underlying POMDP model. Although it may be possible to learn the dynamics and control in the continuous space of belief state, the exposition in this paper is restricted to the discrete case, i.e., the case for which the continuous belief-state space is quantized into a finite set of disjoint regions. The quantization can be viewed as a stochastic counterpart of the Markov partition (Sondik 1978), discussed at the end of Section 2. With the quantization, we learn the dynamics of belief regions and the local optimal control in each region, both represented stochastically. The stochasticity manifests the uncertainty arising from the belief quantization (the policy is parameterized in terms of latent belief *regions*, not the precise belief state). The stochastic policy reduces to a deterministic one when the policy is finitely transient, in which case the quantization becomes a Markov partition. The resulting framework is termed *regionalized policy representation* to reflect the fact that the policy of action selection is expressed through the dynamics of belief regions as well as the local controls in each region. We also use *decision state* as a synonym of *belief region*, in recognition of the fact that each belief region is an elementary unit to encode the decisions of action selection.

3.1 Formal Framework

Definition 1. A *regionalized policy representation* (RPR) is a tuple $\langle \mathcal{A}, \mathcal{O}, \mathcal{Z}, W, \mu, \pi \rangle$ specified as follows. The $\mathcal{A}$ and $\mathcal{O}$ are respectively a finite set of actions and observations. The $\mathcal{Z}$ is a finite set of decision states (belief regions). The W are decision-state transition matrices with $W(z, a, o', z')$ denoting the probability of transiting from z to z' when taking action a in decision state z results in observing o'. The μ is the initial distribution of decision states with $\mu(z)$ denoting the probability of initially being in decision state z. The π are state-dependent *stochastic* policies with $\pi(z, a)$ denoting the probability of taking action a in decision state z.

The stochastic formulation of W and π in Definition 1 is fairly general and subsumes two special cases.

1. If z shrinks down to a single belief-state b, $z = b$ becomes a sufficient statistic of the POMDP (Smallwood and Sondik 1973) and there is a unique action associated with it, thus $\pi(z, a)$ is deterministic and the local policy can be simplified as $a = \pi(b)$.

2. If the belief regions form a Markov partition of the belief-state space (Sondik 1978), i.e., $\mathcal{B} = \cup_{z\in\mathcal{Z}}\mathcal{B}_z$, then the action choice in each region is constant and one region transits completely to another (or itself). In this case, both W and π are deterministic and, moreover, the policy yielded by the RPR (see (9)) is finite transient deterministic. In fact this is the same case as considered in (Hansen 1997).

In both of the two special cases, each z has one action choice $a = \pi(z)$ associated with it, and one can write $W(z, a, o', z') = W(z, \pi(z), o', z')$, thus the transition of z is driven solely by o. In general, each z represents multiple individual belief-states, and the belief region transition is driven jointly by a and o. The action-dependency captures the state dynamics of the POMDP, and the observation-dependency reflects the partial observability of the state (perception aliasing).

To make notation simple, the following conventions are observed throughout the paper:

- The elements of $\mathcal{A}$ are enumerated as $\mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$, where $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. Similarly, $\mathcal{O} = \{1, 2, \cdots, |\mathcal{O}|\}$ and $\mathcal{Z} = \{1, 2, \cdots, |\mathcal{Z}|\}$.

- A sequence of actions $(a_0, a_1, \cdots, a_T)$ is abbreviated as $a_{0:T}$, where the subscripts index discrete time steps. Similarly a sequence of observations $(o_1, o_2, \cdots, o_T)$ is abbreviated as $o_{1:T}$, and a sequence of decision states $(z_0, z_1, \cdots, z_T)$ is abbreviated as $z_{0:T}$, etc.

- A history h_t is the set of actions executed and observation received up to time step t, i.e., $h_t = \{a_{0:t-1}, o_{1:t}\}$.

Let $\Theta = \{\pi, \mu, W\}$ denote the parameters of the RPR. Given a history of actions and observations, $h_t = (a_{0:t-1}, o_{1:t})$, collected up to time step t, the RPR yields a joint probability distribution of $z_{0:t}$ and $a_{0:t}$

$$p(a_{0:t}, z_{0:t}|o_{1:t}, \Theta) = \mu(z_0)\pi(z_0, a_0)\prod_{\tau=1}^{t} W(z_{\tau-1}, a_{\tau-1}, o_\tau, z_\tau)\pi(z_\tau, a_\tau) \tag{6}$$

where application of local controls $\pi(z_t, a_t)$ at every time step implies that $a_{0:t}$ are all drawn according to the RPR. The decision states $z_{0:t}$ in (6) are hidden variables and we marginalize them to get

$$p(a_{0:t}|o_{1:t}, \Theta) = \sum_{z_0,\cdots,z_t=1}^{|\mathcal{Z}|}\left[\mu(z_0)\pi(z_0, a_0)\prod_{\tau=1}^{t} W(z_{\tau-1}, a_{\tau-1}, o_\tau, z_\tau)\pi(z_\tau, a_\tau)\right] \tag{7}$$

It follows from (7) that

$$p(a_{0:t-1}|o_{1:t}, \Theta) = \sum_{a_t=1}^{|\mathcal{A}|} p(a_{0:t}|o_{1:t}, \Theta)$$

$$
\begin{aligned}
&= \sum_{z_0,\cdots,z_{t-1}=1}^{|\mathcal{Z}|} \left[\mu(z_0)\pi(z_0,a_0) \prod_{\tau=1}^{t-1} W(z_{\tau-1},a_{\tau-1},o_\tau,z_\tau)\pi(z_\tau,a_\tau) \right] \\
&\quad \times \underbrace{\sum_{a_t=1}^{|\mathcal{A}|} \sum_{z_t=1}^{|\mathcal{Z}|} W(z_{t-1},a_{t-1},o_t,z_t)\pi(z_t,a_t)}_{=1} \\
&= p(a_{0:t-1}|o_{1:t-1},\Theta) \qquad (8)
\end{aligned}
$$

which implies that observation o_t does not influence the actions before t, in agreement with expectations. From (7) and (8), we can write the history-dependent distribution of action choices

$$
p(a_\tau|h_\tau,\Theta) = p(a_\tau|a_{0:\tau-1},o_{1:\tau},\Theta) = \frac{p(a_{0:\tau}|o_{1:\tau},\Theta)}{p(a_{0:\tau-1}|o_{1:\tau},\Theta)} = \frac{p(a_{0:\tau}|o_{1:\tau},\Theta)}{p(a_{0:\tau-1}|o_{1:\tau-1},\Theta)} \qquad (9)
$$

which gives a stochastic RPR policy for choosing the action a_t, given the historical actions and observations. The policy is purely history-dependent, with the unobservable belief regions z integrated out.

The history h_t forms a Markov process with transitions driven by actions and observations: $h_t = h_{t-1} \cup \{a_{t-1}, o_t\}$. Applying this recursively, we get $h_t = \cup_{\tau=1}^{t}\{a_{\tau-1}, o_\tau\}$, and therefore

$$
\begin{aligned}
\prod_{\tau=0}^{t} p(a_\tau|h_\tau,\Theta) &= \left[\prod_{\tau=0}^{t-2} p(a_\tau|h_\tau,\Theta) \right] p(a_{t-1}|h_{t-1},\Theta) p(a_t|h_{t-1},a_{t-1},o_t,\Theta) \\
&= \left[\prod_{\tau=0}^{t-2} p(a_\tau|h_\tau,\Theta) \right] p(a_{t-1:t}|h_{t-1},o_t,\Theta) \\
&= \left[\prod_{\tau=0}^{t-3} p(a_\tau|h_\tau,\Theta) \right] p(a_{t-2}|h_{t-2},\Theta) p(a_{t-1:t}|h_{t-2},a_{t-2},o_{t-1},o_t,\Theta) \\
&= \left[\prod_{\tau=0}^{t-3} p(a_\tau|h_\tau,\Theta) \right] p(a_{t-2:t}|h_{t-2},o_{t-1:t},\Theta) \\
&\vdots \\
&= p(a_{0:t}|h_0,o_{1:t},\Theta) \\
&= p(a_{0:t}|o_{1:t},\Theta) \qquad (10)
\end{aligned}
$$

where we have used $p(a_\tau|h_\tau, o_{\tau+1:t}) = p(a_\tau|h_\tau)$ and $h_0 = \text{null}$. The rightmost side of (10) is the observation-conditional probability of joint action-selection at multiple time steps $\tau = 0, 1, \cdots, t$. Equation (10) can be verified directly by multiplying (9) over $\tau = 0, 1, \cdots, t$

$$
\begin{aligned}
&\prod_{\tau=0}^{t} p(a_\tau|h_\tau,\Theta) \\
&= p(a_0|\Theta) \frac{p(a_{0:1}|o_1,\Theta)}{p(a_0|\Theta)} \frac{p(a_{0:2}|o_{1:2},\Theta)}{p(a_{0:1}|o_1,\Theta)} \cdots \frac{p(a_{0:t-1}|o_{1:t-1},\Theta)}{p(a_{0:t-2}|o_{1:t-2},\Theta)} \frac{p(a_{0:t}|o_{1:t},\Theta)}{p(a_{0:t-1}|o_{1:t-1},\Theta)} \\
&= p(a_{0:t}|o_{1:t},\Theta) \qquad (11)
\end{aligned}
$$

It is of interest to point out the difference between the RPR and previous reinforcement learning algorithms for POMDPs. The reactive policy and history truncation (Jaakkola et al. 1995; Baxter and Bartlett 2001) condition the action only upon the immediate observation or a truncated sequence of observations, without using the full history, and therefore these are clearly different from the RPR. The U-tree (McCallum 1995) stores historical information along the branches of decision trees, with the branches split to improve the prediction of future return or utility. The drawback is that the tree may grow intolerably fast with the episode length. The finite policy graphs (Meuleau et al. 1999), finite state controllers (Aberdeen and Baxter 2002), and utile distinction HMMs (Wierstra and Wiering 2004) use internal states to memorize the full history, however, their state transitions are driven by observations only. In contrast, the dynamics of decision states in the RPR are driven jointly by actions and observations, the former capturing the dynamics of world-states and the latter reflecting the perceptual aliasing. Moreover, none of the previous algorithms is based on Bayesian learning, and therefore they are intrinsically not amenable to the Dirichlet process framework that is used in the RPR for multi-task examples.

3.2 The Learning Objective

We are interested in empirical learning of the RPR, based on a set of episodes defined as follows.

Definition 2. *(Episode)* An episode is a sequence of agent-environment interactions terminated in an absorbing state that transits to itself with zero rewards (Sutton and Barto 1998). An episode is denoted by $(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)$, where the subscripts are discrete times, k indexes the episodes, and o, a, and r are respectively observations, actions, and immediate rewards.

Definition 3. *(Sub-episode)* A sub-episode is an episode truncated at a particular time step and retaining the immediate reward only at the time step where truncation occurs. The t-th sub-episode of episode $(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)$ is defined as $(a_0^k o_1^k a_1^k \cdots o_t^k a_t^k r_t^k)$, which yields a total of $T_k + 1$ sub-episodes for this episode.

The learning objective is to maximize the optimality criterion given in Definition 4. Theorem 5 introduced below establishes the limit of the criterion when the number of episodes approaches infinity.

Definition 4. *(The RPR Optimality Criterion)* Let $\mathcal{D}^{(K)} = \{(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)\}_{k=1}^K$ be a set of episodes obtained by an agent interacting with the environment by following policy Π to select actions, where Π is an arbitrary stochastic policy with action-selecting distributions $p^{\Pi}(a_t|h_t) > 0$, $\forall$ action a_t, $\forall$ history h_t. The RPR optimality criterion is defined as

$$\widehat{V}(\mathcal{D}^{(K)};\Theta) \stackrel{def.}{=} \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\gamma^t r_t^k}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau^k|h_\tau^k)}\prod_{\tau=0}^{t} p(a_\tau^k|h_\tau^k,\Theta) \tag{12}$$

where $h_t^k = a_0^k o_1^k a_1^k \cdots o_t^k$ is the history of actions and observations up to time t in the k-th episode, $0 < \gamma < 1$ is the discount, and Θ denotes the parameters of the RPR.

Theorem 5. *Let $\widehat{V}(\mathcal{D}^{(K)};\Theta)$ be as defined in Definition 4, then $\lim_{K\rightarrow\infty}\widehat{V}(\mathcal{D}^{(K)};\Theta)$ is the expected sum of discounted rewards within the environment under test by following the RPR policy parameterized by Θ, over an infinite horizon.*

Theorem 5 shows that the optimality criterion given in Definition 4 is the expected sum of discounted rewards in the limit, when the number of episodes approaches infinity. Throughout the paper, we call $\lim_{K\rightarrow\infty}\widehat{V}(\mathcal{D}^{(K)};\Theta)$ the value function and $\widehat{V}(\mathcal{D}^{(K)};\Theta)$ the empirical value function. The Θ maximizing the (empirical) value function is the best RPR policy (given the episodes).

It is assumed in Theorem 5 that the behavior policy Π used to collect the episodic data is an arbitrary policy that assigns nonzero probability to any action given any history, i.e., Π is required to be a soft policy (Sutton and Barto 1998). This premise assures a complete exploration of the actions that might lead to large immediate rewards given any history, i.e., the actions that might be selected by the optimal policy.

4. Single-Task Reinforcement Learning (STRL)

We develop techniques to maximize the empirical value function in (12) and the Θ resulting from value maximization is called a Maximum-Value (MV) estimate (related to maximum *likelihood*). An MV estimate of the RPR is preferred when the number of episodes is large, in which case the empirical value function approaches the true value function and the estimate is expected to approach the optimal (assuming the algorithm is not trapped in a local minima). The episodes $\mathcal{D}^{(K)}$ are assumed to have been collected in a single partially observable stochastic environment, which may corresponds to a single physical environment or a pool of multiple identical/similar physical environments. As a result, the techniques developed in this section are for single-task reinforcement learning (STRL).

By substituting (7) and (10) into (12), we rewrite the empirical value function,

$$\widehat{V}(\mathcal{D}^{(K)};\Theta)=\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\tilde{r}_t^k\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}p(a_{0:t}^k,z_{0:t}^k|o_{1:t}^k,\Theta) \tag{13}$$

where

$$\tilde{r}_t^k=\frac{\gamma^t r_t^k}{\prod_{\tau=0}^{t}p^{\Pi}(a_\tau^k|h_\tau^k)} \tag{14}$$

is the discounted immediate reward $\gamma^t r_t^k$ weighted by the inverse probability that the behavior policy Π has generated r_t^k. The weighting is a result from importance sampling (Robert and Casella 1999), and reflects the fact that r_t^k is obtained by following Π but the Monte Carlo integral (i.e., the empirical value function) is with respect to the RPR policy Θ. For simplicity, $\tilde{r}_t^k$ is also referred to as discounted immediate reward or simply reward throughout the paper.

We assume $r_t\geq 0$ (and hence $\widetilde{r}_t\geq 0$), which can always be achieved by adding a constant to r_t; this results in a constant added to the value function (the value function of a POMDP is linear in immediate reward) and does not influence the policy.

Theorem 6. *(Maximum Value Estimation) Let*

$$q_t^k(z_{0:t}^k|\Theta^{(n)}) = \frac{\tilde{r}_t^k}{\widehat{V}(\mathcal{D}^{(K)};\Theta^{(n)})} p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \Theta^{(n)}) \tag{15}$$

for $z_t^k = 1, 2, \cdots, |\mathcal{Z}|$, $t = 1, 2, \cdots, T_k$, *and* $k = 1, 2, \cdots, K$. *Let*

$$\Theta^{(n+1)} = \arg\max_{\widehat{\Theta}\in\mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k|\Theta^{(n)}) \ln \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widehat{\Theta})}{q_t^k(z_{0:t}^k|\Theta^{(n)})} \tag{16}$$

where

$$\mathcal{F} = \Bigg\{ \Theta = (\mu, \pi, W) : \sum_{j=1}^{|\mathcal{Z}|} \widehat{\mu}(j) = 1, \sum_{a=1}^{|\mathcal{A}|} \widehat{\pi}(i,a) = 1, \sum_{j=1}^{|\mathcal{Z}|} \widehat{W}(i,a,o,j) = 1, \\ i = 1, 2, \cdots, |\mathcal{Z}|,\ a = 1, 2, \cdots, |\mathcal{A}|,\ o = 1, 2, \cdots, |\mathcal{O}| \Bigg\} \tag{17}$$

is the set of feasible parameters for the RPR in question. Let $\{\Theta^{(0)}\Theta^{(1)}\cdots\Theta^{(n)}\cdots\}$ *be a sequence yielded by iteratively applying (15) and (16), starting from* $\Theta^{(0)}$. *Then*

$$\lim_{n\to\infty} \widehat{V}(\mathcal{D}^{(K)};\Theta^{(n)})$$

exists and the limit is a maxima of $\widehat{V}(\mathcal{D}^{(K)};\Theta)$.

To gain a better understanding of Theorem 6, we rewrite (15) to get

$$q_t^k(z_{0:t}^k|\Theta) \quad = \quad \frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)} p(z_{0:t}^k|a_{0:t}^k, o_{1:t}^k, \Theta) \tag{18}$$

where $p(z_{0:t}^k|a_{0:t}^k, o_{1:t}^k, \Theta)$ is an standard posterior distribution of the latent decision states given the Θ updated in the most recent iteration (the superscript $^{(n)}$ indicating the iteration number has been dropped for simplicity), and

$$\sigma_t^k(\Theta) \stackrel{Def.}{=} \tilde{r}_t^k p(a_{0:t}^k|o_{1:t}^k, \Theta) \tag{19}$$

is called the *re-computed reward* at time step t in the k-th episode. The re-computed reward represents the discounted immediate reward $\tilde{r}_t^k$ *weighted* by the probability that the action sequence yielding this reward is generated by the RPR policy parameterized by Θ, therefore $\sigma_t^k(\Theta)$ is a function of Θ. The re-computed reward reflects the update of the RPR policy which, if allowed to re-interact with the environment, is expected to accrue larger rewards than in the previous iteration. Recall that the algorithm does not assume real re-interactions with the environment so the episodes themselves cannot update. However, by recomputing the rewards as in (19), the agent is allowed to generate an *internal* set of episodes in which the immediate rewards are modified. The internal episodes represent the *new* episodes that would be collected if the agent followed the updated RPR to *really*

re-interact with the environment. In this sense, the reward re-computation can be thought of as virtual re-interactions with the environment.

By (18), $q_t^k(z_{0:t}^k)$ is a weighted version of the standard posterior of $z_{0:t}^k$, with the weight given by the reward recomputed by the RPR in the previous iteration. The normalization constant $\widehat{V}(\mathcal{D}^{(K)};\Theta)$, which is also the empirical value function in (12), can be expressed as the recomputed rewards averaged over all episodes at all time steps,

$$\widehat{V}(\mathcal{D}^{(K)};\Theta) = \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k(\Theta) \tag{20}$$

which ensures

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k|\Theta) = 1 \tag{21}$$

The maximum value (MV) algorithm based on alternately applying (15) and (16) in Theorem 6 bears strong resemblance to the expectation-maximization (EM) algorithms (Dempster et al. 1977) widely used in statistics, with (15) and (16) respectively corresponding to the E-step and M-step in EM. However, the goal in standard EM algorithms is to maximize a likelihood function, while the goal of the MV algorithm is to maximize an empirical value function. This causes significant differences between the MV and the EM. It is helpful to compare the MV algorithm in Theorem 6 to the EM algorithm for maximum likelihood (ML) estimation in hidden Markov models (Rabiner 1989), since both deal with sequences or episodes. The sequences in an HMM are treated as uniformly important, therefore parameter updating is based solely on the frequency of occurrences of latent states. Here the episodes are not equally important because they have different rewards associated with them, which determine their importance relative to each other. As seen in (18), the posterior of $z_{0:t}^k$ is weighted by the recomputed reward σ_t^k, which means that the contribution of episode k (at time t) to the update of Θ is not solely based on the frequency of occurrences of $z_{0:t}^k$ but also based on the associated σ_t^k. Thus the new parameters $\widehat{\Theta}$ will be adjusted in such a way that the episodes earning large rewards have more "credits" recorded into $\widehat{\Theta}$ and, as a result, the policy parameterized by $\widehat{\Theta}$ will more likely generate actions that lead to high rewards.

The objective function being maximized in (16) enjoys some interesting properties due to the fact that $q_t^k(z_{0:t}^k)$ is a weighted posterior of $z_{0:t}^k$. These properties not only establish a more formal connection between the MV algorithm here and the traditional ML algorithm based on EM, they also shed light on the close relations between Theorem 6 and the policy improvement theorem of POMDP (Blackwell 1965). To show these properties, we rewrite the objective function in (16) (with the subscript $^{(n)}$ dropped for simplicity) as

$$\begin{aligned}
\mathrm{LB}(\widehat{\Theta}|\Theta) &\overset{Def.}{=} \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k|\Theta)\ln\frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k,\widehat{\Theta})}{q_t^k(z_{0:t}^k|\Theta)} \\
&= \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} p(z_{0:t}^k|a_{0:t}^k,o_{1:t}^k,\Theta)\ln\frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k,\widehat{\Theta})}{\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}p(z_{0:t}^k|a_{0:t}^k,o_{1:t}^k,\Theta)}
\end{aligned} \tag{22}$$

where the second equation is obtained by substituting (18) into the left side of it. Since $\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}=1$ and $\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}p(z_{0:t}^k|a_{0:t}^k,o_{1:t}^k,\Theta)=1$, one can apply Jensen's inequality *twice* to the rightmost side of (22) to obtain two inequalities

$$\begin{aligned}\mathrm{LB}(\widehat{\Theta}|\Theta) &\leq \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}\ln\frac{\tilde{r}_t^k p(a_{0:t}^k|o_{1:t}^k,\widehat{\Theta})}{\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}} \stackrel{Def.}{=} \Upsilon(\widehat{\Theta}|\Theta)\\ &\leq \ln\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\tilde{r}_t^k p(a_{0:t}^k|o_{1:t}^k,\widehat{\Theta})\right] = \ln\widehat{V}(\mathcal{D}^{(K)};\widehat{\Theta}) \end{aligned} \tag{23}$$

where the first inequality is with respect to $p(z_{0:t}^k|a_{0:t}^k,o_{1:t}^k,\Theta)$ while the second inequality is with respect to $\left\{\frac{\sigma_t^k(\Theta)}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}:t=1,\cdots,T_k,k=1,\cdots,K\right\}$. Each inequality yields a lower bound to the logarithmic empirical value function $\ln\widehat{V}(\mathcal{D}^{(K)};\widehat{\Theta})$. It is not difficult to verify from (22) and (23) that both of the two lower bounds are tight (the respective equality can be reached), i.e.,

$$\mathrm{LB}(\Theta|\Theta)=\ln\widehat{V}(\mathcal{D}^{(K)};\Theta)=\Upsilon(\Theta|\Theta) \tag{24}$$

The equations in (24) along with the inequalities in (23) show that any $\widehat{\Theta}$ satisfying $\mathrm{LB}(\Theta|\Theta)<\mathrm{LB}(\widehat{\Theta}|\Theta)$ or $\Upsilon(\Theta|\Theta)<\Upsilon(\widehat{\Theta}|\Theta)$ also satisfies $\widehat{V}(\mathcal{D}^{(K)};\Theta)<\widehat{V}(\mathcal{D}^{(K)};\widehat{\Theta})$. Thus one can choose to maximize either of the two lower bounds, $\mathrm{LB}(\widehat{\Theta}|\Theta)$ or $\Upsilon(\widehat{\Theta}|\Theta)$, when trying to improve the empirical value of $\widehat{\Theta}$ over that of Θ. In either case, the maximization is with respect to $\widehat{\Theta}$.

The two alternatives, though both yielding an improved RPR, are quite different in the manner the improvement is achieved. Suppose one has obtained $\Theta^{(n)}$ by applying (15) and (16) for n iterations, and is seeking $\Theta^{(n+1)}$ satisfying $\widehat{V}(\mathcal{D}^{(K)};\Theta^{(n)})<\widehat{V}(\mathcal{D}^{(K)};\Theta^{(n+1)})$. Maximization of the first lower bound gives $\Theta^{(n+1)}=\arg\max_{\widehat{\Theta}\in\mathcal{F}}\mathrm{LB}(\widehat{\Theta}|\Theta^{(n)})$, which has an analytic solution that will be given in Section 4.2. Maximization of the second lower bound yields

$$\Theta^{(n+1)}=\arg\max_{\widehat{\Theta}\in\mathcal{F}}\Upsilon(\widehat{\Theta}|\Theta^{(n)}) \tag{25}$$

The definition of Υ in (23) is substituted into (25) to yield

$$\begin{aligned}\Theta^{(n+1)} &= \arg\max_{\widehat{\Theta}\in\mathcal{F}}\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\sigma_t^k(\Theta^{(n)})}{\widehat{V}(\mathcal{D}^{(K)};\Theta^{(n)})}\ln\frac{\tilde{r}_t^k p(a_{0:t}^k|o_{1:t}^k,\widehat{\Theta})}{\frac{\sigma_t^k(\Theta^{(n)})}{\widehat{V}(\mathcal{D}^{(K)};\Theta^{(n)})}}\\ &= \arg\max_{\widehat{\Theta}\in\mathcal{F}}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k(\Theta^{(n)})\ln p(a_{0:t}^k|o_{1:t}^k,\widehat{\Theta}) \end{aligned} \tag{26}$$

which shows that maximization of the second lower bound is equivalent to maximizing a weighted sum of the log-likelihoods of $\{a_{0:t}^k\}$, with the weights being the rewards recomputed by $\Theta^{(n)}$. Through (26), the connection between the maximum value algorithm in Theorem 6 and the traditional ML algorithm is made more formal and clearer: with the recomputed

rewards given and fixed, the MV algorithm is a weighted version of the ML algorithm, with $\Upsilon(\widehat{\Theta}|\Theta^{(n)})$ a weighted log-likelihood function of $\widehat{\Theta}$.

The above analysis also sheds light on the relations between Theorem 6 and the policy improvement theorem in POMDP (Blackwell 1965). By (23), (24), and (26), we have

$$\begin{aligned} \ln V(\mathcal{D}^{(K)};\Theta^{(n)}) = \Upsilon(\Theta^{(n)}|\Theta^{(n)}) &\leq \Upsilon(\Theta^{(n+1)}|\Theta^{(n)}) \\ &\leq \ln V(\mathcal{D}^{(K)};\Theta^{(n+1)}) \end{aligned} \tag{27}$$

The first inequality, achieved by the weighted likelihood maximization in (26), represents the policy improvement on the old episodes collected by following the previous policy. The second inequality ensures that, if the improved policy is followed to collect new episodes in the environment, the expected sum of newly accrued rewards is no less than that obtained by following the previous policy. This is similar to policy evaluation. Note that the update of episodes is simulated by reward computation. The actual episodes are collected by a fixed behavior policy Π and do not change.

The maximization in (26) can be performed using any optimization techniques. As long as the maximization is achieved, the policy is improved as guaranteed by Theorem 6. Since the latent z variables are involved, it is natural to employ EM to solve the maximization. The EM solution to (26) is obtained by solving a sequence of maximization problems: starting from $\Theta^{(n)(0)} = \Theta^{(n)}$, one successively solves

$$\Theta^{(n)(j)} = \arg\max_{\widehat{\Theta}\in\mathcal{F}} \mathrm{LB}(\widehat{\Theta}|\Theta^{(n)(j-1)}) \text{ subject to } \sigma_t^k(\Theta^{(n)(j-1)}) = \sigma_t^k(\Theta^{(n)}) \ \forall\, t,k \tag{28}$$
$$j = 1, 2, \cdots$$

where in each problem one maximizes the first lower bound with an updated posterior of $\{z_t^k\}$ but with the recomputed rewards fixed at $\{\sigma_t^k(\Theta^{(n)})\}$; upon convergence, the solution of (28) is the solution to (26). The EM solution here is almost the same as the likelihood maximization of sequences for hidden Markov models (Rabiner 1989). The only difference is that here we have a weighted log-likelihood function, but with the weights given and fixed. The posterior of $\{z_t^k\}$ can be updated by employing the dynamical programming techniques similar to those used in HMM, as we discuss below.

It is interesting to note that, with standard EM employed to solve (26), the overall maximum value algorithm is a "double-EM" algorithm, since reward computation constitutes an outer EM-like loop.

4.1 Calculating the Posterior of Latent Belief Regions

To allocate the weights or recomputed rewards and update the RPR as in (16), we do not need to know the full distribution of $z_{0:t}^k$. Instead, a small set of marginals of $p(z_{0:t}^k|a_{0:t}^k, o_{1:t}^k, \Theta)$ are necessary for the purpose, in particular,

$$\xi_{t,\tau}^k(i,j) = p(z_\tau^k = i, z_{\tau+1}^k = j|a_{0:t}^k, o_{1:t}^k, \Theta) \tag{29}$$
$$\phi_{t,\tau}^k(i) = p(z_\tau^k = i|a_{0:t}^k, o_{1:t}^k, \Theta) \tag{30}$$

Lemma 7. *(Factorization of the ξ and ϕ Variables) Let*

$$\alpha_\tau^k(i) = p(z_\tau^k = i|a_{0:\tau}^k, o_{1:\tau}^k, \Theta)$$

$$= \frac{p(z^k_\tau = i, a^k_{0:\tau}|o^k_{1:\tau}, \Theta)}{\prod_{\tau'=0}^{\tau} p(a^k_{\tau'}|h^k_{\tau'}, \Theta)} \tag{31}$$

$$\beta^k_{t,\tau}(i) = \frac{p(a^k_{\tau+1:t}|z^k_\tau = i, o^k_{\tau+1:t}, \Theta)}{\prod_{\tau'=\tau}^{t} p(a^k_{\tau'}|h^k_{\tau'}, \Theta)} \tag{32}$$

Then

$$\xi^k_{t,\tau}(i,j) = \alpha^k_\tau(i) W(z^k_\tau = i, a^k_\tau, o^k_{\tau+1}, z^k_{\tau+1} = j)\pi(a^k_{\tau+1}|z^k_{\tau+1} = j)\beta^k_{t,\tau+1}(j) \tag{33}$$

$$\phi^k_{t,\tau}(i) = \alpha^k_\tau(i)\beta^k_{t,\tau}(i)p(a^k_\tau|h^k_\tau) \tag{34}$$

The α and β variables in the Lemma 7 are similar to the scaled forward variables and backward variables in hidden Markov models (HMM) (Rabiner 1989). The scaling factors here are $\prod_{\tau'=0}^{\tau} p(a^k_{\tau'}|h^k_{\tau'}, \Theta)$, which is equal to $p(a^k_{0:\tau}|o^k_{1:\tau}, \Theta)$ as shown in (10) and (11). Recall from Definition 3 that one episode of length T has $T+1$ sub-episodes with each having a different ending time step. For this reason, one must compute the β variables for each sub-episode separately, since the β variables depend on the ending time step. For α variables, one needs to compute them once per episode, since it does not involve the ending time step.

Similar to the forward variables and backward variables in HMM models, the α and β variables can be computed recursively, via dynamical programming,

$$\alpha^k_\tau(i) = \begin{cases} \dfrac{\mu(z^k_0 = i)\pi(a^k_0|z^k_0 = i)}{p(a^k_0|h^k_0, \Theta)}, & \tau = 0 \\ \dfrac{\sum_{j=1}^{|\mathcal{Z}|} \alpha^k_{\tau-1}(j) W(z^k_{\tau-1} = j, a^k_{\tau-1}, o^k_\tau, z^k_\tau = i)\pi(a^k_\tau|z^k_\tau = i)}{p(a^k_\tau|h^k_\tau, \Theta)}, & \tau > 0 \end{cases} \tag{35}$$

$$\beta^k_{t,\tau}(i) = \begin{cases} \dfrac{1}{p(a^k_t|h^k_t, \Theta)}, & \tau = t \\ \dfrac{\sum_{j=1}^{|\mathcal{Z}|} W(z^k_\tau = i, a^k_\tau, o^k_{\tau+1}, z^k_{\tau+1} = j)\pi(z^k_{\tau+1} = j, a^k_{\tau+1})\beta^k_{\tau+1}(j)}{p(a^k_\tau|h^k_\tau, \Theta)}, & \tau < t \end{cases} \tag{36}$$

for $t = 0, \cdots, T_k$ and $k = 1, \cdots, K$. Since $\sum_{i=1}^{|\mathcal{Z}|} \alpha^k_\tau(i) = 1$, it follows from (35) that

$$p(a^k_\tau|h^k_\tau, \Theta) = \begin{cases} \sum_{i=1}^{|\mathcal{Z}|} \mu(z^k_0 = i)\pi(a^k_0|z^k_0 = i), & \tau = 0 \\ \sum_{i=1}^{|\mathcal{Z}|} \sum_{j=1}^{|\mathcal{Z}|} \alpha^k_{\tau-1}(j) W(z^k_{\tau-1} = j, a^k_{\tau-1}, o^k_\tau, z^k_\tau = i)\pi(a^k_\tau|z^k_\tau = i), & \tau > 0 \end{cases} \tag{37}$$

4.2 Updating the Parameters

We rewrite the lower bound in (22),

$$\mathrm{LB}(\widehat{\Theta}|\Theta) = \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z^k_0, \cdots, z^k_t = 1}^{|\mathcal{Z}|} q^k_t(z^k_{0:t}|\Theta^{(n)}) \ln \frac{\tilde{r}^k_t p(a^k_{0:t}, z^k_{0:t}|o^k_{1:t}, \widehat{\Theta})}{q^k_t(z^k_{0:t}|\Theta^{(n)})}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k|\Theta^{(n)})\ln p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k,\widehat{\Theta}) + \text{constant} \quad (38)$$

where the "constant" collects all the terms irrelevant to $\widehat{\Theta}$. Substituting (6) and (18) gives

$$\begin{aligned}\text{LB}(\widehat{\Theta}|\Theta) &= \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\frac{\sigma_t^k}{\widehat{V}(\mathcal{D}^{(K)};\Theta)}\Bigg\{\sum_{i=1}^{|\mathcal{Z}|}\phi_{t,0}^k(i)\ln\widehat{\mu}(i) + \sum_{\tau=0}^{t}\sum_{i=1}^{|\mathcal{Z}|}\phi_{t,\tau}^k(i)\ln\widehat{\pi}(i,a_\tau^k) \\ &\quad + \sum_{\tau=1}^{t}\sum_{i,j=1}^{|\mathcal{Z}|}\xi_{t,\tau}^k(i,j)\ln\widehat{W}(i,a_{\tau-1}^k,o_\tau^k,j)\Bigg\} + \text{constant} \quad (39)\end{aligned}$$

It is not difficult to show that $\widehat{\Theta} = \arg\max_{\widehat{\Theta}\in\mathcal{F}}\text{LB}(\widehat{\Theta}|\Theta)$ is given by

$$\widehat{\mu}(i) = \frac{\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\phi_{t,0}^k(i)}{\sum_{i=1}^{|\mathcal{Z}|}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\phi_{t,0}^k(i)} \quad (40)$$

$$\widehat{\pi}(i,a) = \frac{\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\sum_{\tau=0}^{t}\phi_{t,\tau}^k(i)\,\delta(a_\tau^k,a)}{\sum_{a=1}^{|\mathcal{A}|}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\sum_{\tau=0}^{t}\phi_{t,\tau}^k(i)\,\delta(a_\tau^k,a)} \quad (41)$$

$$\widehat{W}(i,a,o,j) = \frac{\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\sum_{\tau=1}^{t-1}\xi_{t,\tau}^k(i,j)\,\delta(a_\tau^k,a)\,\delta(o_{\tau+1}^k,o)}{\sum_{j=1}^{|\mathcal{Z}|}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sigma_t^k\sum_{\tau=1}^{t-1}\xi_{t,\tau}^k(i,j)\,\delta(a_\tau^k,a)\,\delta(o_{\tau+1}^k,o)} \quad (42)$$

for $i,j = 1,2,\cdots,|\mathcal{Z}|$, $a = 1,\cdots,|\mathcal{A}|$, and $o = 1,\cdots,|\mathcal{O}|$, where $\delta(a,b) = \begin{cases}1, & a=b\\ 0, & a\neq b\end{cases}$, and σ_t^k is the recomputed reward as defined in (19). In computing σ_t^k one employs the equation $p(a_{0:t}^k|o_{1:t}^k,\Theta) = \prod_{\tau=0}^{t}p(a_\tau^k|h_\tau^k,\Theta)$ established in (10) and (11), to get

$$\sigma_t^k(\Theta) \overset{Def.}{=} \tilde{r}_t^k\prod_{\tau=0}^{t}p(a_\tau^k|h_\tau^k,\Theta) \quad (43)$$

with $p(a_\tau^k|h_\tau^k,\Theta)$ computed from the α variables by using (37). Note that the normalization constant, which is equal to the empirical value $\widehat{V}(\mathcal{D}^{(K)};\Theta)$, is now canceled in the update formulae of $\widehat{\Theta}$.

4.3 The Complete Value Maximization Algorithm for Single-Task RPR Learning

4.3.1 Algorithmic Description

The complete value maximization algorithm for single-task RPR learning is summarized in Table 1. In earlier discussions regarding the relations of the algorithm to EM, we have mentioned that reward computation constitutes an outer EM-like loop; the standard EM employed to solve (26) is embedded in the outer loop and constitutes an inner EM loop. The double EM loops are not explicitly shown in Table 1. However, one may separate these two loops by keeping $\{\sigma_t^k\}$ fixed when updating Θ and the posterior of z's, until the empirical value converges; see (28) for details. Once $\{\sigma_t^k\}$ are updated, the empirical value

will further increase by continuing updating Θ and the posterior of z's. Note that the $\{\sigma_t^k\}$ used in the convergence check are always updated at each iteration, even though the new $\{\sigma_t^k\}$ may not be used for updating Θ and the posterior of z's.

Table 1: The value maximization algorithm for single-task RPR learning

Input: $\mathcal{D}^{(K)}$, $\mathcal{A}$, $\mathcal{O}$, $|\mathcal{Z}|$.
Output: $\Theta = \{\mu, \pi, W\}$.

1. **Initialize** Θ, $\ell = [\,]$, iteration $= 1$.
2. **Repeat**
   2.1 **Dynamical programming:**
   Compute α and β variables with equations (35)(36)(37).
   2.2 **Reward re-computation:**
   Calculate $\{\sigma_t^k\}$ using (43)(37).
   2.3 **Convergence check:**
   Compute $\ell(\text{iteration}) = \widehat{V}(\mathcal{D}^{(K)}; \Theta)$ using (20).
   If the sequence of ℓ converges
   Stop the algorithm and exit.
   Else
   iteration := iteration + 1
   2.4 **Posterior update for z:**
   Compute the ξ and ϕ variables using equations (33)(34).
   2.5 **Update of Θ:**
   Compute the updated Θ using (40)(41)(42).

Given a history of actions and observations $(a_{0:t-1}, o_{1:t})$ collected up to time step t, the single RPR yields a distribution of a_t as given by (9). The optimal choice for a_t can be obtained by either sampling from this distribution or taking the action that maximizes the probability.

4.3.2 Time Complexity Analysis

We quantify the time complexity by the number of real number multiplications performed per iteration and present it in the Big-O notation. Since there is no compelling reason for the number of iterations to depend on the size of the input[2], the complexity per iteration also represents the complexity of the complete algorithm. A stepwise analysis of the time complexity of the value maximization algorithm in Table 1 is given as follows.

- Computation of the α variables with (35) and (37) runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^K T_k)$.
- Computation of β's with (36) and (37) runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^K \sum_{t=0, r_t^k \neq 0}^{T_k} (t+1))$, which depends on the degree of sparsity of the immediate rewards $\{r_0^k r_2^k \cdots r_{T_k}^k\}_{k=1}^K$. In the worst case the time is $O(|\mathcal{Z}|^2 \sum_{k=1}^K \sum_{t=0}^{T_k} (t+1)) = O(|\mathcal{Z}|^2 \sum_{k=1}^K T_k^2)$, which occurs when the immediate reward in each episode is nonzero at every time step. In the best case the time is $O(|\mathcal{Z}|^2 \sum_{k=1}^K T_k)$, which occurs when the immediate reward

2. The number of iterations usually depends on such factors as initialization of the algorithm and the required accuracy, etc.

in each episode is nonzero only at a fixed number of time steps (only at the last time step, for example, as is the case of the benchmark problems presented in Section 6).

- The reward re-computation using (43) and (37) requires time $O(\sum_{k=1}^{K} T_k)$ in the worst case and $O(K)$ in the best case, where the worse/best cases are as defined above.

- Update of Θ using (40), (41), and (42), as well as computation of the ξ and ϕ variables using (33) and (34), runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k^2)$ in the worst case and $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k)$ in the best case, where the worse/best cases are defined above.

Since $\sum_{k=1}^{K} T_k \gg |\mathcal{A}||\mathcal{O}|$ in general, the overall complexity of the value maximization algorithm is $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k^2)$ in the worst case and $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k)$ in the best case, depending on the degree of sparsity of the immediate rewards. Therefore the algorithm scales linearly with the number of episodes and to the square of the number of belief regions. The time dependency on the lengths of episodes is between linear and square. The sparser the immediate rewards are, the more the time is towards being linear in the lengths of episodes.

Note that in many reinforcement problems, the agent does not receive immediate rewards at every time step. For the benchmark problems and maze navigation problems considered in Section 6, the agent receives rewards only when the goal state is reached, which makes the value maximization algorithm scale linearly with the lengths of episodes.

5. Multi-Task Reinforcement Learning (MTRL) with RPR

We formulate our MTRL framework by placing multiple RPRs in a Bayesian setting and develop techniques to learn the posterior of each RPR within the context of all other RPRs.

Several notational conventions are observed in this section. The posterior of Θ is expressed in terms of probability density functions. The notation $G_0(\Theta)$ is reserved to denote the density function of a parametric prior distribution, with the associated probability measure denoted by G_0 without a parenthesized Θ beside it. For the Dirichlet process (which is a nonparametric prior), G_0 denotes the base measure and $G_0(\Theta)$ denotes the corresponding density function. The twofold use of G_0 is for notational simplicity; the difference can be easily discerned by the presence or absence of a parenthesized Θ. The δ is a Dirac delta for continuous arguments and a Kronecker delta for discrete arguments. The notation δ_{Θ_j} is the Dirac measure satisfying $\delta_{\Theta_j}(d\Theta_m) = \begin{cases} 1, & \Theta_j \in d\Theta_m \\ 0, & \text{otherwise} \end{cases}$.

5.1 Basic Bayesian Formulation of RPR

Consider M partially observable and stochastic environments indexed by $m = 1, 2 \cdots, M$, each of which is apparently different from the others but may actually share fundamental common characteristics with some other environments. Assume we have a set of episodes collected from each environment, $\mathcal{D}_m^{(K_m)} = \left\{ (a_0^{m,k} r_0^{m,k} o_1^{m,k} a_1^{m,k} r_1^{m,k} \cdots o_{T_{m,k}}^{m,k} a_{T_{m,k}}^{m,k} r_{T_{m,k}}^{m,k}) \right\}_{k=1}^{K_m}$, for $m = 1, 2, \cdots, M$, where $T_{m,k}$ represents the length of episode k in environment m. Following the definitions in Section 3, we write the empirical value function of the m-th

environment as

$$\widehat{V}(\mathcal{D}_m^{(K_m)};\Theta_m) = \frac{1}{K_m}\sum_{k=1}^{K_m}\sum_{t=0}^{T_{m,k}} \tilde{r}_t^{m,k} p(a_{0:t}^{m,k}|o_{1:t}^{m,k},\Theta_m) \tag{44}$$

for $m = 1, 2, \cdots, M$, where $\Theta_m = \{\pi_m, \mu_m, W_m\}$ are the RPR parameters for the m-th individual environment.

Let $G_0(\Theta_m)$ represent the prior of Θ_m, where $G_0(\Theta)$ is assumed to be the density function of a probability distribution. We define the posterior of Θ_m as

$$p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0) \overset{Def.}{=} \frac{\widehat{V}(\mathcal{D}_m^{(K_m)};\Theta_m)G_0(\Theta_m)}{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{45}$$

where the inclusion of G_0 in the left hand side is to explicitly indicate that the prior being used is G_0, and $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ is a normalization constant

$$\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)}) \overset{Def.}{=} \int \widehat{V}(\mathcal{D}_m^{(K_m)};\Theta_m)G_0(\Theta_m)d\Theta_m \tag{46}$$

which is also referred to as the *marginal empirical value*[3], since the parameters Θ_m are integrated out (marginalized). The marginal empirical value $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ represents the accumulated discounted reward in the episodes, averaged over infinite RPR policies independently drawn from G_0.

Equation (45) is literally a normalized product of the empirical value function and a prior $G_0(\Theta_m)$. Since $\int p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0)d\Theta_m = 1$, (45) yields a valid probability density, which we call the posterior of Θ_m given the episodes $\mathcal{D}_m^{(K_m)}$. It is noted that (45) would be the Bayes rule if $\widehat{V}(\mathcal{D}_m^{(K_m)};\Theta_m)$ were a likelihood function. Since $\widehat{V}(\mathcal{D}_m^{(K_m)};\Theta_m)$ is a value function in our case, (45) is a somewhat non-standard use of Bayes rule. However, like the classic Bayes rule, (45) indeed gives a posterior whose shape incorporates both the prior information about Θ_m and the empirical information from the episodes.

Equation (45) has another interpretation that may be more meaningful from the perspective of standard probability theory. To see this we substitute (44) into (45) to obtain

$$p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0) = \frac{\frac{1}{K_m}\sum_{k=1}^{K_m}\sum_{t=0}^{T_{m,k}} \tilde{r}_t^{m,k} p(a_{0:t}^{m,k}|o_{1:t}^{m,k},\Theta_m)G_0(\Theta_m)}{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{47}$$

$$= \frac{\frac{1}{K_m}\sum_{k=1}^{K_m}\sum_{t=0}^{T_{m,k}} \nu_t^{m,k} p(\Theta_m|a_{0:t}^{m,k}, o_{1:t}^{m,k}, G_0)}{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{48}$$

where

$$\begin{aligned}\nu_t^{m,k} &= \tilde{r}_t^{m,k} p(a_{0:t}^{m,k}|o_{1:t}^{m,k}, G_0) \\ &= \tilde{r}_t^{m,k}\int p(a_{0:t}^{m,k}|o_{1:t}^{m,k},\Theta_m)G_0(\Theta_m)d\Theta_m\end{aligned}$$

3. The term "marginal" is borrowed from the probability theory. Here we use it to indicate that the dependence of the value on the parameter is removed by integrating out the parameter.

$$= \int \sigma_t^{m,k}(\Theta_m) G_0(\Theta_m) d\Theta_m \tag{49}$$

with $\sigma_t^{m,k}$ the re-computed reward as defined in (19) and therefore $\nu_t^{m,k}$ is the averaged re-computed reward, obtained by taking the expectation of $\sigma_t^{m,k}(\Theta_m)$ with respect to $G_0(\Theta_m)$.

In arriving (48), we have used the fact the RPR parameters are independent of the observations, which is true due to the following reasons: RPR is a policy concerning generation of the actions, employing as input the observations (which themselves are generated by the unknown environment); therefore, observations carry no information about the RPR parameters, i.e., $p(\Theta|\text{observations}) = p(\Theta) \equiv G_0(\Theta)$.

It is noted that $p(\Theta_m|a_{0:t}^{m,k}, o_{1:t}^{m,k}, G_0)$ in (48) is the standard posterior of Θ_m given the action sequence $a_{0:t}^{m,k}$, and $p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0)$ is a mixture of these posteriors with the mixing proportion given by $\nu_t^{m,k}$. The meaning of (47) is fairly intuitive: each action sequence affects the posterior of Θ_m in proportion to its re-evaluated reward. This is distinct from the posterior in the classic hidden Markov model (Rabiner 1989) where sequences are treated as equally important.

Since $p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0)$ integrates to one, the normalization constant $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ is

$$\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)}) = \frac{1}{K_m} \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} \nu_t^{m,k} \tag{50}$$

We obtain a more convenient form of the posterior by substituting (7) into (48) to expand the summation over the latent z variables, yielding

$$p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0) = \frac{\frac{1}{K_m} \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} \tilde{r}_t^{m,k} \sum_{z_0^{m,k}, \cdots, z_t^{m,k}=1}^{|\mathcal{Z}|} p(a_{0:t}^{m,k}, z_{0:t}^{m,k}|o_{1:t}^{m,k}, \Theta_m) G_0(\Theta_m)}{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{51}$$

To obtain an analytic posterior, we let the prior be conjugate to $p(a_{0:t}^{m,k}, z_{0:t}^{m,k}|o_{1:t}^{m,k}, \Theta_m)$. As shown by (6), $p(a_{0:t}^{m,k}, z_{0:t}^{m,k}|o_{1:t}^{m,k}, \Theta_m)$ is a product of multinomial distributions, and hence we choose the prior as a product of Dirichlet distributions, with each Dirichlet representing an independent prior for a subset of parameters in Θ. The density function of such a prior is given by

$$\begin{aligned}
G_0(\Theta_m) &= p(\mu^m|\upsilon) p(\pi^m|\rho) p(W^m|\omega) && (52)\\
p(\mu^m|\upsilon) &= \text{Dir}\Big(\mu^m(1), \cdots, \mu^m(|\mathcal{Z}|)\Big|\upsilon\Big) && (53)\\
p(\pi^m|\rho) &= \prod_{i=1}^{|\mathcal{Z}|} \text{Dir}\Big(\pi^m(i,1), \cdots, \pi^m(i,|\mathcal{A}|)\Big|\rho_i\Big) && (54)\\
p(W^m|\omega) &= \prod_{a=1}^{|\mathcal{A}|} \prod_{o=1}^{|\mathcal{O}|} \prod_{i=1}^{|\mathcal{Z}|} \text{Dir}\Big(W^m(i,a,o,1), \cdots, W^m(i,a,o,|\mathcal{Z}|)\Big|\omega_{i,a,o}\Big) && (55)
\end{aligned}$$

where $\upsilon = \{\upsilon_1, \ldots, \upsilon_{|\mathcal{Z}|}\}$, $\rho = \{\rho_1, \ldots, \rho_{|\mathcal{Z}|}\}$ with $\rho_i = \{\rho_{i,1}, \ldots, \rho_{i,|\mathcal{A}|}\}$, and $\omega = \{\omega_{i,a,o} : i = 1 \ldots |\mathcal{Z}|, a = 1 \ldots |\mathcal{A}|, o = 1 \ldots |\mathcal{O}|\}$ with $\omega_{i,a,o} = \{\omega_{i,a,o,1}, \ldots, \omega_{i,a,o,|\mathcal{Z}|}\}$. Substituting the expression of G_0 into (51), one gets

$$p(\Theta_m|\mathcal{D}_m^{(K_m)}, G_0)$$

$$= \frac{\frac{1}{K_m}\sum_{k=1}^{K_m}\sum_{t=0}^{T_{m,k}}\sum_{z_0^{m,k},\cdots,z_t^{m,k}=1}^{|\mathcal{Z}|}\nu_t^{m,k}(z_{0:t}^{m,k})\; p(\Theta_m|a_{0:t}^{m,k}, o_{1:t}^{m,k}, z_{0:t}^{m,k}, G_0)}{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{56}$$

where

$$\begin{aligned}\nu_t^{m,k}(z_{0:t}^{m,k}) &= \tilde{r}_t^{m,k}\int p(a_{0:t}^{m,k}, z_{0:t}^{m,k}|o_{1:t}^{m,k}, \Theta_m)G_0(\Theta_m)d\Theta_m \\ &= \tilde{r}_t^{m,k}\frac{\prod_i\Gamma(\widehat{\upsilon}_i^{m,k,t})}{\Gamma(\sum_i\widehat{\upsilon}_i^{m,k,t})}\frac{\Gamma(\sum_i\upsilon_i^{m,k,t})}{\prod_i\Gamma(\upsilon_i^{m,k,t})}\frac{\prod_i\prod_a\Gamma(\widehat{\rho}_{i,a}^{m,k,t})}{\prod_i\Gamma(\sum_a\widehat{\rho}_{i,a}^{m,k,t})}\frac{\prod_i\Gamma(\sum_a\rho_{i,a}^{m,k,t})}{\prod_i\prod_a\Gamma(\rho_{i,a}^{m,k,t})} \\ &\quad\times\frac{\prod_a\prod_o\prod_i\prod_j\Gamma(\widehat{\omega}_{i,a,o,j}^{m,k,t})}{\prod_a\prod_o\prod_i\Gamma(\sum_j\widehat{\omega}_{i,a,o,j}^{m,k,t})}\frac{\prod_a\prod_o\prod_i\Gamma(\sum_j\omega_{i,a,o,j}^{m,k,t})}{\prod_a\prod_o\prod_i\prod_j\Gamma(\omega_{i,a,o,j}^{m,k,t})}\end{aligned} \tag{57}$$

represents the averaged recomputed reward over a given z sequence $z_{0:t}^{m,k}$, and

$$p(\Theta_m|a_{0:t}^{m,k}, o_{1:t}^{m,k}, z_{0:t}^{m,k}, G_0) = p(\mu^m|\widehat{\upsilon}^{m,k,t})p(\pi^m|\widehat{\rho}^{m,k,t})p(W^m|\widehat{\omega}^{m,k,t}) \tag{58}$$

is the density of a product of Dirichlet distributions and has the same form as $G_0(\Theta)$ in (52) but with υ, ρ, ω respectively replaced by $\widehat{\upsilon}^{m,k,t}$, $\widehat{\rho}^{m,k,t}$, $\widehat{\omega}^{m,k,t}$ as given by

$$\widehat{\upsilon}_i^{m,k,t} = \upsilon_i^m + \delta(z_0^{m,k} - i) \tag{59}$$

$$\widehat{\rho}_{i,a}^{m,k,t} = \rho_{i,a}^m + \sum_{\tau=0}^{t}\delta(z_\tau^{m,k} - i)\delta(a_\tau^{m,k} - a) \tag{60}$$

$$\widehat{\omega}_{i,a,o,j}^{m,k,t} = \omega_{i,a,o,j}^m + \sum_{\tau=1}^{t}\delta(z_{\tau-1}^{m,k} - i)\delta(a_{\tau-1}^{m,k} - a)\delta(o_\tau^{m,k} - o)\delta(z_\tau^{m,k} - j) \tag{61}$$

The normalization constant $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ (which is also the marginal empirical value) can now be expressed as

$$\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)}) = \frac{1}{K_m}\sum_{k=1}^{K_m}\sum_{t=0}^{T_{m,k}}\sum_{z_0^{m,k},\cdots,z_t^{m,k}=1}^{|\mathcal{Z}|}\nu_t^{m,k}(z_{0:t}^{m,k}) \tag{62}$$

5.2 The Dirichlet Process Prior

In order to identify related tasks and introduce sharing mechanisms for multi-task learning, we employ the Dirichlet process (Ferguson 1973; Blackwell and MacQueen 1973; Antoniak 1974; Sethuraman 1994) as a nonparametric prior that is shared by Θ_m, $m = 1, 2, \cdots, M$. A draw from a DP has the nice property of being almost surely discrete (Blackwell and MacQueen 1973), which is known to promote clustering (West et al. 1994); therefore, related tasks (as judged by the empirical value function) are encouraged to be placed in the same group and be learned simultaneously by sharing the episodic data across all tasks in the same group. Assuming the prior of Θ_m, $m = 1, 2, \cdots, M$, is drawn from a Dirichlet process with base measure G_0 and precision α, we have

$$\Theta_m|G \sim G$$

$$G|\alpha, G_0 \quad \sim \quad DP(\alpha, G_0) \tag{63}$$

where the precision α provides an expected number of dominant clusters, with this driven by the number of samples (West 1992). It usually suffices to set the precision α using the rule in (West 1992). If desired, however, one may also put a Gamma prior on α and draw from its posterior (Escobar and West 1995), which yields greater model flexibility. Note the DP precision is denoted by the same symbol as the α variables in (31). The difference is easy to recognize, since the former is a single quantity bearing neither superscripts and nor subscripts while the latter represent a set of variables and always bear superscripts and subscripts.

By marginalizing out G, one obtains the Polya-urn representation of DP (Blackwell and MacQueen 1973), expressed in terms of density functions [4]

$$p(\Theta_m|\Theta_{-m}, \alpha, G_0) = \frac{\alpha}{\alpha+M-1} G_0(\Theta_m) + \frac{1}{\alpha+M-1} \sum_{\substack{j=1 \\ j\neq m}}^{M} \delta(\Theta_m - \Theta_j), \quad m = 1, \cdots, M \tag{64}$$

where the probability is conditioned on $\Theta_{-m} = \{\Theta_1, \Theta_2, \cdots, \Theta_M\} \setminus \{\Theta_m\}$. The Polya-urn representation in (64) gives a set of full conditionals for the joint prior $p(\Theta_1, \Theta_2, \cdots, \Theta_M)$.

The fact that $G \sim DP(\alpha, G_0)$ is almost surely discrete implies that the set $\{\Theta_1, \Theta_2, \cdots, \Theta_M\}$, which are iid drawn from G, can have duplicate elements and the number of distinct elements N cannot exceed M, the total number of environments. It is useful to consider an equivalent representation of (64) based on the distinct elements (Neal 1998). Let $\overline{\Theta} = \{\overline{\Theta}_1, \overline{\Theta}_2, \cdots, \overline{\Theta}_N\}$ represent the set of distinct elements of $\{\Theta_1, \Theta_2, \cdots, \Theta_M\}$, with $N \leq M$. Let $c = \{c_1, c_2, \ldots, c_M\}$ denote the vector of indicator variables defined by $c_m = n$ iff $\Theta_m = \overline{\Theta}_n$ and $c_{-m} = \{c_1, c_2, \cdots, c_M\} \setminus \{c_m\}$. The prior conditional distribution $p(c_m|c_{-m})$ that arises from the Polya-urn representation of the Dirichlet process is as follows (MacEachern 1994)

$$p(c_m|c_{-m}, \alpha) = \frac{\alpha}{\alpha + M - 1}\delta(c_m) + \sum_{n=1}^{N} \frac{l_{-m,n}}{\alpha + M - 1}\delta(c_m - n) \tag{65}$$

where $l_{-m,n}$ denotes the number of elements in $\{i : c_i = n, i \neq m\}$ and $c_m = 0$ indicates a new sample is drawn from the base G_0. Given c_m and $\overline{\Theta}$, the density of Θ_m is given by

$$p(\Theta_m|c_m, \overline{\Theta}, G_0) = \delta(c_m)G_0(\Theta_m) + \sum_{n=1}^{N} \delta(c_m - n)\delta(\Theta_m - \overline{\Theta}_n) \tag{66}$$

4. The corresponding expression in terms of probability measures (Escobar and West 1995) is given by

$$\Theta_m|\Theta_{-m}, \alpha, G_0 \sim \frac{\alpha}{\alpha + M - 1} G_0 + \frac{1}{\alpha + M - 1} \sum_{j=1, j\neq m}^{M} \delta_{\Theta_j}, \quad m = 1, \cdots, M,$$

where δ_{Θ_j} is the Dirac measure.

5.3 The Dirichlet Process Posterior

We take two steps to derive the posterior based on the representation of the DP prior given by (65) and (66). First we write the conditional posterior of c_m, $\forall\ m \in \{1, \cdots, M\}$,

$$p(c_m|c_{-m}, \overline{\Theta}, \mathcal{D}_m^{(K_m)}, \alpha, G_0) = \frac{\int \widehat{V}(\mathcal{D}_m^{(K_m)}; \Theta_m) p(\Theta_m|c_m, \overline{\Theta}, G_0) p(c_m|c_{-m}, \alpha) d\Theta_m}{\sum_{c_m=0}^{N} \int \widehat{V}(\mathcal{D}_m^{(K_m)}; \Theta_m) p(\Theta_m|c_m, \overline{\Theta}, G_0) p(c_m|c_{-m}, \alpha) d\Theta_m} \tag{67}$$

which is rewritten, by substituting (65) and (66) into the righthand side, to yield an algorithmically more meaningful expression

$$p(c_m|c_{-m}, \overline{\Theta}, \mathcal{D}_m^{(K_m)}, \alpha, G_0) = \frac{\alpha\, \widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})\, \delta(c_m) \ + \ \sum_{n=1}^{N} l_{-m,n} \widehat{V}(\mathcal{D}_m^{(K_m)}; \overline{\Theta}_n)\, \delta(c_m - n)}{\alpha\, \widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)}) + \sum_{j=1}^{N} l_{-m,j} \widehat{V}(\mathcal{D}_m^{(K_m)}; \overline{\Theta}_j)} \tag{68}$$

where the $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ is the marginal empirical value defined in (46) and its expression is given by (62) when the DP base has a density function as specified in (52).

It is observed from (68) that the indicator c_m tends to equal n if $\widehat{V}(\mathcal{D}_m^{(K_m)}; \overline{\Theta}_n)$ is large, which occurs when the n-th distinct RPR produces a high empirical value in the m-th environment. If none of the other RPRs produces a high empirical value in the m-th environment, c_m will tend to be equal to zero, which means a new cluster will be generated to account for the novelty. The merit of generating a new cluster is measured by the empirical value weighted by α and averaged with respect to G_0. Therefore the number of distinct RPRs is jointly dictated by the DP prior and the episodes.

Given the indicator variables c, the clusters are formed. Let $I_n(c) = \{m : c_m = n\}$ denote the indices of the environments that have been assigned to the n-th cluster. Given the clusters, we now derive the conditional posterior of $\overline{\Theta}_n$, $\forall\ n \in \{1, \cdots, N\}$. If $I_n(c)$ is an empty set, there is no empirical evidence available for it to obtain a posterior, therefore one simply removes this cluster. If $I_n(c)$ is nonempty, the density function of the conditional posterior of $\overline{\Theta}_n$ is given by

$$p(\overline{\Theta}_n|\bigcup\nolimits_{m\in I_n(c)} \mathcal{D}_m^{(K_m)}, G_0) = \frac{\sum_{m\in I_n(c)} \widehat{V}(\mathcal{D}_m^{(K_m)}; \overline{\Theta}_n) G_0(\overline{\Theta}_n)}{\int \sum_{m\in I_n(c)} \widehat{V}(\mathcal{D}_m^{(K_m)}; \overline{\Theta}_n) G_0(\overline{\Theta}_n)\, d\overline{\Theta}_n} \tag{69}$$

$$= \frac{\sum_{m\in I_n(c)} \frac{1}{K_m} \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} \tilde{r}_t^{m,k} \sum_{z_0^{m,k}, \cdots, z_t^{m,k}=1}^{|\mathcal{Z}|} p(a_{0:t}^{m,k}, z_{0:t}^{m,k}|o_{1:t}^{m,k}, \overline{\Theta}_n) G_0(\overline{\Theta}_n)}{\sum_{m\in I_n(c)} \widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \tag{70}$$

where (70) results from substituting (13) into the righthand side of (69). Note that $\overline{\Theta}_n$, which represents the set of parameters of the n-th distinct RPR, is conditioned on all episodes aggregated across all environments in the n-th cluster. The posterior in (69) has the same form as the definition in (45) and it is obtained by applying Bayes law to the empirical value function constructed from the aggregated episodes. As before, the Bayes law is applied in a nonstandard manner, treating the value function as if it were a likelihood function.

A more concrete expression of (70) can be obtained by letting the DP base G_0 have a density function as in (52),

$p(\overline{\Theta}_n|\bigcup_{m\in I_n(c)} \mathcal{D}_m^{(K_m)}, G_0)$

$$= \frac{\sum_{m\in I_n(c)} \frac{1}{K_m} \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} \tilde{r}_t^{m,k} \sum_{z_0^{m,k},\cdots,z_t^{m,k}=1}^{|\mathcal{Z}|} \nu_t^{m,k}(z_{0:t}^{m,k})\; p(\overline{\Theta}_n|a_{0:t}^{m,k}, o_{1:t}^{m,k}, z_{0:t}^{m,k}, G_0)}{\sum_{m\in I_n(c)} \widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})} \quad (71)$$

where $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ is the marginal empirical value given in (62), $\nu_t^{m,k}(z_{0:t}^{m,k})$ is the average recomputed reward as given in (57), and

$$p(\overline{\Theta}_n|a_{0:t}^{m,k}, o_{1:t}^{m,k}, z_{0:t}^{m,k}, G_0) \;=\; p(\overline{\mu}^n|\widehat{\upsilon}^{m,k,t})p(\overline{\pi}^n|\widehat{\rho}^{m,k,t})p(\overline{W}^n|\widehat{\omega}^{m,k,t}) \quad (72)$$

is the density of a product of Dirichlet distributions and has the same form as $G_0(\Theta)$ in (52) but with υ, ρ, ω respectively replaced by $\widehat{\upsilon}^{m,k,t}$, $\widehat{\rho}^{m,k,t}$, $\widehat{\omega}^{m,k,t}$ as given by (59), (60), and (61).

It is noted that, conditional on the indicator variables c and the episodes across all environments, the distinct RPRs are independent of each other. The indicator variables cluster the M environments into $N \leq M$ groups, each of which is associated with a distinct RPR. Given the clusters, the environments in the n-th group merge their episodes to form a pool, and the posterior of $\overline{\Theta}_n$ is derived based on this pool. Existing clusters may become empty and be removed, and new clusters may be introduced when novelty is detected, thus the pools change dynamically. The dynamic changes are implemented inside the algorithm presented below. Therefore, the number of distinct RPRs is not fixed but is allowed to vary.

5.4 Challenges for Gibbs Sampling

The DP posterior as given by (68) and (71) may be analyzed using the technique of Gibbs sampling (Geman and Geman 1984; Gelfand and Smith 1990). The Gibbs sampler successively draws the indicator variables $c_1, c_2, \cdots, c_M$ and the distinct RPRs $\overline{\Theta}_1, \overline{\Theta}_2, \cdots, \overline{\Theta}_N$ according to (68) and (71). The samples are expected to represent the posterior when the Markov chain produced by the Gibbs sampler reaches the stationary distribution. However, the convergence of Gibbs sampling can be slow and a long sequence of samples may be required before the stationary distribution is reached. The slow convergence can generally be attributed to the fact that the Gibbs sampler implements message-passing between dependent variables through the use of samples, instead of sufficient statistics (Jordan et al. 1999). Variational methods have been suggested as a replacement for Gibbs sampling (Jordan et al. 1999). Though efficient, variational methods are known to suffer from bias. A good trade-off is to combine the two, which is the idea of hybrid variational/Gibbs inference in (Welling et al. 2008).

In our present case, Gibbs sampling is further challenged by the particular form of the conditional posterior of $\overline{\Theta}_n$ in (71), which is seen to be a large mixture resulting from the summation over environment m, episode k, time step t, and latent z variables. Thus it has a total of $\sum_{m\in I_n} \sum_{k=1}^{K_m} \sum_{t=0}^{T_{m,k}} |\mathcal{Z}|^t$ components and each component is uniquely associated with a single sub-episode and a specific instantiation of latent z variables. To sample from this mixture, one first makes a draw to decide a component and then draws $\overline{\Theta}_n$ from this component. Obviously, any particular draw of $\overline{\Theta}_n$ makes use of one single sub-episode only, instead of simultaneously employing all sub-episodes in the n-th cluster as one would wish.

In essence, mixing with respect to (m, k, t) effectively introduces additional latent indicator variables, i.e., those for locating environment m, episode k, and time step t. It is

important to note that these new indicator variables play a different role than z's in affecting the samples of $\overline{\Theta}_n$. In particular, the z's are intrinsic latent variables inside the RPR model, while the new ones are extrinsic latent variables resulting from the particular form of the empirical value function in (44). Each realization of the new indicators is uniquely associated with a distinct sub-episode while each realization of z's is uniquely associated specific decision states. Therefore, the update of $\overline{\Theta}_n$ based on one realization of the new indicators employs a single sub-episode, but the update based on one realization of z's employs all sub-episodes.

5.5 The Gibbs-Variational Algorithm for Learning the DP Posterior

The fact that the Gibbs sampler cannot update the posterior RPR samples by using more than one sub-episode motivates us to develop a hybrid Gibbs-variational algorithm for learning the posterior.

We restrict the joint posterior of the latent z variables and the RPR parameters to the variational Bayesian (VB) approximation that assumes a factorized form. This restriction yields a variational approximation to $p(\overline{\Theta}_n|\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)}, G_0)$ that is a single product of Dirichlet density functions, where the terms associated with different episodes are collected and added up. Therefore, updating of the variational posterior of $\overline{\Theta}_n$ in each Gibbs-variational iteration is based on simultaneously employing all sub-episodes in the n-th cluster. In addition, the variational method yields an approximation of the marginal empirical value $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ as given in (46).

The overall Gibbs-variational algorithm is an iterative procedure based on the DP posterior represented by (68) and (69). At each iteration one successively performs the following for $m = 1, 2, \cdots, M$. First, the cluster indicator variable c_m is drawn according to (68), where $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ is replaced by its variational Bayesian approximation; accordingly the clusters $I_n = \{m : c_m = n\}$, $n = 1, \ldots, N$ are updated. For each nonempty cluster n, the associated distinct RPR is updated by drawing from, or finding the mode of, the variational Bayesian approximation of $p(\overline{\Theta}_n|\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)}, G_0)$. The steps are iterated until the variational approximation of $\sum_{n=1}^{N}\widehat{V}_{G_0}(\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)})$ converges. Note that the number of clusters is not fixed but changes with the iteration, since existing clusters may become empty and be removed and new clusters may be added in.

5.5.1 Variational Bayesian Approximation of $\widehat{V}_{G_0}(\mathcal{D}^{(K)})$ and $p(\Theta|\mathcal{D}^{(K)}, G_0)$

In this subsection we drop the variable dependence on environment m, for notational simplicity. The discussion assumes a set of episodes $\mathcal{D}^{(K)} = \{(a_0^k r_0^k o_1^k a_1^k r_1^k \cdots o_{T_k}^k a_{T_k}^k r_{T_k}^k)\}_{k=1}^K$, which may come from a single environment or a conglomeration of several environments.

We now derive the variational Bayesian approximation of the marginal empirical value function $\widehat{V}_{G_0}(\mathcal{D}^{(K)})$ as defined in (46). We begin by rewriting (46), using (7) and (44), as

$$\widehat{V}_{G_0}(\mathcal{D}^{(K)}) \;=\; \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\tilde{r}_t^k \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}\int p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \Theta)G_0(\Theta)\,d\Theta \tag{73}$$

We follow the general variational Bayesian approach (Jordan et al. 1999; Jaakkola 2001; Beal 2003) [5] to find a variational lower bound to $\ln \widehat{V}_{G_0}(\mathcal{D}^{(K)})$, and the variational Bayesian approximation of $\widehat{V}_{G_0}(\mathcal{D}^{(K)})$ is obtained as the exponential of the lower bound. The lower bound is a functional of a set of factorized forms $\{q_t^k(z_{0:t}^k)g(\Theta) : z_t^k \in \mathcal{Z}, t = 1 \dots T_k, k = 1 \dots K\}$ that satisfies the following normalization constraints:

$$\sum_{k=1}^{K}\sum_{t=1}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k) = K \quad \text{and} \quad q_t^k(z_{0:t}^k) \geq 0 \; \forall \, z_{0:t}^k, t, k$$
$$\int g(\Theta) d\Theta = 1 \quad \text{and} \quad g(\Theta) \geq 0 \; \forall \, \Theta$$

The lower bound is maximized with respect to $\left\{q_t^k(z_{0:t}^k)g(\Theta)\right\}$. As will come clear below, maximization of the lower bound is equivalent to minimization of the Kullback-Leibler (KL) distance between the factorized forms and *weighted* true joint posterior of z's and Θ. In this sense, the optimal $g(\Theta)$ is a variational Bayesian approximation to the posterior $p(\Theta|\mathcal{D}^{(K)}, G_0)$. It should be noted that, as before, the weights result from the empirical value function and are not a part of standard VB (as applied to likelihood functions).

The variational lower bound is obtained by applying Jensen's inequality to $\ln \widehat{V}_{G_0}(\mathcal{D}^{(K)})$,

$$\begin{aligned}
& \ln \widehat{V}_{G_0}(\mathcal{D}^{(K)}) \\
= \; & \ln \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} \int q_t^k(z_{0:t}^k)g(\Theta) \frac{\tilde{r}_t^k G_0(\Theta) p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)}{q_t^k(z_{0:t}^k)g(\Theta)} \, d\Theta \\
\geq \; & \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} \int q_t^k(z_{0:t}^k)g(\Theta) \ln \frac{\tilde{r}_t^k G_0(\Theta) p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)}{q_t^k(z_{0:t}^k)g(\Theta)} \, d\Theta \\
= \; & \ln \widehat{V}_{G_0}(\mathcal{D}^{(K)}) - \mathrm{KL}\left(\left\{q_t^k(z_{0:t}^k)g(\Theta)\right\} \middle\| \left\{\frac{\nu_t^k}{\widehat{V}_{G_0}(\mathcal{D}^{(K)})} p(z_{0:t}^k, \Theta | a_{0:t}^k, o_{1:t}^k)\right\}\right) \\
\overset{Def.}{=} \; & \mathrm{LB}\left(\left\{q_t^k\right\}, g(\Theta)\right)
\end{aligned} \tag{74}$$

where ν_t^k is the average recomputed reward as given in (49), and

$$\begin{aligned}
& \mathrm{KL}\left(\left\{q_t^k(z_{0:t}^k)g(\Theta)\right\} \middle\| \left\{\frac{\nu_t^k}{\widehat{V}_{G_0}(\mathcal{D}^{(K)})} p(z_{0:t}^k, \Theta | a_{0:t}^k, o_{1:t}^k)\right\}\right) \\
= \; & \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} \int q_t^k(z_{0:t}^k)g(\Theta) \ln \frac{q_t^k(z_{0:t}^k)g(\Theta)}{\frac{\nu_t^k}{\widehat{V}_{G_0}(\mathcal{D}^{(K)})} p(z_{0:t}^k, \Theta | a_{0:t}^k, o_{1:t}^k)} \, d\Theta
\end{aligned} \tag{75}$$

with $\mathrm{KL}(q||p)$ denoting the Kullback-Leibler distance.

For any set $\left\{q_t^k(z_{0:t}^k)g(\Theta) : z_t^k \in \mathcal{Z}, t = 1 \dots T_k, k = 1 \dots K\right\}$ satisfying the above normalization constraints, the inequality in (74) holds. In order to obtain the lower bound that is

5. The standard VB applies to a likelihood function. Since we are using a value function instead of a likelihood function, the VB derivation here is not a standard one, just as the Bayes rule in (45) is non-standard.

closest to $\ln \widehat{V}(\mathcal{D}^{(K)})$, one maximizes the lower bound by optimizing $\left(\left\{q_t^k\right\}, g(\Theta)\right)$ subject to the normalization constraints. Since $\ln \widehat{V}_{G_0}(\mathcal{D}^{(K)})$ is independent of Θ and $\{q_t^k\}$, it is clear that maximization of the lower bound $\mathrm{LB}\left(\left\{q_t^k\right\}, g(\Theta)\right)$ is equivalent to minimization of the KL distance between $\left\{q_t^k(z_{0:t}^k)g(\Theta)\right\}$ and the weighted posterior $\left\{\frac{\nu_t^k}{\widehat{V}_{G_0}(\mathcal{D}^{(K)})}p(z_{0:t}^k, \Theta|a_{0:t}^k, o_{1:t}^k)\right\}$, where the weight for episode k at time step t is $\frac{\nu_t^k}{\widehat{V}_{G_0}(\mathcal{D}^{(K)})} = K\frac{\nu_t^k}{\sum_{k=1}^K\sum_{t=0}^{T_k}\nu_t^k}$ (the equation results directly from (50)), i.e., K times the fraction that the average recomputed reward ν_t^k occupies in the total average recomputed reward. Therefore the factorized form $\left\{q_t^k(z_{0:t}^k)g(\Theta)\right\}$ represents an approximation of the weighted posterior when the lower bound reaches the maximum, and the corresponding $g(\Theta)$ is called the approximate variational posterior of Θ.

The lower bound maximization is accomplished by solving $\left\{q_t^k(z_{0:t}^k)\right\}$ and $q(\Theta)$ alternately, keeping one fixed while solving for the other, as shown in Theorem 8.

Theorem 8. *Iteratively applying the following two equations produces a sequence of monotonically increasing lower bounds* $\mathrm{LB}\left(\left\{q_t^k\right\}, g(\Theta)\right)$*, which converges to a maxima,*

$$q_t^k(z_{0:t}^k) = \frac{\tilde{r}_t^k}{C_z}\exp\left\{\int g(\Theta)\ln p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \Theta)\,d\Theta\right\} \tag{76}$$

$$g(\Theta) = \frac{G_0(\Theta)}{C_\Theta}\exp\left\{\frac{1}{K}\sum_{k=1}^K\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k)\ln\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \Theta)\right\} \tag{77}$$

where C_z and C_Θ are normalization constants such that $\int g(\Theta)d\Theta = 1$ and $\sum_{k=1}^K\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k) = K$.

It is seen from (77) that the variational posterior $g(\Theta)$ takes the form of a product, where each term in the product is uniquely associated with a sub-episode. As will be clear shortly, the terms are properly collected and the associated sub-episodes simultaneously employed in the posterior. We now discuss the computations involved in Theorem 8.

Calculation of $\left\{q_t^k(z_{0:t}^k)\right\}$ We uses the prior of Θ as specified by (52). It is not difficult to verify from (77) that the variational posterior $g(\Theta)$ takes the same form as the prior, i.e.,

$$g(\Theta) \;=\; p(\mu|\widehat{\upsilon})p(\pi|\widehat{\rho})p(W|\widehat{\omega}) \tag{78}$$

where the three factors respectively have the forms of (53),(54), and (55); we have put a hat $\widehat{\ }$ above the hyper-parameters of $g(\Theta)$ to indicate the difference from those of the prior.

Substituting (6) and (78) into (76), we obtain

$$\begin{aligned} & q_t^k(z_{0:t}^k) \\ &= \frac{\tilde{r}_t^k}{C_z}\exp\left\{\sum_{\tau=0}^t\left\langle\ln\pi(z_\tau^k, a_\tau^k)\right\rangle_{p(\pi|\widehat{\rho})} + \left\langle\ln\mu(z_0^k)\right\rangle_{p(\mu|\widehat{\upsilon})} + \sum_{\tau=1}^t\left\langle\ln W(z_{\tau-1}^k, a_{\tau-1}^k, o_\tau^k, z_\tau^k)\right\rangle_{p(W|\widehat{\omega})}\right\} \\ &= \frac{\tilde{r}_t^k}{C_z}\widetilde{\mu}(z_0^k)\widetilde{\pi}(z_0^k, a_0^k)\prod_{\tau=1}^t\widetilde{W}(z_{\tau-1}^k, a_{\tau-1}^k, o_\tau^k, z_\tau^k)\widetilde{\pi}(z_\tau^k, a_\tau^k) \end{aligned} \tag{79}$$

where $\langle\cdot\rangle_{p(\pi|\widehat{\rho})}$ denotes taking expectation with respect to $p(\pi|\widehat{\rho})$, and

$$\begin{aligned}
\widetilde{\mu}(j) &= \exp\left\{\left\langle \ln \mu(j)\right\rangle_{p(\mu|\widehat{v})}\right\} \\
&= \exp\left\{\psi(\widehat{v}_j) - \psi(\sum_{j'=1}^{|\mathcal{Z}|}\widehat{v}_{j'})\right\}, \quad j = 1\ldots|\mathcal{Z}| \qquad (80)\\
\widetilde{\pi}(i,m) &= \exp\left\{\left\langle \ln \pi(i,m)\right\rangle_{p(\pi|\widehat{\rho})}\right\} \\
&= \exp\left\{\psi(\widehat{\rho}_{i,m}) - \psi(\sum_{m'=1}^{|\mathcal{A}|}\widehat{\rho}_{i,m'})\right\}, \quad m = 1\ldots|\mathcal{A}| \qquad (81)\\
\widetilde{W}(i,a,o,j) &= \exp\left\{\left\langle \ln W(i,a,o,j)\right\rangle_{p(W|\widehat{\omega})}\right\} \\
&= \exp\left\{\psi(\widehat{\omega}_{i,a,o,j}) - \psi(\sum_{j'=1}^{|\mathcal{Z}|}\widehat{\omega}_{i,a,o,j'})\right\}, \quad j = 1\ldots|\mathcal{Z}| \qquad (82)
\end{aligned}$$

each of which is a finite set of nonnegative numbers with a sum less than one. Such a finite set is called under-normalized probabilities in (Beal 2003) and used there to perform variational Bayesian learning of hidden Markov models (HMM). The $\psi(\cdot)$ is the digamma function.

It is interesting to note that the product $\widetilde{\mu}(z_0^k)\widetilde{\pi}(z_0^k, a_0^k)\prod_{\tau=1}^{t}\widetilde{W}(z_{\tau-1}^k, a_{\tau-1}^k, o_\tau^k, z_\tau^k)\widetilde{\pi}(z_\tau^k, a_\tau^k)$ on the left side of (79) has exactly the same form as the expression of $p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \Theta)$ in (6), except that the Θ is replaced by $\widetilde{\Theta} = \{\widetilde{\mu}, \widetilde{\pi}, \widetilde{W}\}$. Therefore, one can nominally rewrite (79) as

$$q_t^k(z_{0:t}^k) = \frac{\tilde{r}_t^k}{C_z}p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta}) \qquad (83)$$

with the normalization constant given by

$$C_z = \frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta}) \qquad (84)$$

such that the constraint $\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k) = K$ is satisfied. One may also find that the normalization constant C_z is a nominal empirical value function that has the same form as the empirical value function in (13). The only difference is that the normalized Θ is replaced by the under-normalized $\widetilde{\Theta}$. Therefore, one may write

$$C_z = \widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta}) \qquad (85)$$

Since $\widetilde{\Theta} = \{\widetilde{\mu}, \widetilde{\pi}, \widetilde{W}\}$ are under-normalized, $p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta})$ is not a proper probability distribution. However, one may still write $p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta}) = p(a_{0:t}^k|o_{1:t}^k, \widetilde{\Theta})p(z_{0:t}^k|a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta})$, where $p(a_{0:t}^k|o_{1:t}^k, \widetilde{\Theta}) = \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta})$ and $p(z_{0:t}^k|a_{0:t}^k, o_{1:t}^k, \widetilde{\Theta}) = \frac{p(a_{0:t}^k, z_{0:t}^k|o_{1:t}^k, \widetilde{\Theta})}{p(a_{0:t}^k|o_{1:t}^k, \widetilde{\Theta})}$.

Note that $p(z^k_{0:t}|a^k_{0:t}, o^k_{1:t}, \widetilde{\Theta})$ is a proper probability distribution. Accordingly, $q^k_t(z^k_{0:t})$ can be rewritten as

$$q^k_t(z^k_{0:t}) = \frac{\sigma^k_t(\widetilde{\Theta})}{\widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta})} p(z^k_{0:t}|a^k_{0:t}, o^k_{1:t}, \widetilde{\Theta}) \tag{86}$$

where

$$\begin{aligned} \sigma^k_t(\widetilde{\Theta}) &= \tilde{r}^k_t p(a^k_{0:t}|o^k_{1:t}, \widetilde{\Theta}) \\ &= \tilde{r}^k_t \prod_{\tau=0}^{t} p(a^k_\tau|h^k_\tau, \widetilde{\Theta}) \end{aligned} \tag{87}$$

is called variational re-computed reward, which has the same form as the re-computed reward given in (19) but with Θ replaced by $\widetilde{\Theta}$. The second equality in (87) is based on the equation $p(a^k_{0:t}|o^k_{1:t}, \Theta) = \prod_{\tau=0}^{t} p(a^k_\tau|h^k_\tau, \Theta)$ established in (10) and (11). The nominal empirical value function $\widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta})$ can now be expressed in terms of $\sigma^k_t(\widetilde{\Theta})$,

$$\widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sigma^k_t(\widetilde{\Theta}) \tag{88}$$

Equation (86) shows that $q^k_t(z^k_{0:t})$ is a weighted posterior of $z^k_{0:t}$. The weights, using (88), can be equivalently expressed as $\frac{\sigma^k_t(\widetilde{\Theta})}{\widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta})} = K\eta^k_t(\widetilde{\Theta})$ where

$$\eta^k_t(\widetilde{\Theta}) \stackrel{Def.}{=} \frac{\sigma^k_t(\widetilde{\Theta})}{\sum_{k=1}^{K} \sum_{t=0}^{T_k} \sigma^k_t(\widetilde{\Theta})} \tag{89}$$

The weighted posterior has the same form as (18) used in single-task RPR learning. Therefore we can borrow the techniques developed there to compute the marginal distributions of $p(z^k_{0:t}|a^k_{0:t}, o^k_{1:t}, \widetilde{\Theta})$, particularly those defined in (29) and (30). For clarity, we rewrite these marginal distributions below without re-deriving them, with Θ replaced by $\widetilde{\Theta}$,

$$\xi^k_{t,\tau}(i,j) = p(z^k_\tau = i, z^k_{\tau+1} = j|a^k_{0:t}, o^k_{1:t}, \widetilde{\Theta}) \tag{90}$$

$$\phi^k_{t,\tau}(i) = p(z^k_\tau = i|a^k_{0:t}, o^k_{1:t}, \widetilde{\Theta}) \tag{91}$$

These marginals along with the $\{\eta^k_t(\widetilde{\Theta})\}$ defined in (89) will be used below to compute the variational posterior $g(\Theta)$.

Calculation of the Variational Posterior $g(\Theta)$ To compute $g(\Theta)$, one substitutes (6) and (86) into (77) and performs summation over the latent z variables. Most z variables are summed out, leaving only the marginals in (90) and (91). Employing these marginals and taking into account the weights $\{K\eta^k_t(\widetilde{\Theta})\}$, the variational posterior (with $\eta^k_t(\widetilde{\Theta})$ abbreviated as η^k_t for notational simplicity) is obtained as

$$g(\Theta) = \frac{G_0(\Theta)}{C_\Theta} \exp\left\{ \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} K\eta^k_t \left[\sum_{i=1}^{|\mathcal{Z}|} \phi^k_{t,0}(i) \ln \mu(i) \right. \right.$$

$$
\begin{aligned}
&+\sum_{\tau=1}^{t}\sum_{i=1}^{|\mathcal{Z}|}\phi_{t,\tau}^{k}(i)\ln\pi(i,a_{\tau}^{k})+\sum_{\tau=1}^{t}\sum_{i,j=1}^{|\mathcal{Z}|}\xi_{t,\tau}^{k}(i,j)\ln W(i,a_{\tau-1}^{k},o_{\tau}^{k},j)\Bigg]\Bigg\}\\
= \;& \frac{G_0(\Theta)}{C_\Theta}\prod_{k=1}^{K}\prod_{t=0}^{T_k}\Bigg\{\prod_{i=1}^{|\mathcal{Z}|}\big[\mu(i)\big]^{\eta_t^k\phi_{t,0}^k(i)}\prod_{\tau=1}^{t}\prod_{i=1}^{|\mathcal{Z}|}\big[\pi(i,a_\tau^k)\big]^{\eta_t^k\phi_{t,\tau}^k(i)}\\
&\times\prod_{\tau=1}^{t}\prod_{i,j=1}^{|\mathcal{Z}|}\big[W(i,a_{\tau-1}^k,o_\tau^k,j)\big]^{\eta_t^k\xi_{t,\tau-1}^k(i,j)}\Bigg\}\\
= \;& p(\mu|\widehat{\upsilon})p(\pi|\widehat{\rho})p(W|\widehat{\omega}) \qquad (92)
\end{aligned}
$$

where $p(\mu|\widehat{\upsilon})$, $p(\pi|\widehat{\rho})$, $p(W|\widehat{\omega})$ have the same forms as in (53), (54), and (55), respectively, but with the hyper-parameters replaced by

$$\widehat{\upsilon}_i = \upsilon_i+\sum_{k=1}^{K}\sum_{t=0}^{T_k}\eta_t^k\phi_{t,0}^k(i) \qquad (93)$$

$$\widehat{\rho}_{i,a} = \rho_{i,a}+\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{\tau=0}^{t}\eta_t^k\phi_{t,\tau}^k(i)\delta(a_\tau^k,a) \qquad (94)$$

$$\widehat{\omega}_{i,a,o,j} = \omega_{i,a,o,j}+\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{\tau=1}^{t}\eta_t^k\xi_{t,\tau-1}^k(i,j)\delta(a_{\tau-1}^k,a)\delta(o_\tau^k,o) \qquad (95)$$

for $i,j=1,\ldots,|\mathcal{Z}|$, $a=1,\ldots,|\mathcal{A}|$, $o=1,\ldots,|\mathcal{O}|$. Note that, for simplicity, we have used $\{\widehat{\upsilon},\widehat{\rho},\widehat{\omega}\}$ to denote the hyper-parameters of $g(\Theta)$ for both before and after the updates in (93)-(95) are made. It should be kept in mind that the η's, ϕ's, and ξ's are all based on the numerical values of $\{\widehat{\upsilon},\widehat{\rho},\widehat{\omega}\}$ before the updates in (93)-(95) are made, i.e., they are based on the $\{\widehat{\upsilon},\widehat{\rho},\widehat{\omega}\}$ updated in the previous iteration.

It is clear from (93)-(95) that the update of the variational posterior is based on using all episodes at all time steps (i.e., all sub-episodes). The η_t^k can be thought of as a variational soft count at time t of episode k, which is appended to the hyper-parameters (initial Dirichlet counts) of the prior. Each decision state z receives η_t^k in the amount that is proportional to the probability specified by the posterior marginals $\{\phi_{t,\tau}^k\}$ and $\{\xi_{t,\tau-1}^k\}$.

Computation of the Lower Bound To compute the lower bound $\mathrm{LB}(\{q_t^k\},g(\Theta))$ given in (74), one first takes the logarithm of (76) to obtain

$$
\begin{aligned}
\ln q_t^k(z_{0:t}^k) &= \ln C_z^{-1}\tilde{r}_t^k\exp\left\{\int g(\Theta)\ln p(a_{0:t}^k,z_{0:t}^k|o_{1:t}^k,\Theta)\,d\Theta\right\}\\
&= -\ln C_z+\int g(\Theta)\ln\tilde{r}_t^k p(a_{0:t}^k,z_{0:t}^k|o_{1:t}^k,\Theta)\,d\Theta \qquad (96)
\end{aligned}
$$

which is then substituted into the right side of (A-11) in the Appendix to cancel the first term, yielding

$$
\begin{aligned}
\mathrm{LB}\left(\left\{q_t^k\right\},g(\Theta)\right) &= \ln C_z-\int g(\Theta)\ln\frac{g(\Theta)}{G_0(\Theta)}\,d\Theta\\
&= \ln C_z-\mathrm{KL}\Big(g(\Theta)||G_0(\Theta)\Big)
\end{aligned}
$$

$$= \ln \widehat{V}(\mathcal{D}^{(K)}; \widetilde{\Theta}) - \mathrm{KL}\Big(g(\Theta)||G_0(\Theta)\Big) \tag{97}$$

where the last equality follows from (85).

The lower bound yields a variational approximation to the logarithm of the marginal empirical value. As variational Bayesian learning proceeds, the lower bound monotonically increases, as guaranteed by Theorem 8, and eventually reaches a maxima, at which point one obtains the best (assuming the maxima is global) variational approximation. By taking exponential of the best lower bound, one gets the approximated marginal empirical value. The lower bound also provides a quantitative measure for monitoring the convergence of variational Bayesian learning.

5.5.2 The Complete Gibbs-Variational Learning Algorithm

Algorithmic Description A detailed algorithmic description of the complete Gibbs-variational algorithm is given in Table 2. The algorithm calls the variational Bayesian (VB) algorithm in Table 3 as a sub-routine, to find the variational Bayesian approximations to intractable computations. In particular, the marginal empirical value $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ in (68) is approximated by the exponential of the variational lower bound returned from the VB algorithm by feeding the episodes $\mathcal{D}_m^{(K_m)}$. The conditional posterior $p(\overline{\Theta}_n|\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)}, G_0)$ in (69) is approximated by the variational posterior $g(\overline{\Theta}_n)$ returned from the VB algorithm by feeding the episodes $\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)}$. The variational approximation of $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$ need be computed only once for each environment m, before the main loop begins, since it solely depends on the DP base G_0 and the episodes, which are assumed given and fixed. The variational posterior $g(\overline{\Theta}_n)$ and $\widehat{V}_{G_0}\left(\bigcup_{m\in I_n(c)}\mathcal{D}_m^{(K_m)}\right)$, however, need be updated inside the main loop, because the clusters $\{I_n(c)\}$ keep changing from iteration to iteration.

Upon convergence of the algorithm, one obtains variational approximations to the posteriors of distinct RPRs $\{g(\overline{\Theta}_n)\}_{n=1}^N$, which along with the cluster indicators $\{c_1, c_2, \cdots, c_M\}$ give the variational posterior $g(\Theta_m)$ for each individual environment m. By simple post-processing of the posterior, we obtain the mean or mode of each Θ_m, which gives a single RPR for each environment and yields the history-dependent policy as given by (9). Alternatively, one may draw samples from the variational posterior and use them to produce an ensemble of RPRs for each environment. The RPR ensemble gives multiple history-dependent policies, that are marginalized (averaged) to yield the final choice for the action.

It should be noted that the VB algorithm in Table 3 can also be used as a stand-alone algorithm to find the variational posterior of the RPR of each environment independently of the RPRs of other environments. In this respect the VB is a Bayesian counterpart of the maximum value (MV) algorithm for single-task reinforcement learning (STRL), presented in Section 4 and Table 1.

Time Complexity Analysis The time complexity of the VB algorithm in Table 3 is given as follows where, as in Section 4.3.2, the complexity is quantified by the number of real number multiplications in each iteration and is presented in the Big-O notation. For the reasons stated in Section 4.3.2, the complexity per iteration also represents the complexity of the complete algorithm.

Table 2: The Gibbs-variational algorithm for learning the DP posterior

Input: $\{\mathcal{D}_m^{(K_m)}\}_{m=1}^M$, $\mathcal{A}$, $\mathcal{O}$, $|\mathcal{Z}|$, $\{\upsilon, \rho, \omega\}$, α
Output: $\{\widehat{\upsilon}_n, \widehat{\rho}_n, \widehat{\omega}_n\}_{n=1}^N$ with $N \leq M$ and $\{c_1, c_2, \cdots, c_M\}$.

1. **Computing the variational approximations of** $\{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})\}$**:**
   1.1 **for** $m = 1$ **to** M
   Call the algorithm in Table 3, with the inputs $\mathcal{D}_m^{(K_m)}$, $\mathcal{A}$, $\mathcal{O}$, $|\mathcal{Z}|$, $\{\upsilon, \rho, \omega\}$. Record the returned hyper-parameters as $\{\widehat{\upsilon}_m, \widehat{\rho}_m, \widehat{\omega}_m\}$ and the approximate $\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})$.
2. **Initializations:** Let $j = 1$, N=M, $\ell = 0$.
   Let $\overline{\upsilon}_n = \widehat{\upsilon}_n$, $\overline{\rho}_n = \widehat{\rho}_n$, $\overline{\omega}_n = \widehat{\omega}_n$, for $n = 1, \cdots, N$.
3. **Repeat**
   3.1 **For** $n = 1$ **to** N
   Update $\overline{\Theta}_n$ by drawing from, or finding the mode of, the G_0 with hyper-parameters $\{\overline{\upsilon}_n, \overline{\rho}_n, \overline{\omega}_n\}$.
   3.2 **For** $m = 1$ **to** M
   Let $c_m^{\text{old}} = c_m$ and draw c_m according to (68).
   If $c_m \neq c_m^{\text{old}}$
   If $c_m = 0$, start a new cluster $I_{N+1}(c) = \{m\}$.
   Elseif $c_m \neq 0$, move the element m from $I_{c_m^{\text{old}}}(c)$ to $I_{c_m}(c)$.
   For $n = \{c_m^{\text{old}}, c_m\}$
   If $I_n(c)$ is an empty set
   Delete the n-th cluster.
   Elseif $I_n(c)$ contain a single element (let it be denoted by m')
   Let $\overline{\upsilon}_n = \widehat{\upsilon}_{m'}$, $\overline{\rho}_n = \widehat{\rho}_{m'}$, $\overline{\omega}_n = \widehat{\omega}_{m'}$. Add $\frac{K_{m'}}{\sum_{i=1}^M K_i} \widehat{V}_{G_0}(\mathcal{D}_{m'}^{(K_{m'})})$ to $\ell(j)$.
   Else
   Call the algorithm in Table 3, with the inputs $\bigcup_{i \in I_n(c)} \mathcal{D}_i^{(K_i)}$, $\mathcal{A}$, $\mathcal{O}$, $|\mathcal{Z}|$, $\{\upsilon, \rho, \omega\}$. Record the returned hyper-parameters as $\{\overline{\upsilon}_n, \overline{\rho}_n, \overline{\omega}_n\}$. Scale the returned $\widehat{V}_{G_0}(\bigcup_{i \in I_n(c)} \mathcal{D}_i^{(K_i)})$ by $\frac{\sum_{i \in I_n(c)} K_i}{\sum_{i=1}^M K_i}$ and add the result to $\ell(j)$.
   If $I_n(c)$ is not empty
   Draw $\overline{\Theta}_n$ drawn from G_0 with hyper-parameters $\{\overline{\upsilon}_n, \overline{\rho}_n, \overline{\omega}_n\}$.
   3.3 **Updating** N**:**
   Let N be the number of nonempty clusters and renumber the nonempty clusters so that their indices are in $\{1, 2, \cdots, N\}$.
   3.4 **Convergence check:**
   If the sequence of ℓ converges
   stop the algorithm and exit.
   Otherwise
   Set $j := j + 1$ and $\ell(j) = 0$.

- The computation of $\widetilde{\Theta}$ based on equations (80), (81), and (82) runs in time $O(|\mathcal{Z}|)$, $O(|\mathcal{A}||\mathcal{Z}|)$, and $O(|\mathcal{A}||\mathcal{O}||\mathcal{Z}|^2)$, respectively.

- Computation of the α variables with (35) and (37) (with Θ replaced by $\widetilde{\Theta}$) runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^K T_k)$.

Table 3: The variational Bayesian learning algorithm for RPR

Input: $\mathcal{D}^{(K)}$, $\mathcal{A}$, $\mathcal{O}$, $|\mathcal{Z}|$, $\{\upsilon, \rho, \omega\}$.
Output: $\{\widehat{\upsilon}, \widehat{\rho}, \widehat{\omega}\}$, $\widehat{V}_{G_0}(\mathcal{D}^{(K)}) \approx \mathrm{LB}(\{q_t^k\}, g(\Theta))$.

1. **Initialize** $\widehat{\upsilon} = \upsilon$, $\widehat{\rho} = \rho$, $\widehat{\omega} = \omega$, $\ell = [\,]$, iteration $= 1$.
2. **Repeat**
   - 2.1 **Computing $\widetilde{\Theta}$:**
     Compute the set of under-normalized probabilities $\widetilde{\Theta}$ using equations (80)(81)(82).
   - 2.2 **Dynamical programming:**
     Compute α and β variables with (35)(36)(37), with Θ replaced by $\widetilde{\Theta}$ in these equations.
   - 2.3 **Reward re-computation:**
     Calculate the variational recomputed reward $\{\sigma_t^k(\widetilde{\Theta})\}$ using (87)(37) and compute the weight $\{\eta_t^k(\widetilde{\Theta})\}$ using (89).
   - 2.4 **Lower bound computation:**
     Calculate the variational lower bound $\mathrm{LB}(\{q_t^k\}, g(\Theta))$ using (97)(88).
   - 2.5 **Convergence check:**
     Let $\ell(\text{iteration}) = \mathrm{LB}(\{q_t^k\}, g(\Theta))$.
     If the sequence of ℓ converges
     Stop the algorithm and exit.
     Else
     Set iteration := iteration + 1.
   - 2.6 **Posterior update for z:**
     Compute the ξ and ϕ variables using equations (33)(34).
   - 2.7 **Update of hyper-paramters:**
     Compute the updated $\{\widehat{\upsilon}, \widehat{\rho}, \widehat{\omega}\}$ using (93)(94)(95).

- Computation of the β variables with (36) and (37) (with Θ replaced by $\widetilde{\Theta}$) runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} \sum_{t=0, r_t^k \neq 0}^{T_k} (t+1))$, which is $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k^2)$ in the worst case and is $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k)$ in the best case, where the worst and best cases are distinguished by the sparseness of immediate rewards, as discussed in Section 4.3.2.
- The reward re-computation using (87), (37), and (89) requires time $O(\sum_{k=1}^{K} T_k)$ in the worst case and $O(K)$ in the best case.
- Computation of the lower bound using (88) and (97) requires time $O(|\mathcal{A}||\mathcal{O}||\mathcal{Z}|^2)$.
- Update of the hyper-parameters using (93), (94), and (95), as well as computation of the ξ and ϕ variables using (33) and (34), runs in time $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k^2)$ in the worst case and $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k)$ in the best case.

The overall complexity of the VB algorithm is seen to be $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k^2)$ in the worst case and $O(|\mathcal{Z}|^2 \sum_{k=1}^{K} T_k)$ in the best case, based on the fact that $\sum_{k=1}^{K} T_k \gg |\mathcal{A}||\mathcal{O}|$ in general. Thus the VB algorithm has the same time complexity as the value maximization algorithm in Table 1. Note that the time dependency on the lengths of episodes is dictated by the sparseness of the immediate rewards; for most problems considered in Section 6, the agent

receives rewards only when the goal state is reached, in which case the VB algorithm scales linearly with the lengths of episodes.

The complexity of the Gibbs-variational algorithm can be easily obtained based on the complexity analysis above for the VB algorithm. At the beginning and before entering the main loop, the Gibbs-variational algorithm calls the VB to compute the variational approximation of the marginal empirical value $\{\widehat{V}_{G_0}(\mathcal{D}_m^{(K_m)})\}$ for each environment m, by feeding the associated episodes $\mathcal{D}_m^{(K_m)}$. These computations are performed only once. For each environment the VB runs until convergence, with a time complexity between $O(|\mathcal{Z}|^2 \sum_{k=1}^{K_m} T_{m,k})$ and $O(|\mathcal{Z}|^2 \sum_{k=1}^{K_m} T_{m,k}^2)$ per iteration, depending on the sparseness of the immediate rewards. Inside the main loop, the Gibbs-variational algorithm calls the VB to compute the variational posterior of distinct RPR for each cluster n, by feeding the merged episodes $\bigcup_{m\in I_n(c)} \mathcal{D}_m^{(K_m)}$. These computations are performed each time the clusters are updated, with a time complexity between $O(|\mathcal{Z}|^2 \sum_{m\in I_n(c)} \sum_{k=1}^{K_m} T_{m,k})$ and $O(|\mathcal{Z}|^2 \sum_{m\in I_n(c)} \sum_{k=1}^{K_m} T_{m,k}^2)$ per iteration for cluster n.

6. Experimental Results

We compare the performance of RPR in multi-task reinforcement learning (MTRL) versus single-task reinforcement learning (STRL), and demonstrate the advantage of MTRL. The RPR for MTRL is implemented by the Gibbs-variational algorithm in Table 2 and the RPR for STRL is implemented by the maximum-value (MV) algorithm in Table 1. The variational Bayesian (VB) algorithm in Table 3, which is a Bayesian counterpart of the MV algorithm, generally performs similar to the MV for STRL and is thus excluded in the comparisons.

Since the MV algorithm is a new technique developed in this paper, we evaluate the performance of the MV before proceeding to the comparison of MTRL and STRL. We also compare the MV to the method of first learning a POMDP model from the episodes and then finding the optimal policy for the POMDP.

6.1 Performance of RPR in Single-Task Reinforcement Learning (STRL)

We consider the benchmark example Hallway2, introduced in (Littman et al. 1995). The Hallway2 problem was originally designed to test algorithms based on a given POMDP model, and it has recently been employed as a benchmark for testing model-free reinforcement algorithms (Bakker 2004; Wierstra and Wiering 2004).

Hallway2 is a navigation problem where an agent is situated in a maze consisting of a number of rooms and walls that are connected to each other and the agent navigates in the maze with the objective of reaching the goal within a minimum number of steps. The maze is characterized by 92 states, each representing one of four orientations (south, north, east, west) in any of 23 rectangle areas, and four of the states (corresponding to a single rectangle area) represent the goal. The observations consist of $2^4 = 16$ combinations of presence/absence of a wall as viewed when standing in a rectangle facing one of the four orientations, and there is an observation uniquely associated with the goal. There are five different actions that the agent can take: {*stay in place*, *move forward*, *turn right*, *turn left*, *turn around*}. Both state transitions and observations are very noisy (uncertain), except

that the goal is fully identified by the unique observation associated with it. The reward function is defined in such a way that a large reward is received when the agent enters the goal from the adjacent states, and zero reward is received otherwise. Thus the reward structure is highly sparse and both the MTRL and STRL algorithms scale linearly with the lengths of episodes in this case, as discussed in Sections 4.3.2 and 5.5.2.

6.1.1 Performance as a Function of Number of Episodes

We investigate the performance of the RPR, as a function of K the number of episodes used in the learning. For each given K, we learn a RPR from a set of K episodes $\mathcal{D}^{(K)}$ that are generated by following the behavior policy Π, and the learning follows the procedures described in Section 4.

The conditions for the policy Π, as given in Theorem 5, are very mild, and are satisfied by a uniformly random policy. However, a uniformly random agent may take a long time to reach the goal, which makes the learning very slow. To accelerate learning, we use a semi-random policy Π, which is simulated by the rule that, with probability p_{query}, Π chooses an action suggested by the PBVI algorithm (Pineau et al. 2003) and, with probability $1 - p_{\text{query}}$, Π chooses an action uniformly sampled from $\mathcal{A}$. The use of PBVI here is similar to the meta-queries used in (Doshi et al. 2008), where a meta-query consults a domain expert (who is assumed to have access to the true POMDP model) for the optimal action at a particular time step. The meta-queries correspond to human-robot interactions in robotics applications. It should be noted that, by implementing the reward re-computation in RPR online, the behavior policy in each iteration simply becomes the RPR in the previous iteration, in which case the use of an external policy like PBVI is eliminated.

In principle, the number of decision states (belief regions) $|\mathcal{Z}|$ can be selected by maximizing the marginal empirical value $\widehat{V}_{G_0}(\mathcal{D}^{(K)}) = \int \widehat{V}(\mathcal{D}^{(K)}) G_0(\Theta) d\Theta$ with respect to $|\mathcal{Z}|$, where an approximation of $\widehat{V}(\mathcal{D}^{(K)})$ can be found by the VB algorithm in Table 3. Because the MV does not employ a prior, we make a nominal prior $G_0(\Theta)$ by letting it take the form of (52) but with all hyper-parameters uniformly set to one. This leads to $G_0(\Theta) \equiv C_{\text{mv}}$, where C_{mv} is a normalization constant. Therefore maximization of $\widehat{V}_{G_0}(\mathcal{D}^{(K)})$ is equivalent to maximization of $\int \widehat{V}(\mathcal{D}^{(K)}; \Theta) d\Theta$, which serves as an evidence of how good the choice of $|\mathcal{Z}|$ fits to the episodes in terms of empirical value. According to the Occam Razor principle (Beal 2003), the minimum $|\mathcal{Z}|$ fitting the episodes has the best generalization. In practice, letting $|\mathcal{Z}|$ be a multiple of the number of actions is usually a good choice (here $|\mathcal{Z}| = 4 \times 5 = 20$) and we find that the performance of the RPR is quite robust to the choice of $|\mathcal{Z}|$. This may be attributed to the fact that learning of the RPR is a process of allocating counts to the decision states — when more decision states are included, they simply share the counts that otherwise would have been allocated to a single decision state. Provided the sharing of counts is consistent among μ, π, and W, the policy will not change much.

The performance of the RPR is compared against EM-PBVI, the method that first learns a predictive model as in (Chrisman 1992) and then learns the policy based on the predictive model. Here the predictive model is a POMDP learned by expectation maximization (EM) based on $\mathcal{D}^{(K)}$ and the PBVI (Pineau et al. 2003) is employed to find the policy given the POMDP. To examine the effect of the behavior policy Π on the RPR's performance, we

consider three different Π's, which respectively have a probability $p_{\text{query}} = 5\%, 30\%, 50\%$ of choosing the actions suggested by PBVI, where p_{query} corresponds to the rate of meta-query in (Doshi et al. 2008). The episodes used to learn EM-PBVI are collected by the behavior policy with $p_{\text{query}} = 50\%$, which is the highest query rate considered here. Therefore the experiments are biased favorably towards the EM-PBVI, in terms of the number of expert-suggested actions that are employed to generate the training episodes.

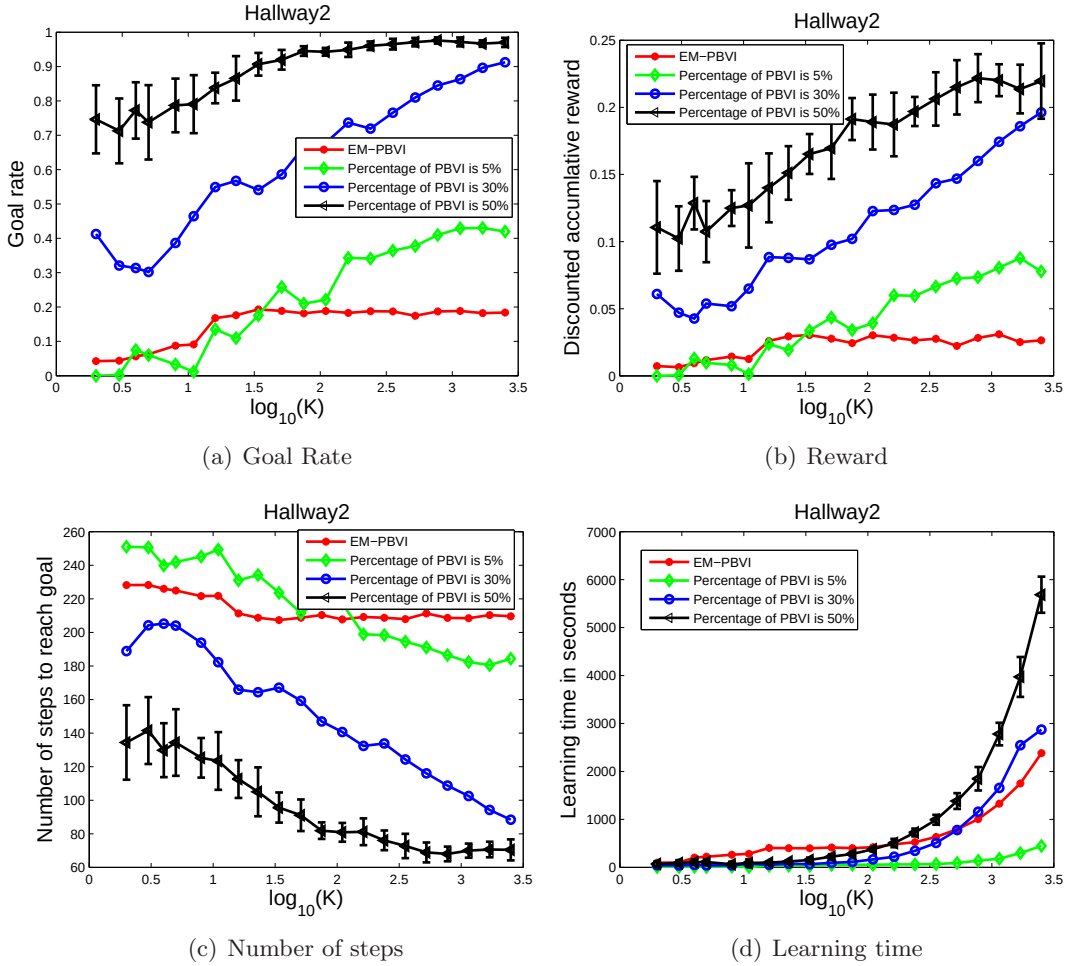
The performance of each RPR, as well as that of EM-PBVI, is evaluated on Hallway2 by following the standard testing procedure as set up in (Littman et al. 1995). For each policy, a total of 251 independent trials are performed and each trial is terminated when either the goal is reached or a maximum budget of 251 steps is consumed. Three performance measures are computed based on the 251 trials: (a) the discounted accumulative reward (i.e., the sum of exponentially discounted rewards received over the $N_{\text{te}} \leq 251$ steps) averaged over the 251 trials; (b) the goal rate, i.e., the percentage of the 251 trials in which the agent has reached the goal; (c) the number of steps that the agent has actually taken, averaged over the 251 trials.

The results on Hallway2 are summarized in Figure 1, where we present each of the three performance measures plus the learning time, as a function of $\log_{10}$ of the number of episodes K used in the learning. The four curves in each figure correspond to the EM-PBVI, and the three RPRs with a rate of PBVI query 5%, 30%, and 50%, respectively. Each curve results from an average over 20 independently generated $\mathcal{D}^{(K)}$ and the error bars show the standard deviations. For simplicity, the error bars are shown only for the RPR with a 50% query rate.

As shown in Figure 1 the performance of the RPR improves as the number of episodes K used to learn it increases, regardless of the behavior policy Π. As recalled from Theorem 5, the empirical value function $\widehat{V}(\mathcal{D}^{(K)}; \Theta)$ approaches the exact value function as K goes to infinity. Assuming the RPR has enough memory (decision states) and the algorithm finds the global maxima, the RPR will approach the optimal policy as K increases. Therefore, Figure 1 serves as an experimental verification of Theorem 5. The CPU time shown in Figure 1(d) is exponential in $\log_{10} K$ or, equivalently, is linear in K. The linear time is consistent with the complexity analysis in Section 4.3.2.

The error bars of goal rate exhibits quick shrinkage with K and those of the median number of steps also shrinks relatively fast. In contrast, the discounted accumulative reward has a very slow shrinkage rate for its error bars. The different shrinkage rates show that it is much easier to reach the goal within the prescribed number of steps (251 here) than to reach the goal in relatively less steps. Note that, when the goal is reached at the t-th step, the number of steps is t but the discounted accumulative reward is $\gamma^{-t} r_{\text{goal}}$, where r_{goal} is the reward of entering the goal state. The exponential discounting explains the difference between the number of steps and the discounted accumulative reward regarding the shrinkage rates of error bars.

A comparison of the three RPR curves in Figure 1 shows that the rate at which the behavior policy Π uses or queries PBVI influences the RPR's performance and the influence depends on K. When K is small, increasing the query rate significantly improves the performance; whereas, when K gets larger, the influence decreases until it eventually vanishes. The decreased influence is most easily seen between the curves of 30% and 50% query rates. To make the performance not degrade when the query rate decreases to as low as 5%, a



(a) Goal Rate (b) Reward

(c) Number of steps (d) Learning time

Figure 1: Performance comparison on the Hallway2 problem. The horizontal axis is $\log_{10}$ of the number of episodes K used in learning the RPR. The horizontal axis in each sub-figure is (a) Goal rate (b) Discounted accumulative reward (c) Number of steps to reach the goal (d) Time in seconds for learning the RPR. The four curves in each figure correspond to the EM-PBVI and the RPR based on a behavior policy Π that queries PBVI with a probability $p_{\text{query}} = 5\%$, 30%, 50%, respectively. The EM-PBVI employs EM to learn a POMDP model based on the episodes collected by Π with $p_{\text{query}} = 50\%$ and then uses the PBVI (Pineau et al. 2003) to find the optimal policy based on the learned POMDP. Each curve results from an average over 20 independent runs and, for simplicity, the error bars are shown only for the RPR with a 50% query rate. The performance measures in (a)-(c) are explained in greater detail in text.

much larger K may be required. These experimental results confirm that random actions can accomplish a good exploration of available rewards (the goal state here) by collecting a large number of (lengthy) episodes and the RPR learned from these episodes perform competitively. With a small number of episodes, however, random actions achieve limited

exploration and the resulting RPR performs poorly. In the latter case, queries to experts like PBVI plays an important role in improving the exploration and the RPR's performance.

It is also seen from Figure 1 that the performance of EM-PBVI is not satisfactory and grows slowly with K. The poor performance is strongly related to insufficient exploration of the environment by the limited episodes. For EM-PBVI, the required amount of episodes is more demanding because the initial objective is to build a POMDP model instead of learning a policy. This is because policy learning is concerned with exploring the reward structure but building a POMDP requires exploration of all aspects of the environment. This demonstrates the drawback of methods that rely on learning an intervening POMDP model, with which a policy is designed subsequently.

6.2 Performance of RPR in Multi-task Reinforcement Learning (MTRL)

6.2.1 Maze Navigation

Problem Description In this problem, there are $M = 10$ environments and each environment is a grid-world, i.e., an array of rectangular cells. Of the ten environments, three are distinct and are shown in Figure 2, the remaining are duplicated from the three distinct ones. Specifically, the first three environments are duplicated from the first distinct one, the following three environments are duplicated from the second distinct one, and the last four environments are duplicates from the third distinct one. We assume 10 sets of episodes, with the m-th set collected from the m-th environment.

In each of the distinct environments shown in Figure 2, the agent can take five actions {*move forward*, *move backward*, *move left*, *move right*, *stay*}. In each cell of the grid-world environments, the agent can only observe the openness of the cell in the four directions. The agent then has a total of 16 possible observations indicating the $2^4 = 16$ different combinations of the openness of a cell in the four orientations. The actions (except the action *stay*) taken by the agent are not accurate and have some noise. The probability of arriving at the correct cell by taking a *move* action is 0.7 and the probability of arriving at other neighboring cells is 0.3. The perception is noisy, with a probability 0.8 of correctly observing the openness and the probability 0.2 of making a mistaken observation. The agent receives a unit reward when the goal (indicated by a basket of food in the figures) is reached and zero reward otherwise. The agent does not know the model of any of the environments but only has access to the episodes, i.e., sequences of actions, observations, and rewards.

Algorithm Learning and Evaluation For each environment $m = 1, 2, \cdots, 10$, there is a set of K episodes $\mathcal{D}_m^{(K)}$, collected by simulating the agent-environment interaction using the models described above and a behavior policy Π that the agent follows to determine his actions. The behavior policy is the semi-random policy described in Section 6.1, with a probability $p_{\text{query}} = 0.5$ of taking the actions suggested by PBVI.

Reinforcement learning (RL) based on the ten sets of episodes $\{\mathcal{D}_m^{(K)}\}_{m=1}^{10}$ leads to ten RPRs, each associated with one of the ten environments. We consider three paradigms of learning: the MTRL in which the Gibbs-variational algorithm in Table 2 is applied to the ten sets of episodes jointly, the STRL in which the MV algorithm in Table 1 is applied to each of the ten episode sets separately, and pooling in which the MV algorithm is applied

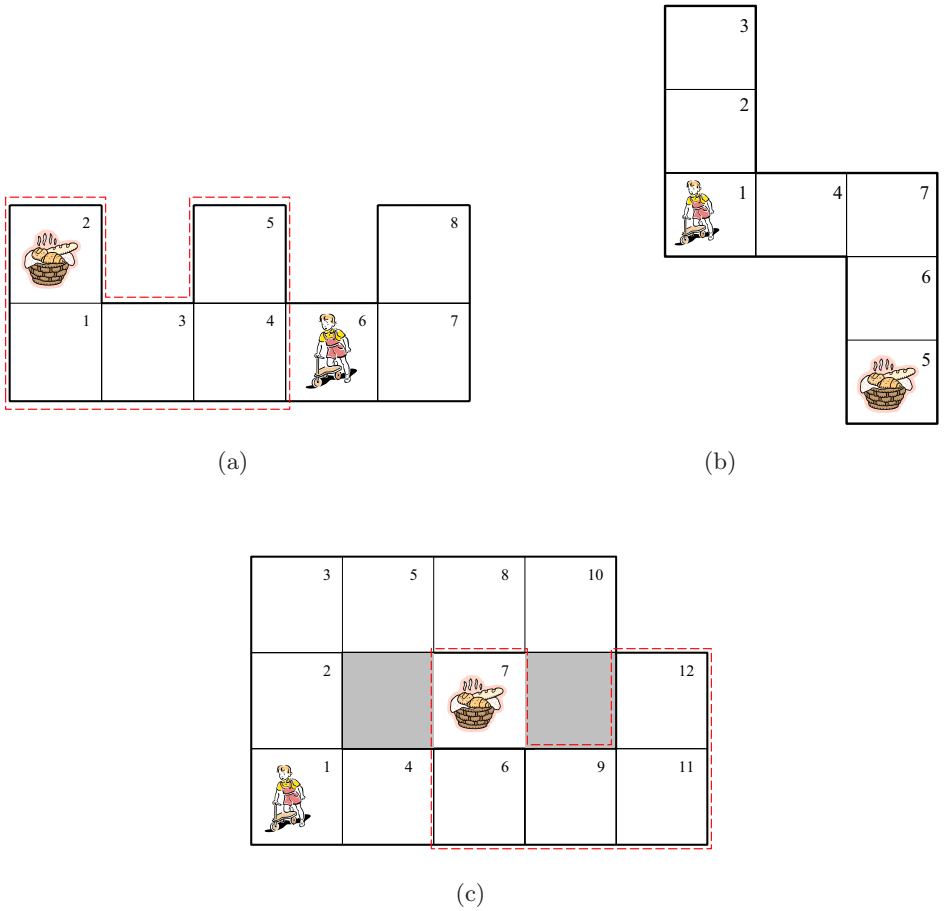


Figure 2: The three distinct grid-world environments, where the goal is designated by the basket of food, each block indicates a cell in the grid world, and the two gray cells are occupied by a wall. The red dashed lines in (a) and (c) indicate the *similar* parts in the two environments. The agent locates himself by observing the openness of a cell in the four orientations. Both the motion and the observation are noisy.

to the union of the ten episode sets. The number of decision states is chosen as $|\mathcal{Z}| = 6$ for all environments and all learning paradigms. Other larger $|\mathcal{Z}|$ give similar results and, if desired, the selection of decision states can be accomplished by maximizing the marginal empirical value with respect to $|\mathcal{Z}|$, as discussed above.

The RPR policy learned by any paradigm for any environment is evaluated by executing the policy 1000 times independently, each time starting randomly from a grid cell in the environment and taking a maximum of 15 steps. The performance of the policy is evaluated by two performance measures: (a) the average success rate at which the agent reaches the goal within 15 steps, and (b) the average number of steps that the agent takes to reach the goal. When the agent does not reach the goal within 15 steps, the number of steps is 15. Each performance measure is computed from the 1000 instances of policy execution, and is averaged over 20 independent trials.

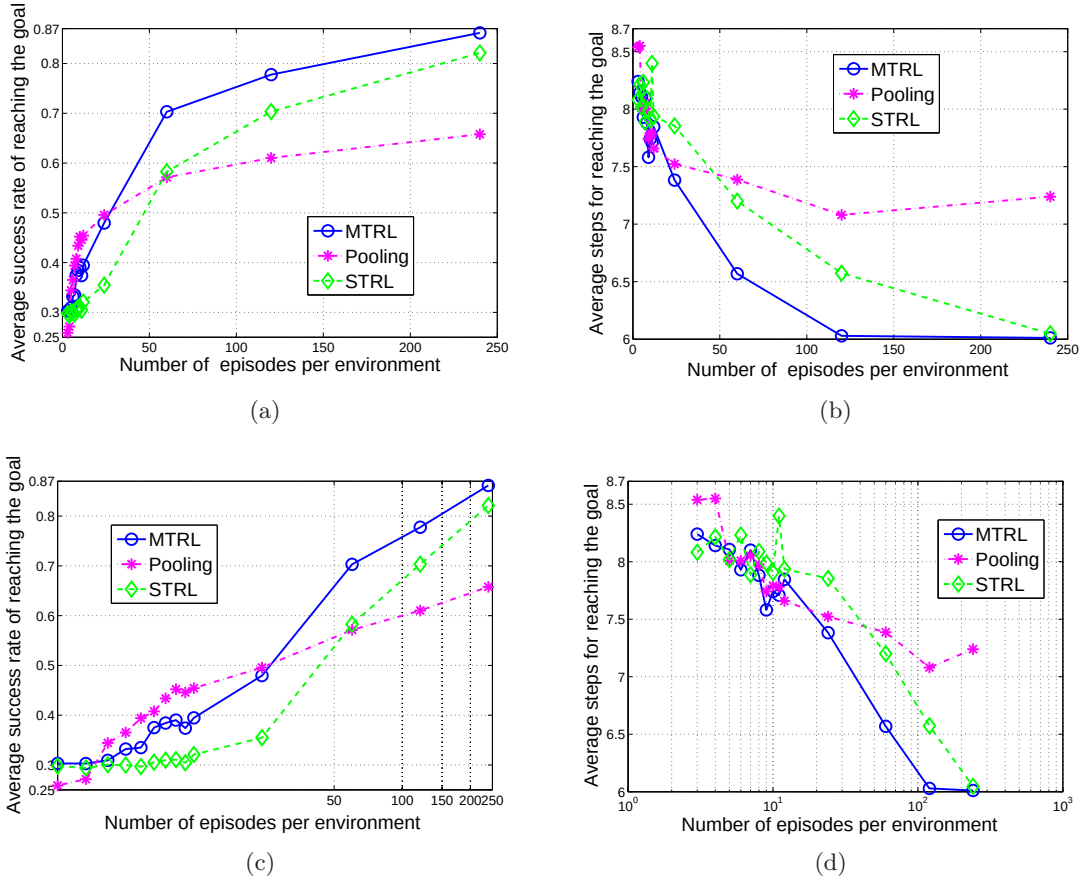


Figure 3: Comparison of MTRL, STRL, and pooling on the problem of multiple stochastic environments summarized in Figure 2. (a) Average success rate for the agent to reach the goal within 15 steps. (b) Average step for the agent reaching the target. (c) Average success rate for the agent with the horizontal axis in log scale.(d) Average step with the horizontal axis in log scale.

We examine the performance of each learning paradigm for various choices of K, the number of episodes per environment. Specifically we consider 16 different choices: $K = 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 60, 120, 240$. The performances of the three learning paradigms, averaged over 20 independent trials, are plotted in Figure 3 as a function of K. Figures 3(c) and 3(d) are respectively duplicates of Figures 3(a) and 3(b), with the horizontal axis displayed in a logarithmic scale. By (68), the choice of the precision parameter α in Dirichlet process influences the probability of sampling a new cluster; it hence influences the resulting number of distinct RPR parameters $\overline{\Theta}$. According to (West 1992), the choice of α is governed by the posterior $p(\alpha|K, N) \propto p(N|K, \alpha)p(\alpha)$, where N is the number of clusters updated in the most recent iteration of the Gibbs-variational algorithm. One may choose α by sampling from the posterior or finding the mean. When K is large and $N \ll K$ and the prior $p(\alpha)$ is a Gamma distribution, the posterior $p(\alpha|K, N)$ is approximately a Gamma distribution with the mean $\mathbb{E}(\alpha) = O(N \log(K))$. For the different choices of K considered

above, we choose $\alpha = 3n$, with $n = 2, 3, \ldots, 15$ respectively. These choices are based on approximations of $\mathbb{E}(\alpha)$ obtained by fixing N at an initial guess $N = 8$. We find that the results are relatively robust to the initial guess provided the logarithmic dependence on K is employed. The density of the DP base G_0 is of the form in (52), with all hyper-parameters set to one, making the base non-informative.

Figures 3(a) and 3(b) show that the performance of MTRL is generally much better than that of STRL and pooling. The improvement is attributed to the fact that MTRL automatically identifies and enforces appropriate sharing among the ten environments to ensure that information transfer is positive. The improvement over STRL indicates that the number of episodes required for finding the correct sharing is generally smaller that that required for finding the correct policies.

The identification of appropriate sharing is based on information from the episodes. When the number of episodes is very small (say, less than 25 in the examples here), the sharing found by MTRL may not be accurate; in this case, simply pooling the episodes across all ten environments may be a more reasonable alternative. When the number of episodes increases, however, pooling begins to show disadvantages since the environments are not all the same and forcing them to share leads to negative information transfer. The seemingly degraded performance of pooling at the first two points in Figure 3(c) may not be accurate since the results have large variations when the episodes are extremely scarce; much more Monte Carlo runs may be required to obtain accurate results in these cases.

The performance of STRL is poor when the number of episodes is small, because a small set of episodes do not provide enough information for learning a good RPR. However, the STRL performance improves significantly with the increase of episodes, which whittles down the advantage brought about by information transfer and allows STRL to eventually catch up with MTRL in performance.

Analysis of the Sharing Mechanism We investigate the sharing mechanism of the MTRL by plotting Hinton diagrams. The Hinton diagram (Hinton and Sejnowski 1986) is a quantitative way of displaying the elements of a data matrix. Each element is represented by a square whose size is proportional to the magnitude. In our case here, the data matrix is the between-task similarity matrix (Xue et al. 2007) learned by the MTRL; it is defined as follows: the between-task similarity matrix is a symmetric matrix of size $M \times M$ (where M denotes the number of tasks and $M = 10$ in the present experiment), the (i, j)-th element measuring the frequency that task i and task j belong to the same cluster (i.e., they result in the same distinct RPR). In order to avoid the bias due to any specific set of episodes, we perform 20 independent trials and average the similarity matrix over the 20 trials. In each trial, if tasks i and j belong to one cluster upon convergence of the Gibbs-variational algorithm, we add one at (i, j) and (j, i) of the matrix. We compute the between-task similarity matrices when the number of episodes is respectively $K = 3, 10, 60, 120$, which represent the typical sharing patterns inferred by the MTRL for the present maze navigation problem. The Hinton diagrams for these four matrices are plotted in Figure 4.

The Hinton diagrams in Figures 4(a) and 4(b) show that when the number of episodes is small, environments 1, 2, 3, 7, 8, 9, 10 have a higher frequency of sharing the same RPR. This sharing can be intuitively justified by first recalling that these environments are duplicates of Figures 2(a) and 2(c), and then noting that the parts circumscribed by red

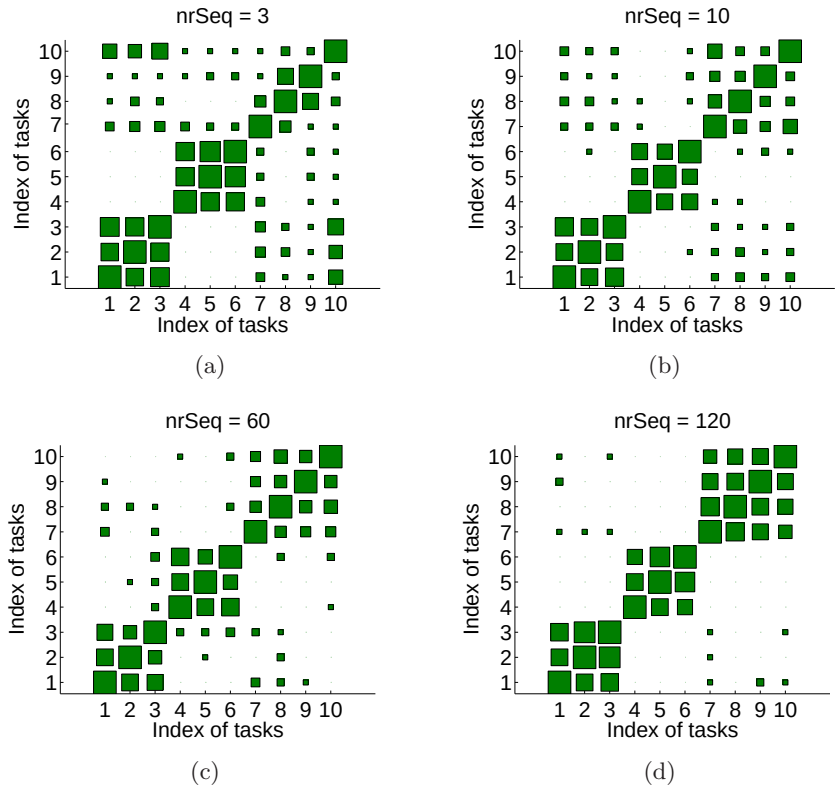


Figure 4: Hinton diagrams of the between-task similarity matrix inferred by the MTRL for the problem of multiple stochastic environments 2. The number of episodes per environment is (a) 3 (b) 10 (c) 60 (d) 120.

dashed lines in Figures 2(a) and 2(c) are quite similar. Meanwhile the Hinton diagrams also show a weak sharing between environments 4,5,6,7,8,9,10, which are duplicates of Figures 2(b) and 2(c). This is probably because the episodes are very few at this stage, and pooling episodes from environments that are not so relevant to each other could also be helpful. This explains why, in Figure 3(a), the performance of pooling is as good as that of the MTRL when the number of episodes is small.

As the number of episodes progressively increases, the ability of MTRL to identify the correct sharing improves and, as seen in Figures 4(b) and 4(c), only those episodes from relevant environments are pooled together to enhance the performance — a simple pooling of all episodes together deteriorates the performance. This explains why the MTRL outperforms pooling with the increase of episodes. Meanwhile, the STRL does not perform well for limited episodes. However, when there are more episodes from each environment, the STRL learns and performs steadily better until it outperforms the pooling and becomes comparable to the MTRL.

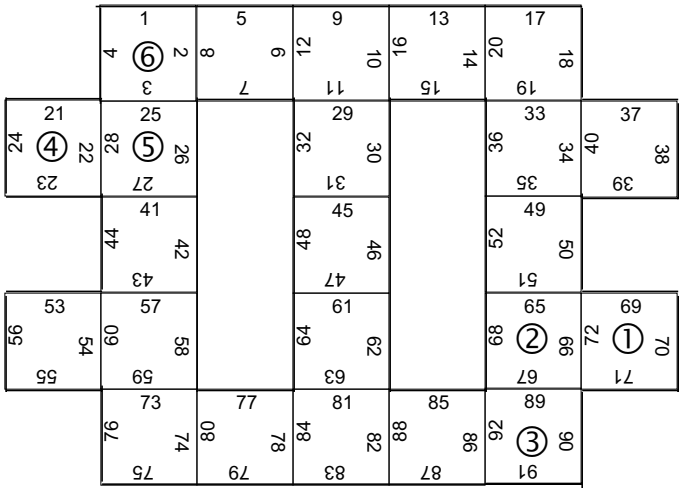
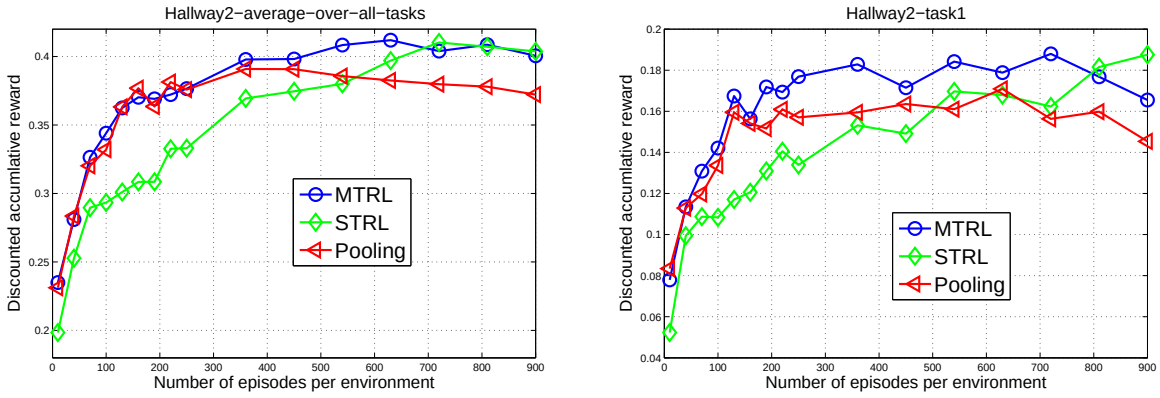


Figure 5: Displacements of goal state in the six environments considered in Maze Navigation 2. Each environment is a variant of the benchmark Hallway2 (Littman et al. 1995) with the goal displaced to a new grid cell designated by a numbered circle and the number indicating the index of the environment. The unique observation associated with the goal is also changed accordingly in each variant.



(a) (b)

Figure 6: Performance comparison on the six environments modified from the benchmark problem Hallway2 (Littman et al. 1995). (a) Discounted accumulative reward averaged over the six environments (b) Discounted accumulative reward in the first environment, which is the original Hallway2.

6.2.2 Maze Navigation 2

We consider six environments, each of which results from modifying the benchmark maze problem Hallway2 (Littman et al. 1995) in the following manner. First the goal state is displaced to a new grid cell and then the unique observation associated with the goal is

changed accordingly. For each environment the location of the goal state is shown in Figure 5 as a numbered circle, where the number indicates the index of the environment. Of the six environments the first one is the original Hallway2. It is seen that environments $1, 2, 3$ have their goal states near the lower right corner while environments $4, 5, 6$ have their goal states near the upper left corner. Thus we expect that the environments are grouped into two clusters.

For each environment, a set of K episodes are collected by following a semi-random behavior policy Π that executes the actions suggested by PBVI with probability $p_{\text{query}} = 0.3$. As in Section 6.2.1 three versions of RPR are obtained for each environment, based respectively on three paradigms, namely MTRL, STRL, and pooling. The α is chosen as $5\log(K)$ with 5 corresponding to an initial guess of N and G_0 is of the form of (52) with all hyper-parameters close to one (thus the prior is non-informative). The number of decision states is $|\mathcal{Z}| = 20$ as in Section 6.1.1. The performance comparison, in terms of discounted accumulative reward and averaged over 20 independent trials, is summarized in Figure 6, as a function of the number of episodes per environment.

Figure 6(a) shows that the MTRL maintains the overall best performance regardless of the number of episodes K. The STRL and the pooling are sensitive to K, with the pooling outperforming the STRL when $K < 540$ but outperformed by the STRL when $K > 540$. In either case, however, the MTRL performs no worse than both. The MTRL consistently performs well because it adaptively adjusts the sharing among tasks as K changes, such that the sharing is appropriate regardless of K. The adaptive sharing can be seen from Figure 7, which shows the Hinton diagram of the between-task similarity matrix learned by the MTRL, for various instances of K. When K is small there is a strong sharing among all tasks, in which case the MTRL reduces to the pooling, explaining why the MTRL performs similar to the pooling when $K \leq 250$. When K is large, the sharing becomes weak between any two tasks, which reduces the MTRL to the STRL, explaining why the two perform similarly when $K \geq 700$. As the number of episodes approaches to $K = 540$, the performances of the STRL and the pooling tend to become closer and more comparable until they eventually meet at $K = 540$. The range of K near this intersection is also the area in which the MTRL yields the most significant margin of improvements over the STRL and the pooling. This is so because, for this range of K, the correct between-task sharing is complicated (as shown in Figure 7(b)), which can be accurately characterized by the fine sharing patterns provided by the MTRL, but cannot be characterized by the pooling or the STRL.

Figure 6(a) plots the overall performance comparison taking all environments into consideration. As an example of the performances in individual environments, we show in Figure 6(a) the performance comparison in the first environment, which is also the original Hallway2 problem. The change of magnitude in the vertical axis is due to the fact the first environment has the goal in a room (instead in the hallway), which makes it more difficult to reach the goal.

6.2.3 Multi-aspect Classification

Problem Description Multi-aspect classification refers to the problem of identifying the class label of an object using observations from a sequence of viewing angles. This

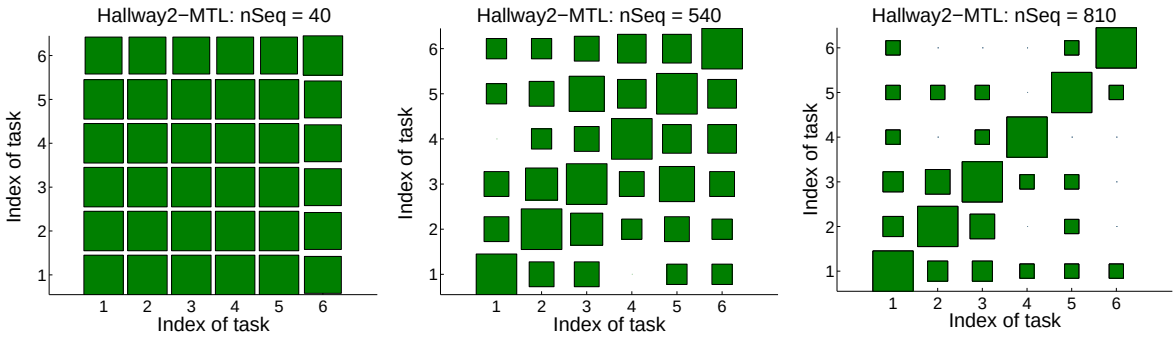


Figure 7: Hinton diagrams of the between-task similarity matrix learned by the MTRL from the six environments modified from the benchmark problem Hallway2 (Littman et al. 1995). The number of episodes is is (a) $K = 40$ (b) $K = 540$ (c) $K = 810$.

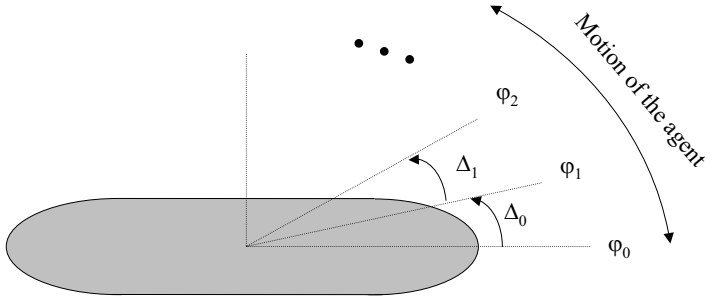


Figure 8: A typical configuration of multi-aspect classification of underwater objects.

problem is generally found in applications where the object responds to interrogations in a angle-dependent manner. In such cases, an observation at a single viewing angle carries the information specific to only that angle and the nearby angles, and one requires observations at many viewing angles to fully characterize the object.

More importantly, the observations at different viewing angles are not independent of each other, and are correlated in a complicated and yet useful way. The specific form of the angle-dependency is dictated by the physical constitution of the object as well as the nature of the interrogator — typically electromagnetic or acoustic waves. By carefully collecting and processing observations sampled at densely spaced angles, it is possible to form an image, based on which classification can be performed. An alternative approach is to treat the observations as a sequence and characterize the angle-dependency by a hidden Markov model (HMM) (Runkle et al. 1999).

In this section we consider multi-aspect classification of underwater objects based on acoustic responses of the objects. Figure 8 shows a typical configuration of the problem. The cylinder represents an underwater object of unknown identity y. We assume that the object belongs to a finite set of categories $\mathcal{Y}$ (i.e., $y \in \mathcal{Y}$). The agent aims to discover the

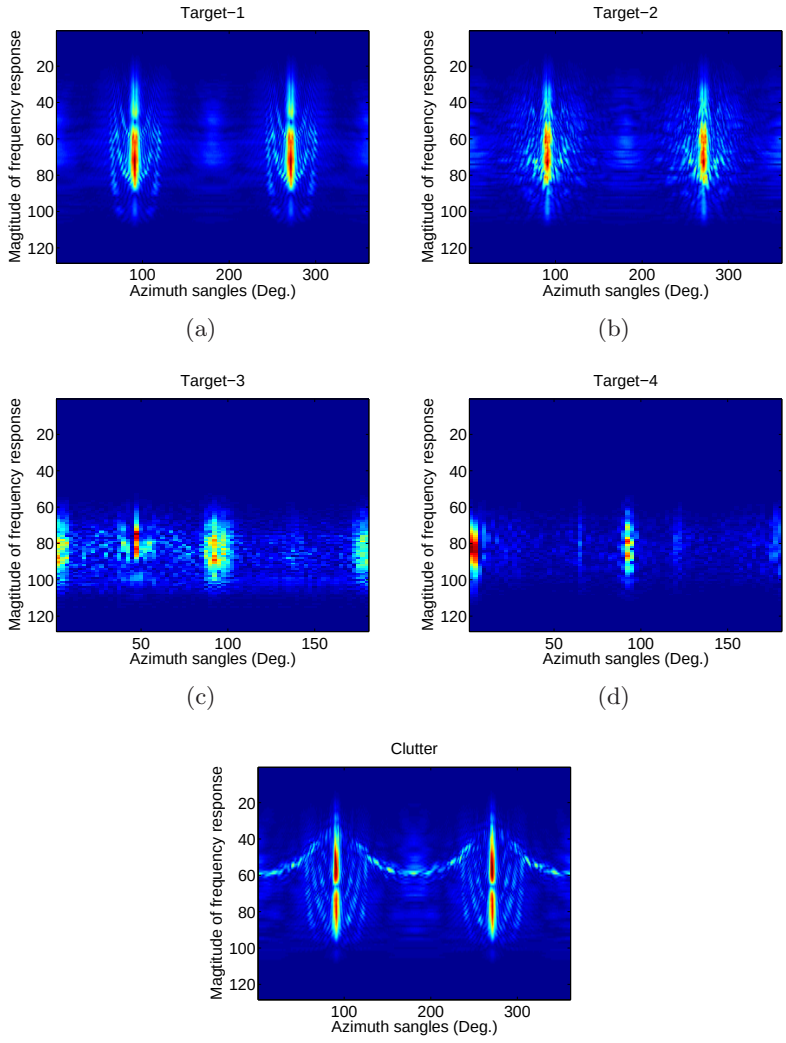


Figure 9: Frequency-domain acoustic responses of the five underwater objects (a) Target-1 (b) Target-2 (c) Target-3 (d) Target-4 (e) Clutter.

unknown y by moving around the object and interrogating it at multiple viewing angles φ. We assume the angular motion is one-dimensional, i.e., the agent moves clockwise or counterclockwise on the page, but does not move out of the page. The set of angles that can be occupied by the agent is then $[0°, 360°]$, which in practice is discretized into a finite number of angular sectors denoted by $\mathcal{S}_\varphi$.

In the HMM approach (Runkle et al. 1999), $\mathcal{S}_\varphi$ constitutes the set of hidden states, and the state transitions can be computed using simple geometry (Runkle et al. 1999), under the assumptions that each time the agent moves by a constant angular step and that the specific angles occupied by the agent are uniformly distributed within any given state. Refinement of state transitions and estimation of state emissions can be achieved by

maximizing the likelihood function constructed from the training sequences. In the training phase, one trains an HMM for each $y \in \mathcal{Y}$. For an unknown object, one collects a sequence of observations (sensor data) and submit it to the HMM for each $y \in \mathcal{Y}$; the y yielding the maximum likelihood is then declared to be the identity of the unknown object. Obviously the agent must follow a common protocol to collect the data sequences in both the training and test phases, to ensure that their statistics are consistent. Since such a protocol is not part of the HMMs, a question arises as to how to specify the protocol.

From the perspective of sequential decision-making, multi-aspect classification can be formulated as a reinforcement learning problem, with a state space $\mathcal{S} = \mathcal{S}_\varphi \times \mathcal{Y}$, where $\times$ is a Cartesian product. Both $\mathcal{S}_\varphi$ and $\mathcal{Y}$ are only partially observable (through sensor data). The RL approach possesses several conspicuous advantages over the HMM approach. First, the sensor data are now collected in an active manner, under the control of agent actions. When two data sequences are collected by following the same policy of choosing actions, they are automatically ensured to be consistent in statistics, hence there is no need to specify a separate common protocol for collecting the sequential data. Second, unlike maximizing the data likelihood (under a given data collection protocol), the agent is now free to choose a more flexible learning objective by setting an appropriate reward structure. Third, unlike building a HMM for each $y \in \mathcal{Y}$, the different categories are now coalesced into a single RPR (details are presented below), making the RL a discriminative approach vis-a-via the generative HMM approach.

In our experiment, there are a total of five objects, four of them are targets of interest and one of them represents the clutter. The frequency-domain acoustic responses of these objects are shown in Figure 9, for a full coverage of angles from 0° to 360°; the data are real measurements as described in (Runkle et al. 1999). We aim to distinguish each target from clutter and this gives four tasks, where task i is defined by the problem of distinguishing target-i from clutter, $i = 1, 2, 3, 4$, and the targets and clutter are as shown in Figure 9. Each task is a multi-aspect classification problem[6]. From the data in Figure 9, targets 1 and 2 have similar angle-dependent scattering phenomena, and therefore Tasks 1 and 2 are expected to be related. Targets 3 and 4 also appear to have similar angle-dependent scattering characteristics, and therefore Tasks 3 and 4 are expected to also be related. In fact, although the target details are too involved to detail here, targets 1 and 2 are both of a cylindrical form (like those in (Runkle et al. 1999)), while targets 3 and 4 are more irregular in shape.

The RPR for Multi-aspect Classification In applying the RPR to multi-aspect classification, our approach is distinct from an HMM construction (Runkle et al. 1999) in two important respects. First, the RPR is a control model and it aims to optimize the value function, instead of the likelihood function. Since the RPR takes into account a reward structure, it can be more flexible in specifying the learning objective. Second, the RPR embraces all objects in the same representation, instead of having a separate model for each individual object. As a result, it is a discriminative model instead of a generative model (this may be viewed as a discriminative extension of the traditional generative HMM).

6. Upon publication, all data from this study will be put on a web site, for others to utilize in comparative studies.

The RPR does not manipulate the angular states — it works directly with observations. Since classification is treated as a control problem in the RPR, we need two extra components, actions and rewards, to complete the specification. We consider four actions, i.e., $\mathcal{A}$ ={*declare as target*, *declare as clutter*, *move clockwise and sense*, *move counterclockwise and sense*}. When the agent takes action *move clockwise and sense*, it moves 5° clockwise and collects an observation; when the agent takes action *move counterclockwise and sense*, it moves 5° counterclockwise and collects an observation. The reward structure is specified as follows. A correct declaration receives a reward of 5 units, a false declaration receives a reward of -5, and the actions *move clockwise and sense* and *move counterclockwise and sense* each receives a reward of zero units. The objective, therefore, is to correctly classify the target with the minimal number of sensing actions.

The episodes used in learning the RPR consist of a number of observation sequences, each observation is associated with the action *move clockwise and sense* or *move counterclockwise and sense* and the terminal action in each episode is the correct declaration. The correction declaration is available because the episodes in this problem are the training data in standard classification, hence the ground truth of class labels is known. Note that the training episodes always terminate with a correct declaration, thus the agent never actually receives the penalty -5 during the training phase. Alternatively, one may split each episode into two, respectively terminated with the correct and the false declaration. Recall the false declaration receives the minimum reward which, after an offset of 5 to make all rewards non-negative, is converted to zero. Since a zero reward received at the end of an episode nullifies the entire episode, such an alternative is equivalent to excluding the penalized episodes.

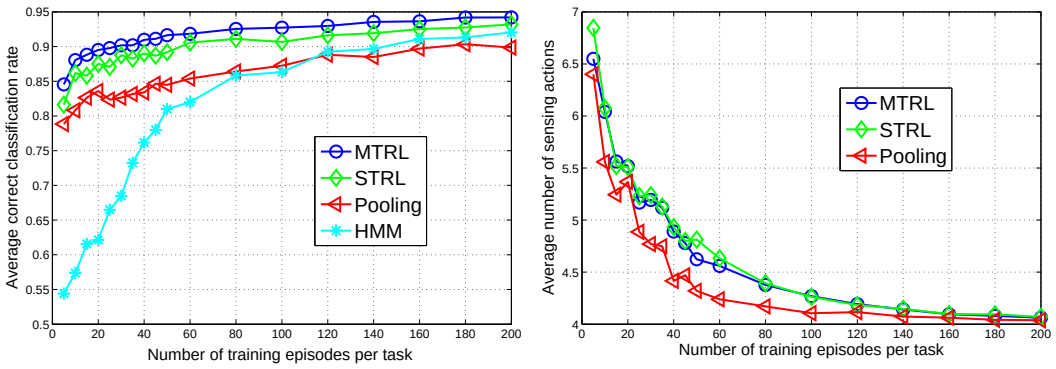
Classification Results The raw data are shown in Figure 9, for the five objects we are considering. Each datum is the response of an object measured at a particular angle and the data set for an object consists of measurements collected at $0°, 1°, \cdots, 359°$. Each raw datum is converted into a feature vector using matching pursuit (McClure and Carin 1997), and the feature vectors are further discretized by vector quantization (Gersho and Gray 1992) to produce a finite code-book. As mentioned earlier, we have a total of four tasks, each task being to distinguish each of the four respective targets from the clutter.

Four methods are compared: the MTRL, the STRL, the pooling, and the hidden Markov models (HMM), where the first three are as described in Section 6.2.1 and the last one is the standard hidden Markov model (Rabiner 1989). The four methods yield four corresponding agents, each following the policy resulting from one of the algorithms.

When the agents collect episodes during the training phase, they start from angles that are uniformly drawn from {1°, 2°, $\cdots$, 360°}. For each starting angle, two episodes are collected: the first is obtained by moving clockwise to collect an observation at every 5° and terminating upon the 10-th observation, and the other is the same as the first but the agent moves counterclockwise. During the testing phase, both the RPR agents and the HMM agent start from angles uniformly drawn from {1°, 2°, $\cdots$, 360°}; however, the RPR agents follow one of the three policies (resulting respectively from the MTRL, the STRL, and the pooling) to choose an action from $\mathcal{A}$, while the HMM agent collects n observations by moving consistently clockwise or counterclockwise (either direction is chosen with a

probability of 0.5) and then makes a declaration, where n is adaptively set to the *maximum* of the numbers of observations used by the three RPR agents starting from the same angle.

Figure 10 summarizes the performance as a function of the number of training episodes K, where the performance is evaluated by the correct classification rate as well as the average number of sensing actions (i.e., the average number of observations collected) before a declaration is made. Each point in the figures is an average from 20 independent trials.



(a) Classification rate (b) Average number of sensing steps

Figure 10: Performance comparison on multi-aspect classification of underwater targets (a) Average classification rate as a function of the number of training episodes per task (b) Average number of sensing actions (average number of observations collected) before a declaration is made, as a function of the number of training episodes per task.

It is seen from Figure 10(a) that the MTRL achieves the highest classification rate regardless of the number of training episodes K. The pooling performs worse than the STRL and the poor performance persists even when K is small. The latter is in contrast with the results on the maze navigation problems in Sections 6.2.1 and 6.2.2, where the pooling performs better than the STRL with a small K. The reason for this will be clear below from the sharing-mechanism analysis. It is noted that all three RPR algorithms perform much better than the HMM, demonstrating the superiority of discriminative models over generative models in classification problems.

As shown by Figure 10(b), pooling takes the least number of sensing actions, which may be attributed to the over-confidence arising from an abundant set of training data, noting that the pooling agent learns its policy by using the episodes accumulated over all tasks. In contrast, the STRL agent takes the most number of actions. Considering that the STRL agent bases policy learning on the episodes collected from a single task, which may contain inadequate information, it is reasonable that the STRL agent is less confident and would make more observations before coming to a conclusion. The sensing steps taken by the MTRL agent lies in between, since it relies on related tasks, but not all tasks, to provide the episodes for policy learning.

Analysis of the Sharing Mechanism The Hinton diagram of the between-task similarity matrix is shown in Figures 11(a), 11(b), 11(c), 11(d), for the cases when the number of training episodes K is equal to 10, 30, 110, 170, respectively.

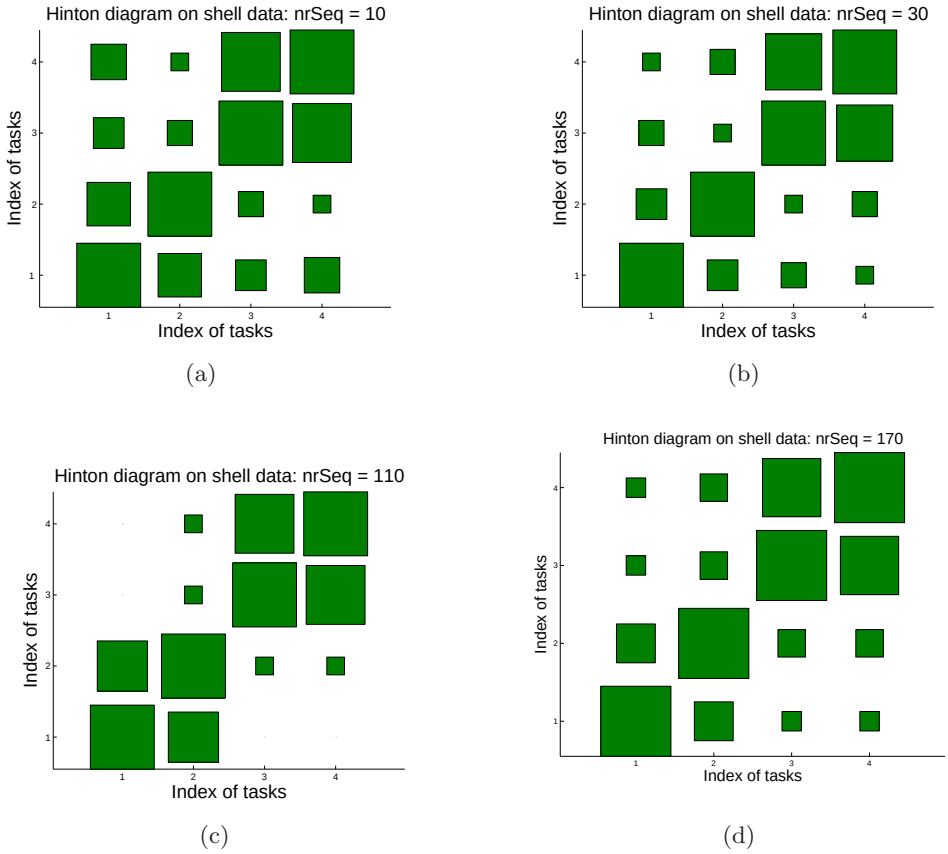


Figure 11: Sharing mechanism for multi-aspect classification of underwater targets. Each figure is the Hinton diagram of the between similarity matrix, with the number of training episodes per task: (a) 10 (b) 30 (c) 110 (d) 170.

It is seen that the sharing patterns are dominated by two clusters, the first consisting of Task 1 and Task 2 and the second consisting of Task 3 and Task 4. The second cluster remains unchanged regardless of K. The first cluster tends to break when $K = 30$, but is resumed later on. The two clusters are consistent with Figure 9 which shows that targets 1 and 2 are similar and so are targets 3 and 4. The fact the two clusters are persistent through the entire range of K implies that the tasks from different clusters are weakly related even when the episodes are scarce, as a result pooling the episodes across all tasks yields poor policies. This explains the poor performance of the pooling in Figure 10(a).

To understand the reason why the cluster of tasks 1 and 2 is less stable, one need delve into some details of the targets. Target 1 and Target 2 both have a cylindrical shape while Task 3 and Task 4 are more irregular in shape. Similar geometry puts Targets 1 and 2 in one cluster and Targets 3 and 4 in another cluster. Moreover, the measurements of Targets 3 and 4 are more noisy than the measurements of Targets 1 and 2, because they are collected under different conditions. The low signal to noise ratio (SNR) increases the similarity between Targets 3 and 4 since their distinctive features are buried in the noise. The more

noise-free measurements of Target 1 and 2 yields a more faithful representation of these targets, which tends to magnify their differences and make them appear less similar.

7. Conclusions

We have presented a multi-task reinforcement learning (MTRL) framework for *partially observable* stochastic environments. To our knowledge, this is the first framework proposed for MTRL in the partially observable domain.

A key element in our MTRL framework is the regionalized policy representation (RPR), which yields a history-dependent stochastic policy for environments characterized by a partially observable Markov decision process (POMDP). Learning of the RPR is based on episodic experiences collected from the environment, without requiring the environment's model. We have developed two algorithms for learning the RPR, one based on maximum-value estimation and the other based on the variational Bayesian paradigm. The latter offers the ability for selecting the number of decision states based on the Occam Razor principle and the possibility of transferring experience between related environments.

Built upon the basic RPR, the proposed MTRL framework consists of multiple RPRs, each for an environment, coupled by a common Dirichlet process (DP) that is used to produce the nonparametric prior over all RPRs. By virtue of the discreteness of the nonparametric prior, the environments are clustered into groups, with each group consisting of a subset of environments that are related in some manner. The number of groups as well as the associated environments are automatically identified, and the experiences are shared among the related environments to increase their respective exploration. A hybrid Gibbs-variational algorithm is presented for learning multiple RPRs simultaneously under the unified MTRL framework, based on selective use of the experiences collected across all environments.

Experimental results demonstrate that the proposed MTRL consistently yields superior performance regardless of the amount of experiences used in learning. The two competitors, one based on single-task reinforcement learning (STRL) and other based on simple pooling, are shown to be sensitive to the amount of experiences. The superior performance is attributed to the ability of the MTRL to automatically identify useful experiences from related environments to enhance the exploration. The MTRL adaptively adjusts sharing patterns to offset the changes in the experience and hence has addressed the problem of how to positively transfer the experience from one environment to the benefit of improving learning in another. In addition, we have also presented experimental results on benchmark problems demonstrating the RPR as a powerful stand-alone algorithm for single-task reinforcement learning.

The work presented in this paper mainly focuses on off-policy batch learning, assuming the learning is based on a fixed set of episodic experiences collected by following an external behavior policy. In the off-policy batch learning mode, the policy improvement is implemented without actually re-interacting with the environment; instead the improvement is implemented through virtual "reward re-computation" (discussed after (18)), which simulates the re-interaction with the environment. By taking reward re-computation out of the algorithm and implementing it via real re-interaction, we can learn the RPRs in an on-policy online manner. In this case, the need for an external behavior policy is eliminated and the

previous version RPR is employed as the behavior policy. In the next phase of this work, we will focus on on-policy online learning of RPRs and investigate how each environment can be better explored via multi-task reinforcement learning. In this on-policy MTRL setting, multi-task learning will have two aspects: co-exploitation (already addressed in the present paper) and co-exploration (not explicitly addressed here). It is of interest to investigate how much benefit can be gained by simultaneous co-exploitation and co-exploration.

Although the experiments considered in the paper mainly involve robot navigation in grid-worlds, there are many other interesting practical problems to which the proposed algorithms are immediately applicable. The multi-aspect classification serves as a preliminary example of such applications. Other examples include using RPRs as policies to control and coordinate a set of sub-models such that the collective performance is optimized and more advanced tasks could be accomplished than by any single sub-model.

For the work presented here, the DP prior is placed directly on Θ. Because of the discrete nature of G, this implies that when parameters Θ are shared between different environments, they are shared exactly. This may be too restrictive for some problems; for two environments that are similar, we may desire the associated parameters to be similar, but not exactly the same. This may be accommodated, for example, via the following modification to the DP prior

$$\begin{aligned} \Theta_m|\Psi_m &\sim H(\Theta_m|\Psi_m) \\ \Psi_m|G &\sim G \\ G|\alpha, G_0 &\sim DP(\alpha, G_0) \end{aligned}$$

This formulation results in an infinite mixture model for Θ, where each component is of the form H. When two environments share, their parameters share a component of this infinite mixture, but the specific draws will generally differ from each other — this can provide greater flexibility. The above modification brings some challenges to the inference. Recall that Θ is set of probability mass functions (pmf), it is natural to require H to be a product of Dirichlets. The difficulty now lies in choosing G that provides a conjugate prior for the parameters of H, which seems not easy. If G is properly specified, however, the inference should be a straightforward extension of the techniques developed in this paper. An alternative to the above modification that may avoid the inference difficulty is to follow the approach in (Liu et al. 2008) to impose soft sharing by replacing the Dirac delta with its soft version.

Acknowledgments

The authors would like to thank Prof. Ronald Parr of Duke University for pointing out the drawback of an early version of the RPR optimality criterion, and for the insightful counter-examples and the long discussions that eventually lead to the optimality criterion defined in Definition 4.

We would also like to thank the anonymous reviewers for their valuable comments and suggestions, which help to improve the paper significantly.

Appendix

Proof of Theorem 5

According to (Kaelbling et al. 1998), the expected sum of exponentially discounted reward (value function) over an infinite horizon can be written as

$$V = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{A-1}$$

where $0 < \gamma < 1$ is the discount factor. Let E denote the environment in question and $\mathscr{P}_E$ the corresponding probabilistic model (POMDP). Let Θ be the parameters specifying the RPR, the expectation in our situation here is $\mathbb{E}_{\text{episodes}|E,\Theta}$. Thus

$$\begin{aligned}
V &= \mathbb{E}_{\text{episodes}|E,\Theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \\
&= \sum_{a_0 r_0 o_1 a_1 r_1 \cdots} p(a_0 r_0 o_1 a_1 r_1 o_2 \cdots | E, \Theta)\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \\
&= \sum_{a_0 a_1 \cdots} p(a_0 a_1 \cdots | \Theta) \mathbb{E}_{r_0 o_1 r_1 o_2 r_2 \cdots | a_0 a_1 \cdots \sim \mathscr{P}_E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \\
&= \sum_{a_0 a_1 \cdots} p^{\Pi}(a_0 a_1 \cdots) \frac{p(a_0 a_1 \cdots | \Theta)}{p^{\Pi}(a_0 a_1 \cdots)} \mathbb{E}_{r_0 o_1 r_1 o_2 r_2 \cdots | a_0 a_1 \cdots \sim \mathscr{P}_E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \\
&\qquad \text{(Importance sampling (Robert and Casella 1999))} \\
&= \sum_{a_0 a_1 \cdots} p^{\Pi}(a_0 a_1 \cdots) \mathbb{E}_{r_0 o_1 r_1 o_2 r_2 \cdots | a_0 a_1 \cdots \sim \mathscr{P}_E}\left[\sum_{t=0}^{\infty} \frac{\gamma^t r_t}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau | h_\tau)} \prod_{\tau=0}^{t} p(a_\tau | h_\tau, \Theta)\right] \\
&= \mathbb{E}_{a_0 a_1 \cdots \sim p^{\Pi}} \mathbb{E}_{r_0 o_1 r_1 o_2 r_2 \cdots | a_0 a_1 \cdots \sim \mathscr{P}_E}\left[\sum_{t=0}^{\infty} \frac{\gamma^t r_t}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau | h_\tau)} \prod_{\tau=0}^{t} p(a_\tau | h_\tau, \Theta)\right] \\
&= \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K}\left[\sum_{t=0}^{\infty} \frac{\gamma^t r_t^k}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau^k | h_\tau^k)} \prod_{\tau=0}^{t} p(a_\tau^k | h_\tau^k, \Theta)\right] \\
&\qquad \left(a_0^k a_1^k \cdots \sim p^{\Pi},\ r_0^k o_1^k r_1^k o_2^k r_2^k \cdots | a_0^k a_1^k \cdots \sim \mathscr{P}_E\right) \\
&= \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K}\left[\sum_{t=0}^{T_k} \frac{\gamma^t r_t^k}{\prod_{\tau=0}^{t} p^{\Pi}(a_\tau^k | h_\tau^k)} \prod_{\tau=0}^{t} p(a_\tau^k | h_\tau^k, \Theta)\right] \\
&= \lim_{K\to\infty} \widehat{V}(\mathcal{D}^{(K)}; \Theta)
\end{aligned} \tag{A-2}$$

where the sum over $0 \le t < \infty$ is equal to the sum over $0 \le t \le T_k$ because $r_t^k = 0$ for $t > T_k$ according to Definition 2. Q.E.D.

Proof of Theorem 6

We begin our derivation by writing the empirical value function in its logarithm

$$\ln \widehat{V}(\mathcal{D}^{(K)}; \Theta) = \ln \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \tilde{r}_t^k \sum_{z_0^k, \cdots, z_t^k = 1}^{|\mathcal{Z}|} p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)$$

$$= \ln \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} \frac{q_t^k(z_{0:t}^k)}{K} \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)}{q_t^k(z_{0:t}^k)} \tag{A-3}$$

where

$$\begin{aligned} q_t^k(z_{0:t}^k) &\geq 0 \\ \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k) &= 1 \end{aligned} \tag{A-4}$$

Applying Jensen's inequality to (A-3), we obtain

$$\ln \widehat{V}(\mathcal{D}^{(K)}; \Theta) \geq \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} \frac{q_t^k(z_{0:t}^k)}{K} \ln \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta)}{q_t^k(z_{0:t}^k)} \tag{A-5}$$

The lower bound is maximized when

$$q_t^k(z_{0:t}^k) = q_t^k(z_{0:t}^k|\Theta) \stackrel{Def.}{=} \frac{\tilde{r}_t^k}{\widehat{V}(\mathcal{D}^{(K)}; \Theta)} p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \Theta) \tag{A-6}$$

which turns the inequality in into an equality. Define

$$\mathrm{LB}(\widehat{\Theta}|\Theta) = \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|} q_t^k(z_{0:t}^k|\Theta) \ln \frac{\tilde{r}_t^k p(a_{0:t}^k, z_{0:t}^k | o_{1:t}^k, \widehat{\Theta})}{q_t^k(z_{0:t}^k|\Theta)} \tag{A-7}$$

By (A-5), $\mathrm{LB}(\widehat{\Theta}|\Theta) \leq \mathrm{LB}(\widehat{\Theta}|\widehat{\Theta}) = \ln \widehat{V}(\mathcal{D}^{(K)}; \widehat{\Theta})$ holds for any Θ and $\widehat{\Theta}$. Therefore, when $\widehat{\Theta} = \arg\max_{\widehat{\Theta} \in \mathcal{F}} \mathrm{LB}(\widehat{\Theta}|\Theta)$, we have

$$\ln \widehat{V}(\mathcal{D}^{(K)}; \Theta) = \mathrm{LB}(\Theta|\Theta) \leq \mathrm{LB}(\widehat{\Theta}|\Theta) \leq \mathrm{LB}(\widehat{\Theta}|\widehat{\Theta}) = \ln \widehat{V}(\mathcal{D}^{(K)}; \widehat{\Theta})$$

Starting from $\Theta^{(0)}$ we compute

$$\begin{aligned} \Theta^{(1)} &= \arg\max_{\widehat{\Theta} \in \mathcal{F}} \mathrm{LB}(\widehat{\Theta}|\Theta^{(0)}) \\ \Theta^{(2)} &= \arg\max_{\widehat{\Theta} \in \mathcal{F}} \mathrm{LB}(\widehat{\Theta}|\Theta^{(1)}) \\ \vdots &\quad \vdots \end{aligned}$$

which satisfy $\widehat{V}(\mathcal{D}^{(K)}; \Theta^{(0)}) \leq \widehat{V}(\mathcal{D}^{(K)}; \Theta^{(1)}) \leq \widehat{V}(\mathcal{D}^{(K)}; \Theta^{(2)}) \leq \cdots$. Since the value function is upper bounded, this monotonically increasing sequence must converge, which happens at a maxima of $\widehat{V}(\mathcal{D}^{(K)}; \Theta)$. Q.E.D.

Proof of Lemma 7

Substituting (31) and (32), we have

$$\text{Right side of (33)} = \frac{p(z_\tau^k = i, z_{\tau+1}^k = j, a_{0:t}^k | o_{1:t}^k, \Theta)}{\prod_{\tau'=0}^{t} p(a_{\tau'}^k | h_{\tau'}^k)} \tag{A-8}$$

Since the denominator is equal to $p(a^k_{0:t}|o^k_{1:t},\Theta)$ by (10), we have

$$\text{Right side of (33)} = p(z^k_\tau = i, z^k_{\tau+1} = j|a^k_{0:t}, o^k_{1:t}, \Theta) = \xi^k_{t,\tau}(i,j) \tag{A-9}$$

Similarly,

$$\begin{aligned}\text{Right side of (34)} &= \frac{p(z^k_\tau = i, a^k_{0:t}|o^k_{1:t},\Theta)}{\prod_{\tau'=0}^t p(a^k_{\tau'}|h^k_{\tau'})} = \frac{p(z^k_\tau = i, a^k_{0:t}|o^k_{1:t},\Theta)}{p(a^k_{0:t}|o^k_{1:t},\Theta)} \\ &= p(z^k_\tau = i|a^k_{0:t}, o^k_{1:t},\Theta) \\ &= \phi^k_{t,\tau}(i)\end{aligned} \tag{A-10}$$

Q.E.D.

Appendix: Proof of Theorem 8

We rewrite the lower bound in (74) as

$$\begin{aligned}\mathrm{LB}\left(\left\{q^k_t\right\}, g(\Theta)\right) = \frac{1}{K}\sum_{k=1}^K\sum_{t=0}^{T_k}\sum_{z^k_0,\cdots,z^k_t=1}^{|\mathcal{Z}|}\int q^k_t(z^k_{0:t})g(\Theta)\ln\tilde{r}^k_t p(a^k_{0:t}, z^k_{0:t}|o^k_{1:t},\Theta)\,d\Theta \\ -\frac{1}{K}\sum_{k=1}^K\sum_{t=0}^{T_k}\sum_{z^k_0,\cdots,z^k_t=1}^{|\mathcal{Z}|} q^k_t(z^k_{0:t})\ln q^k_t(z^k_{0:t}) - \int g(\Theta)\ln\frac{g(\Theta)}{G_0(\Theta)}\,d\Theta\end{aligned} \tag{A-11}$$

We alternatively find the $\{q^k_t\}$ and $g(\Theta)$ that maximizes the lower bound, keeping one fixed while finding the other.

Keeping $g(\Theta)$ fixed, we solve $\max_{\{q^k_t\}}\mathrm{LB}\left(\left\{q^k_t\right\}, g(\Theta)\right)$ subject to the normalization constraint for $\{q^k_t\}$. We construct the Lagrangian

$$\ell_q = \mathrm{LB}\left(\left\{q^k_t\right\}, g(\Theta)\right) - \lambda\left(K - \sum_{k=1}^K\sum_{t=1}^{T_k}\sum_{z^k_0,\cdots,z^k_t=1}^{|\mathcal{Z}|} q^k_t(z^k_{0:t})\right) \tag{A-12}$$

where λ is the Lagrangian multiplier. Differentiating ℓ_q with respect to $q^k_t(z^k_{0:t})$ and setting the result to zero, we obtain

$$\frac{\partial\ell_q}{\partial\left(q^k_t(z^k_{0:t})\right)} = \frac{1}{K}\int g(\Theta)\ln\tilde{r}^k_t p(a^k_{0:t}, z^k_{0:t}|o^k_{1:t},\Theta)\,d\Theta - \frac{1}{K}\ln q^k_t(z^k_{0:t}) - \frac{1}{K} + \lambda = 0 \tag{A-13}$$

which is solved to give

$$q^k_t(z^k_{0:t}) = e^{K\lambda-1}\tilde{r}^k_t\exp\left\{\int g(\Theta)\ln p(a^k_{0:t}, z^k_{0:t}|o^k_{1:t},\Theta)\,d\Theta\right\} \tag{A-14}$$

Using the constraint $\sum_{k=1}^K\sum_{t=1}^{T_k}\sum_{z^k_0,\cdots,z^k_t=1}^{|\mathcal{Z}|} q^k_t(z^k_{0:t}) = K$, (76) is arrived with $e^{1-K\lambda} = C_z$.

Keeping $\{q^k_t\}$ fixed, we solve $\max_{g(\Theta)}\mathrm{LB}\left(\left\{q^k_t\right\}, g(\Theta)\right)$ subject to the normalization constraint that $\int g(\Theta)d\,\Theta = 1$. Construct the Lagrangian

$$\ell_g = \mathrm{LB}\left(\left\{q^k_t\right\}, g(\Theta)\right) - \lambda\left(1 - \int g(\Theta)d\,\Theta\right) \tag{A-15}$$

where λ is the Lagrangian multiplier. Differentiating ℓ_g with respect to $g(\Theta)$ and setting the result to zero, we obtain

$$\frac{\partial \ell_g}{\partial\left(g(\Theta)\right)}=\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}q_t^k(z_{0:t}^k)\ln\tilde{r}_t^k p(a_{0:t}^k,z_{0:t}^k|o_{1:t}^k,\Theta)-1-\ln\frac{g(\Theta)}{G_0(\Theta)}+\lambda=0 \quad \text{(A-16)}$$

which is solved to give

$$g(\Theta)=\frac{G_0(\Theta)}{e^{1-\lambda}}\exp\left\{\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T_k}\sum_{z_0^k,\cdots,z_t^k=1}^{|\mathcal{Z}|}q_t^k(z_{0:t}^k)\ln\tilde{r}_t^k p(a_{0:t}^k,z_{0:t}^k|o_{1:t}^k,\Theta)\right\} \quad \text{(A-17)}$$

By using the constraint $\int g(\Theta)d\,\Theta=1$, we arrive at (77) with $e^{1-\lambda}=C_\Theta$. Q.E.D.

References

D. Aberdeen and J. Baxter. Scalable internal-state policy-gradient methods for POMDPs. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 3–10. Morgan Kaufmann Publishers Inc., 2002.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, November 1974.

B. Bakker. *The State of Mind: Reinforcement Learning with Recurrent Neural Networks*. PhD thesis, Unit of Cognitive Psychology, Leiden University, 2004.

B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

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

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

D. Blackwell. Discounted dynamic programming. *Ann. Math. Stat.*, 36:226–235, 1965.

D. Blackwell and J. MacQueen. Ferguson distributions via Polya urn schemes. *Annals of Statistics*, 1:353–355, 1973.

H.-T. Cheng. *Algorithms for partially observable Markov decision processes*. PhD thesis, University of British Columbia, Vancouver, BC, 1988.

L. Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth International Conference on Artificial Intelligence*, pages 183–188. San Jose, California: AAAI Press, 1992.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.

F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. In *Proceedings of the 25th international conference on Machine learning*, pages 256–263. ACM, 2008.

M. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

T. Ferguson. A Bayesian analysis of some non-parametric problems. *The Annals of Statistics*, 1:209–230, 1973.

A. Gelfand and A. Smith. Sample based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Press/Springer, 1992.

C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

E. A. Hansen. An improved policy iteration algorithm for partially observable MDPs. In *Advances in neural information processing systems*, volume 10, 1997.

G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In J. L. McClelland, D. E. Rumelhart, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 282–317. MIT Press, Cambridge, MA, 1986.

T. Jaakkola, S. P. Singh, and M. I. Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, Cambridge, MA., 1995.

T. S. Jaakkola. Tutorial on variational approximation methods. In M. Opper and D. Saad, editors, *Advanced mean field methods: Theory and practice*, pages 129–160. MIT Press, 2001.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in graphical models*, pages 105–161, Cambridge, MA, 1999. MIT Press.

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

N.D. Lawrence and J.C. Platt. Learning to learn with the informative vector machine. *In Proceedings of the 21st International Conference on Machine Learning*, 2004.

H. Li. *Planning and Learning in Partially Observable Stochastic Environments*. PhD thesis, Duke University, 2006. publically available at http://people.ee.duke.edu/~hl1/.

H. Li, X. Liao, and L. Carin. Incremental least squares policy iteration for POMDPs. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, 2006a.

H. Li, X. Liao, and L. Carin. Region-based value iteration for partially observable Markov decision processes. In *Proceedings of the 23nd International Machine Learning Conference*, 2006b.

X. Liao, H. Li, R. Parr, and L. Carin. Regionalized policy representation for reinforcement learning in POMDPs. In *The Snowbird Learning Workshop*, 2007.

M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially obsevable environments: scaling up. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 362–370, 1995.

Qiuhua Liu, Xuejun Liao, and Lawrence Carin. Semi-supervised multitask learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 937–944. MIT Press, Cambridge, MA, 2008.

W. S. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. *Operations Research*, 39(1):162–175, 1991.

S. N. MacEachern. Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics: Simulation and Computation*, 23:727–741, 1994.

R. A. McCallum. *Reinforcement Learning with Selective Attention and Hidden State*. PhD thesis, Department of Computer Science, University of Rochester, 1995.

M. McClure and L. Carin. Matched pursuits with a wave-based dictionary. *IEEE Trans. Signal Proc.*, 45:2912–2927, Dec. 1997.

N. Meuleau, L. Peshkin, K. Kim, and L. Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 427–43, San Francisco, CA, 1999. Morgan Kaufmann.

R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, 1998.

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

P. Poupart and C. Boutilier. Bounded finite state controllers. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

S. Ross, B. Chaib-draa, and J. Pineau. Bayes-adaptive POMDPs. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, 2008.

P. R. Runkle, P. K. Bharadwaj, L. Couchman, and L. Carin. Hidden Markov models for multiaspect target classification. *IEEE Transactions on Signal Processing*, 47:2035–2040, July 1999.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operational Research*, 21:1071–1088, 1973.

T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*, 2005.

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

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

M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195220, 2005.

R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 1998.

S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems (NIPS)*, 1996.

T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML)*, pages 961–968, 2005.

M. Welling, Y. W. Teh, and B. Kappen. Hybrid variational/Gibbs collapsed inference in topic models. In *Proceedings of the Conference on Uncertainty in Artifical Intelligence (UAI)*, pages 587–594, 2008.

M. West. Hyperparameter estimation in Dirichlet process mixture models. Technical Report 92-A03, ISDS Discussion Paper, Duke University, 1992.

M. West, P. Muller, and M.D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In A.F.M. Smith and P. Freeman, editors, *Aspects of Uncertainty: A Tribute to D. V. Lindley*, pages 363–386. New York: Wiley, 1994.

D. Wierstra and M. Wiering. Utile distinction hidden Markov models. In *Proceedings of the International Conference on Machine Learning*, 2004.

A. Wilson, A. Fern, S. Ray, and P. Tadepalli. Multi-task reinforcement learning: A hierarchical Bayesian approach. *In Proceedings of the 24st International Conference on Machine Learning*, 2007.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research (JMLR)*, 8:35–63, 2007.

K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical Bayes. *In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical Bayesian framework for information filtering. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.