On the Recognition of Abstract Markov Policies

Hung H. Bui, Svetha Venkatesh and Geoff West
Department of Computer Science
Curtin University of Technology
PO Box U1987, Perth, WA 6001, Australia
{buihh, svetha, geoff}@cs.curtin.edu.au

Abstract

Abstraction plays an essential role in the way the agents plan their behaviours, especially to reduce the computational complexity of planning in large domains. However, the effects of abstraction in the inverse process – plan recognition – are unclear. In this paper, we present a method for recognising the agent's behaviour in noisy and uncertain domains, and across multiple levels of abstraction. We use the concept of abstract Markov policies in abstract probabilistic planning as the model of the agent's behaviours and employ probabilistic inference in Dynamic Bayesian Networks (DBN) to infer the correct policy from a sequence of observations. When the states are fully observable, we show that for a broad and often-used class of abstract policies, the complexity of policy recognition scales well with the number of abstraction levels in the policy hierarchy. For the partially observable case, we derive an efficient hybrid inference scheme on the corresponding DBN to overcome the exponential complexity.

Introduction

While planning their behaviours in large domains, the agents often have to employ abstraction techniques to reduce the complexity associated with large state spaces. As a consequence, their behaviours often follow a hierarchichal plan structure. In abstract probabilistic planning (Sutton, Precup, & Singh 1999; Parr & Russell 1997; Forestier & Varaiya 1978; Hauskrecht *et al.* 1998; Dean & Lin 1995), such a hierarchichal plan structure can be modelled using the concept of *abstract Markov policies* [1]. In this paper, we address the problem of recognising such a policy – the inverse of the abstract probabilistic planning problem, i.e. to infer the underlying abstract policy from the external observation of the induced behaviour.

The problem can be classified under the umbrella of key-hole plan recognition (Cohen, Perrault, & Allen 1981; Kautz & Allen 1986) where the agent who carries out the plan (the actor) is not aware that it is being observed. The recent trend in approaching this problem is first constructing a probabilistic model for the plan execution, and then employing abductive reasoning techniques such as probabilistic (Bayesian) inference to infer the underlying plan from the observation sequence (Huber, Durfee, & Wellman 1994; Goldmand, Geib, & Miller 1999). To represent the evolution of the plan over time, some authors (Forbes *et al.* 1995; Pynadath & Wellman 1995; Albrecht, Zukerman, & Nicholson 1998) have employed the Dynamic Bayesian Network (DBN) (Dean & Kanazawa 1989; Nicholson & Brady 1992) as the framework for inferencing. All of this work in plan recognition uses a hierarchichal structure of plans and actions. However, to the best of our knowledge, the question of how the structure of a plan hierarchy would affect the complexity of the recognition process has not been investigated.

In our approach, we adopt the Abstract Markov Policies (AMP) as the model for plan execution. The AMP is an extension of a policy in Markov Decision Processes (MDP) that enables an abstract policy to invoke other more refined policies and so on down the policy hierarchy. The concept originates from the abstract probabilistic planning literature as a way to scale up MDP-based planning to domains whose state spaces are large. An AMP can be described simply in terms of a state space and a Markov policy that selects among a set of other AMP's. Thus, using the AMP as the model for plan execution helps us focus on the structure of the policy hierarchy alone.

Throughout the paper, we make the assumption that only the states (which bear the effects of actions) can be observed, but not the actions themselves. Thus, the policy recognition problem can be divided into two cases: when state observations are certain (*fully observable*), and when the observations are partial and uncertain (*partially observable*) [2].

Computationally, we view policy recognition as probabilistic inference on the Dynamic Bayesian Network representing the execution of the AMP (together with the noisy observation of states in the partially observable case). Intuitively, the DBN models how an AMP causes the adoption of other policies and actions at different levels of abstraction, which in turn generate a sequence of states and observation. In policy recognition, we need to reverse the direction of causality and compute the conditional probability of the top-

[1] Also known as *options*, *policies of Abstract Markov Decision Processes*, *supervisor's policies*.

[2] This only means that the external observer has partial information. Our model however assumes that the actor always has full information about the current state.

level AMP given an observation sequence.

It is known that the complexity of this kind of inferencing in the DBN depends on the size of the representation of the so-called *belief state*, the conditional joint distribution of the variables in the DBN at time t given the observation sequence up to t (Boyen & Koller 1998). Thus we can ask the following question: how does the structure of an AMP affect the size of its belief state representation? Using conditional independence relations on the network, we show that a reduction in size of the belief state is possible if at some level of abstraction, the set of applicable domains of all the policies forms a partition of the full state space. As a consequence, if the AMP is constructed through the region-based decomposition of the state space as in (Forestier & Varaiya 1978; Dean & Lin 1995; Hauskrecht *et al.* 1998), its belief state can be represented compactly as a chain in the fully observable case, facilitating very efficient inferencing. Based on this result, for the partially observable case, we derive a hybrid inference scheme combining both exact inference as in the fully observable case, and sampling-based approximative inference to handle the noisy observation. We provide experiment results showing the hybrid inference scheme achieving much better accuracy/time ratio than the original sampling-based inference (Kanazawa, Koller, & Russell 1995).

The main body of the paper is organised as follows. The next two sections introduce the abstract Markov policy model and its DBN representation. The algorithms for policy recognition are discussed next, first for the fully observable and then for the partially observable case. We then present the experiment results supporting the new hybrid inference scheme.

Abstract Markov Policies

In this section, we formally introduce the AMP concept as originating from the literature of abstract probabilistic planning with MDP (Sutton, Precup, & Singh 1999; Parr & Russell 1997; Forestier & Varaiya 1978; Hauskrecht *et al.* 1998; Dean & Lin 1995). Note that the work in planning is concerned with computing the optimal (abstract) policy given some reward function. In policy recognition, although it is possible to derive some information about the reward function by observing the agent's behaviour, we choose not to do this, thus omit from our model the reward function and also the optimality notion. This leaves the model open to tracking arbitrary agent's behaviours, regardless of whether they are optimal or not.

The general model

In an MDP, the world is modelled as a set of possible states S, termed the state space. At each state s, an agent has a set of actions A available, where each action a, if employed, will cause the world to evolve to the next state s' via a transition probability $\sigma_a(s, s')$. An agent's plan of actions is modelled as a policy that prescribes how the agent would choose its action at each state. For a policy π, this is modelled by a selection function $\sigma_\pi : S \times A \rightarrow [0, 1]$ where at each state s, $\sigma_\pi(s, a)$ is the probability that the agent will choose the action a.

In the original MDP, behaviours are modelled at only two levels: the primitive action level, and the plan level (policy). We would like to consider policies that selects other more refined policies and so on, down a number of abstraction levels. The idea is to form intermediate-level abstract policies as policies defined over a local region of the state space, having a certain terminating condition, and can be invoked and executed just like primitive actions (Forestier & Varaiya 1978; Sutton, Precup, & Singh 1999). Formally, let Π be a set of abstract policies. We can then define an abstract policy π^* over Π as a tuple $\langle S_{\pi^*}, D_{\pi^*}, \sigma_{\pi^*} \rangle$ where $S_{\pi^*} \subset S$ is the set of applicable states, $D_{\pi^*} \subset S$ is the set of destination states, and $\sigma_{\pi^*} : S_{\pi^*} \times \Pi \rightarrow [0, 1]$ is the selection function where $\sigma_{\pi^*}(s, \pi)$ is the probability that π^* selects the policy π at the state s. When an abstract policy π^* is invoked, it will keep on selecting the policies in Π for execution, and terminate whenever a destination state $d \in D_{\pi^*}$ is reached [3]. Note the recursiveness in this definition that allows an abstract policy to select among a set of other abstract policies. At the base level, primitive actions can be viewed as abstract policies themselves (Sutton, Precup, & Singh 1999).

Using the abstract policies as building blocks, we can construct a hierarchy of abstract policies as follows: A policy hierarchy is a sequence $\mathcal{H} = (\Pi_0, \Pi_1, \ldots, \Pi_K)$ where Π_0 is a set of primitive actions, and for $k = 1, \ldots, K$, Π_k is a set of abstract policies over the policies in Π_{k-1}. When a top-level policy π_K is executed, it invokes a sequence of level-(K-1) policies, each of which invokes a sequence of level-(K-2) policies and so on. A level-1 policy will invoke a sequence of primitive actions which leads to a sequence of states. Thus, the execution of π_K generates an overall state sequence $(s^{(0)}, s^{(1)}, \ldots, s^{(t)}, \ldots)$ that terminates in one of the destination states in D_{π_K}. When $K = 1$ this sequence is simply a Markov chain, however, for $K \geq 2$, it will generally be non-Markovian (Sutton, Precup, & Singh 1999).

State-space region-based decomposition

An intuitive and often-used method for constructing the policy hierarchy is via region-based decomposition of the state space (Dean & Lin 1995; Hauskrecht *et al.* 1998). Here the state space S is successively partitioned into a sequence of partitions $\mathcal{P}_K, \mathcal{P}_{K-1}, ...\mathcal{P}_1$ where $\mathcal{P}_K = \{S\}$ is the coarsest partition, and $\mathcal{P}_1$ is the finest, corresponding to the K levels of abstraction. For each region R_i of $\mathcal{P}_i$, the periphery of R_i, $Per(R_i)$ is defined as the set of states not in R_i, but connected to some state in R_i. Let Per_i be the set of all peripheral states at level i: $Per_i = \cup_{R_i \in \mathcal{P}_i} Per(R_i)$. Fig. 1(b) shows an example where the state space representing a building is partitioned into 4 regions corresponding to the 4 rooms. The peripheral states for a region is shown in Fig 1(a), and Fig 1(b) shows all such peripheral states.

To construct the policy hierarchy, we first define for each region $R_1 \in \mathcal{P}_1$ a set of abstract policies applicable on R_1,

[3] Sutton, Precup and Singh's model allows non-deterministic stopping conditions and thus is more general than what we consider here. Nevertheless, our theorem 1 can also be shown to hold in the general case as well.

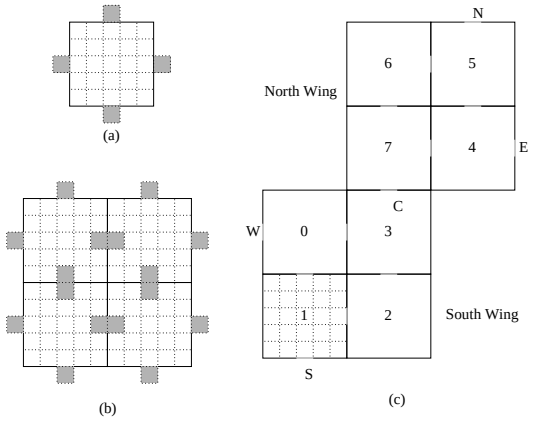


Figure 1: The environment and its partition

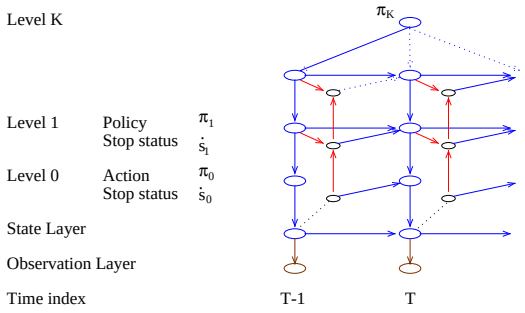


Figure 2: The DBN representation

and having $Per(R_1)$ as the destination states. For example, for each room in Fig 1, we can define a set of policies that model the agent's different behaviours while it is inside the room, e.g. getting out through a particular door. These policies can be initiated whenever the agent steps inside the room, and terminate when the agent steps out of the room (not necessarily through the target door since the policy might fail to achieve its intended target). Let the set of all policies defined be Π_1. At the higher level $\mathcal{P}_2$, for each region R_2, we can define a set of policies that model the the agent's behaviours inside that region with the constraint that these policies must use the policies previously defined at level-1 to achieve their goals. An example is a policy to navigate between the room-doors to get from one building gate to another. Let the set of all policies defined at this level be Π_2. Continuing doing this at the higher levels, we obtain the policy hierarchy $\mathcal{H} = (\Pi_0, \Pi_1, \Pi_2, \ldots, \Pi_K)$.

Dynamic Bayesian Network Representation

The process of executing the top-level abstract policy π_K can be represented by a DBN (Fig. 2). At time t, the current time slice consists of the variables representing the current state $s^{(t)}$, and the current policies at different levels of abstraction $\pi_k^{(t)}, k = 0, \ldots, K$. We assume that the top-level policy remains unchanged, so $\pi_K^{(t)} = \pi_K$ for all t.

Let $\dot{s}_k^{(t)}, k = 0, \ldots, K-1$ represent the stopping status of the policy $\pi_k^{(t)}$. That is $\dot{s}_k^{(t)} = s^{(t)}$ if the policy terminates at time t, and $\dot{s}_k^{(t)} = false$ otherwise. At the base level, since the primitive action always terminates after one time step, $\dot{s}_0^{(t)} = s^{(t)}$ for all t. This is indicated by the special dotted lines in Fig. 2. Generally, we can look at the stopping status of the current policy at the lower level $\dot{s}_{k-1}^{(t)}$ and see whether this falls inside the destination states of $\pi_k^{(t)}$ to determine the value of $\dot{s}_k^{(t)}$. Thus, $\dot{s}_k^{(t)}$ is dependent (deterministically) on $\pi_k^{(t)}$ and $\dot{s}_{k-1}^{(t)}$.

Now the evolution from one time-slice of the DBN to the next is as follows: the current policy $\pi_k^{(t)}$ is dependent on the current policy at the higher level $\pi_{k+1}^{(t)}$, the policy at the previous time $\pi_k^{(t-1)}$, and its stopping status $\dot{s}_k^{(t-1)}$. If $\dot{s}_k^{(t-1)} = false$, the previous policy $\pi_k^{(t-1)}$ persists to time t, otherwise, a new policy at level-k is generated by $\pi_{k+1}^{(t)}$ using the distribution $\sigma_{\pi_{k+1}^{(t)}}(\dot{s}_k^{(t-1)}, .)$.

We term the dynamical process in executing a top-level abstract policy π_K the *Abstract Markov Model* (AMM). When the states are only partially observable, an observation layer can be attached to the state layer (Fig. 2). The resulting process is termed the *Abstract Hidden Markov Model* (AHMM) since the states are hidden as in the Hidden Markov Model (Rabiner 1989).

The belief state of the AMM is a joint distribution of $K+2$ variables in the current time-slice: $s^{(t)}, \pi_0^{(t)}, \ldots, \pi_K^{(t)}$. Thus generally, the size of the belief state representation will be exponential of K. However, due to the way policies are invoked in the AMM, we can make an intuitive remark that, the higher level policies can only influence what happens at the lower level through the current level. Therefore, knowing enough information about the current level would make the higher level policies probabilistically independent of the lower level policies. If this kind of probabilistic independence relations can be exploited, the belief state could be represented more compactly.

Thus, our motivation here is to find the least amount of information we need to know about the current level, in order to make the current policies at the higher and lower levels independent. The following theorem states that if at level k, we know the current policy, together with its starting time and starting state, then the higher levels are independent of the lower levels. The condition obtained is the strictest, in the sense that, if one of these three variables is unknown, there are examples of AMMs in which the higher level policies can influence the lower level ones.

Theorem 1. *Let $\tau_k^{(t)}$ and $b_k^{(t)}$ be two random variables representing the starting time and the starting state, respectively, of the current level-k policy $\pi_k^{(t)}$: $\tau_k^{(t)} = \max\{t' < t \mid \dot{s}_k^{(t)} \neq false\}$ and $b_k^{(t)} = s^{(\tau_k^{(t)})}$. Let $\pi_{>k}^{(t)} = \{\pi_{k+1}^{(t)}, \ldots, \pi_K^{(t)}\}$ denote the set of current policies from level $k+1$ up to K, and $\pi_{<k}^{(t)} = \{s^{(t)}, \pi_0^{(t)}, \ldots, \pi_{k-1}^{(t)}\}$ de-*

note the set of current policies from level $k-1$ down to 0 together with the current state. We have:

$$\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t)}_k, b^{(t)}_k, \tau^{(t)}_k \qquad (1)$$

Proof. (Sketch) Induction by t. Can easily verify for $t = 1$. Suppose that (1) holds for $t-1$, need to prove it for t. There are two cases. If $\tau^{(t)}_k < t-1$, meaning $\pi^{(t)}_k$ started before $t-1$, therefore $\pi^{(t)}_k = \pi^{(t-1)}_k$, $\tau^{(t)}_k = \tau^{(t-1)}_k$, $b^{(t)}_k = b^{(t-1)}_k$, and also for the higher levels, $\pi^{(t)}_{>k} = \pi^{(t-1)}_{>k}$. From the structure of the DBN, it can be seen that $\pi^{(t-1)}_k, \pi^{(t)}_k, \pi^{(t-1)}_{<k}$ d-separate $\pi^{(t)}_{<k}$ from all other variables up to time t. From this, we can obtain $\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t-1)}_{<k}, \pi^{(t)}_k, b^{(t)}_k, \tau^{(t)}_k$. Combining this with the inductive assumption $\pi^{(t)}_{>k} \perp \pi^{(t-1)}_{<k} \,|\, \pi^{(t)}_k, b^{(t)}_k, \tau^{(t)}_k$, and using the contraction property of the relation $\perp$ (Pearl 1988, p84), we obtain $\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t)}_k, b^{(t)}_k, \tau^{(t)}_k$. In the second case, if $\tau^{(t)}_k \geq t-1$, but since $\tau^{(t)}_k \leq t-1$, we have $\tau^{(t)}_k = t-1$ and $b^{(t)}_k = s^{(t-1)}$. This means that the policy $\pi^{(t)}_k$, and all the policies in $\pi^{(t)}_{<k}$ have just been formed at the previous time at the state $s^{(t-1)}$. Therefore, $\pi^{(t)}_{<k}$ is d-separated from the upper levels by $\pi^{(t)}_k$ and $s^{(t-1)}$. Thus, $\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t)}_k, b^{(t)}_k = s^{(t-1)}, \tau^{(t)}_k = t-1$. □

Policy Recognition

The policy recognition problem can be formulated as to compute the conditional probabilities of the current policies, given the current observation sequence. In this section, we will assume full observability, i.e. the observation sequence at time t is the state sequence $\tilde{s}^{(t-1)} = (s^{(0)}, \ldots, s^{(t-1)})$. In more concrete terms, we are interested in the probabilities $\Pr(\pi^{(t)}_k \,|\, \tilde{s}^{(t-1)})$ for all level k. This gives us information about the agent's behaviour at all levels of abstraction, from the current action ($k = 0$), to the top-level policy ($k = K$). In addition, continuous monitoring would require to compute these probabilities in all time steps t (whenever an observation is made).

Our solution to this problem is based on updating the belief state of the AMM, i.e. the joint distribution $\Pr(\pi^{(t)}_K, \ldots, \pi^{(t)}_0, s^{(t)} \,|\, \tilde{s}^{(t-1)})$. Thus, the size of the representation of this distribution is crucial.

The belief chain

From Theorem 1, if the starting time of the current level-k policy $\tau^{(t)}_k$ can be determined from the state sequence $\tilde{s}^{(t-1)}$, then so can the starting state $b^{(t)}_k$, and the belief state can be split into two independent parts conditioned on $\pi^{(t)}_k$: $\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t)}_k, \tilde{s}^{(t-1)}$. This condition is satisfied when the applicable domains of all the policies at level k do not partially overlap (i.e. any two domains are either identical or mutually exclusive), so that the starting point of the current policy $\pi^{(t)}_k$ can be identified as the last time the state history sequence $\tilde{s}^{(t-1)}$ crossed one of the peripheral states in Per_k: $\tau^{(t)}_k = \max_{t'}\{t' < t, s^{(t')} \in Per_k\}$.

As a consequence, if the policy hierarchy is constructed via region-based decomposition of the state-space, $\pi^{(t)}_{>k} \perp \pi^{(t)}_{<k} \,|\, \pi^{(t)}_k, \tilde{s}^{(t-1)}$ holds for all k, and thus the belief state can be represented by a Bayesian network with a simple chain structure. We term this network the *belief chain* (Fig. 3(a)), and denote it by $\mathcal{C}^{(t)}(\tilde{s}^{(t-1)}) \equiv \Pr(\pi^{(t)}_K, \ldots, \pi^{(t)}_0, s^{(t)} \,|\, \tilde{s}^{(t-1)})$. If a chain is drawn so that all links point away from the level-k node, we say the chain has root at level k. The root of the chain can be moved from k to another level k' simply by reversing the links between k and k'. For consistency purpose, level -1 will refer to the node $s^{(t)}$, and thus $\pi^{(t)}_{-1} \equiv s^{(t)}$. In the remaining of the paper, we assume that the policy hierarchy has been constructed via the region-based decomposition method, and thus deal exclusively with this chain structure.

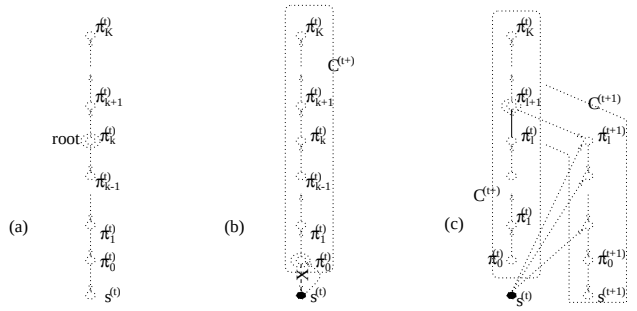


Figure 3: The belief chain and its updating process

Updating the belief chain

Since the belief state can be represented by a chain (and thus has size linear to K), we can expect that exact inference method based on updating the belief state will work efficiently. Here, we briefly describe such a method. More details can be found in a longer version of this paper (Bui, Venkatesh, & West 2000).

Assuming that we have a complete specification of the chain $\mathcal{C}^{(t)}$, we need to compute the parameters for the new chain $\mathcal{C}^{(t+1)}$. This is done in two steps, as in the standard "roll-over" of the belief state of a DBN (Boyen & Koller 1998): (1) absorbing the new evidence $s^{(t)}$, and (2) projecting the belief state into the next time step.

In the first step, we need to compute the chain $\mathcal{C}^{(t+)} \equiv \Pr(\pi^{(t)}_K, \ldots, \pi^{(t)}_0 \,|\, \tilde{s}^{(t)})$. This can be achieved by positioning the root of the chain $\mathcal{C}^{(t)}$ at level 0, and then reverse the link from $\pi^{(t)}_0$ to $s^{(t)}$ to absorb the new evidence $s^{(t)}$ (Fig. 3(b)).

In the second step, we continue to compute $\mathcal{C}^{(t+1)}$ from $\mathcal{C}^{(t+)}$. We note that the new state $s^{(t)}$ might cause some policies at the lower levels to terminate. Let l be the highest level of such a policy; l is deterministically determined since $l = \max\{0\} \cup \{k \geq 1 \,|\, s^{(t)} \in Per_k\}$. Since all the poli-

cies at levels higher than l do not terminate, $\pi_{>l}^{(t+1)} = \pi_{>l}^{(t)}$, and we can retain this upper sub-chain from $\mathcal{C}^{(t+)}$ to $\mathcal{C}^{(t+1)}$. In the lower part, for $k \leq l$, a new policy $\pi_k^{(t+1)}$ is created by the policy $\pi_{k+1}^{(t+1)}$ at the state $s^{(t)}$, and thus a new sub-chain can be formed among the variables $\pi_{<l}^{(t+1)}$ with parameters $\Pr(\pi_k^{(t+1)} \mid \pi_{k+1}^{(t+1)}, s^{(t)}) = \sigma_{\pi_{k+1}^{(t+1)}}(s^{(t)}, \pi_k^{(t+1)})$. The new chain $\mathcal{C}^{(t+1)}$ is then the combination of these two sub-chains, which will be a chain with root at level $l+1$ (see Fig. 3(c)).

The complexity of this updating process is $O(l)$. However, most of the time, none of the policies from level one up would terminate, and thus $l = 0$. More precisely, the probability that the current policy at level l terminates is exponentially small w.r.t. l. Thus, on average, the updating complexity at each time-step is $O(\sum_l l/exp(l))$ which is constant-bounded. Once the current belief chain is obtained, the required probability $\Pr(\pi_k^{(t)} \mid \tilde{s}^{(t-1)})$ is simply the marginal $\mu_k^{(t)}$ at the level-k node in the chain $\mathcal{C}^{(t)}$.

The Partially Observable Case

We now consider the scenarios when the states cannot be observed with certainty. The observation at time t is denoted by $o^{(t)}$, and is assumed to depend stochastically on $s^{(t)}$ only, via the observation model $\omega(s, o) = \Pr(o \mid s)$ (the probability that o is observed given the actual state is s). Thus, in comparison with the previous section, we now deal with a process represented by the AHMM, and the main inference problem becomes computing the probabilities $\Pr(\pi_k^{(t)} \mid \tilde{o}^{(t-1)})$.

The hidden states make the inference tasks in the AHMM much more difficult. Since the exact state sequence is not available, neither the starting times or the starting states of the current policies are known with certainty. Thus, theorem 1 cannot be used. We therefore cannot hope to represent the belief state by a chain as we did previously. In this case, an exact method for updating the belief state will have to operate on a structure with size exponential in K. To cope with this complexity, one generally has to resort to some approximative inference method instead.

In (Bui, Venkatesh, & West 1999), the stochastic sampling method for DBN (Kanazawa, Koller, & Russell 1995) has been applied to a network structure similar to the AHMM. This inference method makes use of a procedure that reverses the link from the state to the observation before sampling the next state (*evidence reversal (ER)*), so that the sampled state sequence would stay close to the true sequence. However, when applied to the multi-layer AHMM, the ER procedure only affects the sampling of the next state, while having no effect on the sampling of the next policies at the higher levels. Furthermore, since the method is intended to be an inference scheme for generic DBNs, it does not utilise the special structure of the AHMM.

Here, we present a method that uses stochastic sampling only at the state level, and performs exact inference through belief-chain updating at the higher levels. Thus, the method can be classified as an *hybrid-inference* scheme (Dawid, Kjærulff, & Lauritzen 1995) which attempts to combine approximative inference with exact inference on the tractable parts of the model network. We describe the details of this inference method below.

Hybrid inference

First, we have the following expansion of the wanted probability $\Pr(\pi_k^{(t)} \mid \tilde{o}^{(t-1)})$:

$$\propto \sum_{\tilde{s}^{(t-1)}} \Pr(\pi_k^{(t)} \mid \tilde{s}^{(t-1)}) \Pr(\tilde{o}^{(t-1)} \mid \tilde{s}^{(t-1)}) \Pr(\tilde{s}^{(t-1)})$$

$$= \mathop{\mathrm{E}}_{\tilde{s}^{(t-1)}} \left[\mu_k^{(t)}(\tilde{s}^{(t-1)}) w^{(t)}(\tilde{s}^{(t-1)}) \right] \quad (2)$$

where $\mu_k^{(t)}$ is the level-k marginal of the chain $\mathcal{C}^{(t)}$, $w^{(t)}(\tilde{s}^{(t-1)}) = \Pr(\tilde{o}^{(t-1)} \mid \tilde{s}^{(t-1)})$ is termed the *weight* of the sequence $\tilde{s}^{(t-1)}$, and the expectation operator is taken over the distribution $\Pr(\tilde{s}^{(t-1)})$.

Using Monte-Carlo approximation of (2), a set of sequences $\mathcal{S}^{t-1}$ called the *sample population* are sampled from the distribution $\Pr(\tilde{s}^{(t-1)})$, and (2) can be approximated by:

$$\sum_{\tilde{s}^{(t-1)} \in \mathcal{S}^{t-1}} \mu_k^{(t)}(\tilde{s}^{(t-1)}) w^{(t)}(\tilde{s}^{(t-1)}) \quad (3)$$

In (3), the marginal $\mu_k^{(t)}$ can be taken from the chain $\mathcal{C}^{(t)}$, which can be updated efficiently from $\mathcal{C}^{(t-1)}$ as we have shown in the previous section. The weight $w^{(t)}$ can also be updated using $w^{(t+1)} = w^{(t)} \Pr(o^{(t)} \mid s^{(t)})$. Thus, all the terms in (3) afford very efficient update from the same terms at the previous time step.

The only remaining problem is how to generate the sample population $\mathcal{S}^t$ from the old population $\mathcal{S}^{t-1}$. This is done by lengthening each sample $\tilde{s}^{(t-1)}$ in $\mathcal{S}^{t-1}$ with a new state $s^{(t)}$, obtained by sampling from the distribution [4] $\Pr(s^{(t)} \mid \tilde{s}^{(t-1)})$, which is simply the marginal $\mu_{-1}^{(t)}$ of the chain $\mathcal{C}^{(t)}(\tilde{s}^{(t-1)})$.

Overall, the main updating step for the hybrid-inference scheme is given in Fig. 4. Note that, in the data structure for a sample, we only have to keep the last state $s^{(t-1)}$, since all the future-relevant information from the sequence $\tilde{s}^{(t-1)}$ has been summarised in the chain and and the weight of the sample. Both the time and space complexity of this updating step are linear to the sample population size, and to the number of levels of abstraction K.

Experiments

By performing exact inference through updating the chain of each sample, our algorithm avoids sampling at the higher layers, and thus can be expected to achieve better accuracy than the original sampling method in (Kanazawa, Koller, &

[4] The sampling process is described here without the evidence reversal step for simplicity. When ER is used, the next state can be sampled from the distribution $\Pr(s^{(t)} \mid \tilde{s}^{(t-1)}, o^{(t)}) \propto \Pr(o^{(t)} \mid s^{(t)}) \mu_{-1}^{(t)}$.

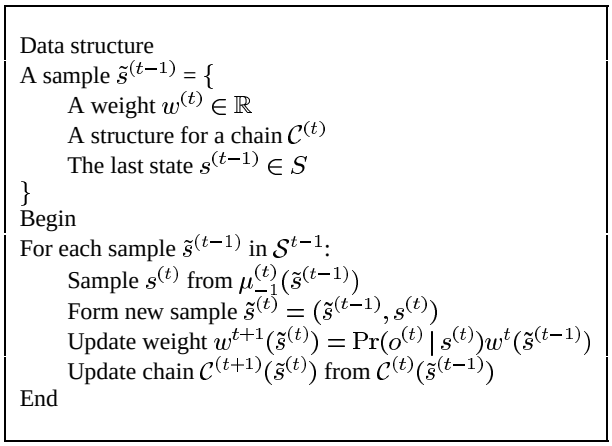


Figure 4: Updating algorithm for hybrid-inference

Russell 1995). Here, we provide the experimental result to back up this claim.

The experiment involves a synthetic tracking task in which it is required to monitor and predict the movement of an agent through a building shown in Fig. 1(c). Each room is represented by a 5x5 grid, and at each state, the agent can move in 4 possible directions. These actions have 0.5 failure probability, in which case the agent either stays in the same cell, or move unintendedly to one of the other three neighbours. The policy hierarchy is constructed based on region-based decomposition at three levels of abstraction. The partition of the environment consists of the 8 rooms at level 1, the two wings (north and south) at level 2, and the entire building at level 3. In each room, we specify 4 level-1 policies to model the agent's behaviours of exiting the room via the 4 different doors. Similarly, we specify 3 level-2 policies in each wing corresponding with the 3 wing exits, and a total of 4 top-level policies corresponding to the 4 building exits. All the parameters of these actions and policies are chosen manually, and then are used to simulate the movement of the agent in the building.

We implement both the sampling inference scheme of (Kanazawa, Koller, & Russell 1995) (with ER and survival-of-the-fittest) and our hybrid inference (with ER). In a typical run, the algorithm can return the probability of the next building exit, the next wing exit, and the next room-door that the agent is currently heading to (Bui, Venkatesh, & West 1999). Here, we run the two algorithms using different sample population sizes to obtain their performance profiles. For a given population size, the standard deviation over 50 runs in the estimated probabilities of the top-level policies is used as the measure of expected error in the probability estimates. We also record the average time taken in each update iteration.

Fig. 5(a) plots the average error of the two algorithms for different sample sizes. As expected, for the same number of samples, the hybrid algorithm delivers much better accuracy. This however comes with the overhead in updating the belief chain for each sample, which makes the new algorithm run about twice slower for a given sample size (Fig. 5(b)). Fig. 5(c) plots the actual CPU time taken versus the expected error for the two algorithms. It shows that for the same CPU time spent, the hybrid inference still significantly reduces the error in the probability estimates. Alternatively, the hybrid inference can achieve the same error margin with only about half the CPU time.

Discussion and Conclusion

In summary, we have presented a framework for the recognition of abstract Markov policies from external observation, based on probabilistic inference on the Dynamic Bayesian Network representation of the abstract policy. The paper presents two contributions to this problem. First, the analysis of the fully-observable case shows that policy recognition can be carried out more efficiently if the domains of the intermediate abstract policies form a partition of the state-space (i.e. non-overlapping), due to fact that the belief state of the DBN can be represented more compactly in this case. As a result, for abstract policies constructed by region-based decomposition of the state-space, policy recognition can be performed with constant averaged complexity (i.e. not dependent on the number of levels of abstraction). In the second contribution, we derive an efficient hybrid inference scheme for the recognition of this class of abstract policies under partial-observability. Experimental results illustrate our hybrid inference scheme performs better than the existing sampling-based scheme of (Kanazawa, Koller, & Russell 1995).

Several future research directions are possible. To solve the more general problem, we need to consider what happens when the policy domains overlap, or their border is "fuzzy" (i.e. when stopping conditions are non-deterministic). Our analysis here points the difficulty of this case to the uncertainty in the starting times of the current policies. In addition, we are currently applying the framework presented here to the problem of tracking and predicting human movement in large spatial environment, and as part of this, addressing the problem of learning the parameters of the model from training data.

Acknowledgement

The research described here was supported by an ARC grant from the Australian Research Council. The authors would like to thank a number of anonymous reviewers for their helpful comments.

References

Albrecht, D. W.; Zukerman, I.; and Nicholson, A. E. 1998. Bayesian models for keyhole plan recognition in an adventure game. *User Modelling and User-adapted Interaction* 8(1–2):5–47.

Boyen, X., and Koller, D. 1998. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*.

Bui, H. H.; Venkatesh, S.; and West, G. 1999. Layered dynamic Bayesian networks for spatio-temporal modelling. *Intelligent Data Analysis* 3(5):339–361.

Bui, H. H.; Venkatesh, S.; and West, G. 2000. On the recognition of abstract Markov policies. Technical Report 3/2000, Department of Computer Science, Curtin University of Technology, Perth, WA, Australia.

Cohen, P. R.; Perrault, C. R.; and Allen, J. F. 1981. Beyond question answering. In Lehnert, W., and Ringle, M., eds., *Strategies for Natural Language Processing*, 245–274. Hillsdale, NJ: Lawrence Erlbaum Associates.

Dawid, A. P.; Kjærulff, U.; and Lauritzen, S. 1995. Hybrid propagation in junction trees. In Zadeh, L. A., ed., *Advances in Intelligent Computing*, Lecture Notes in Computer Science, 87–97.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.

Dean, T., and Lin, S.-H. 1995. Decomposition techniques for planning in stochastic domains. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*.

Forbes, J.; Huang, T.; Kanazawa, K.; and Russell, S. 1995. The BATmobile: towards a Bayesian automated taxi. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1878–1885.

Forestier, J.-P., and Varaiya, P. 1978. Multilayer control of large Markov chains. *IEEE Transactions on Automatic Control* 23(2):298–305.

Goldmand, R.; Geib, C.; and Miller, C. 1999. A new model of plan recognition. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*.

Hauskrecht, M.; Meuleau, N.; Kaelbling, L. P.; Dean, T.; and Boutilier, C. 1998. Hierarchical solution of Markov decision processes using macro-actions. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*.

Huber, M. J.; Durfee, E. H.; and Wellman, M. P. 1994. The automated mapping of plans for plan recognition. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence*.

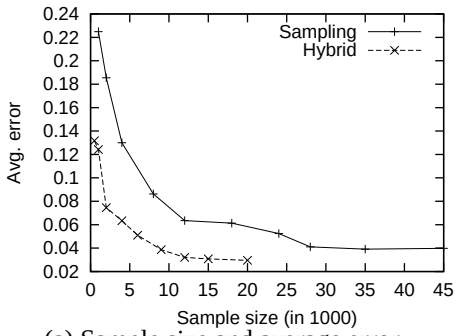
Kanazawa, K.; Koller, D.; and Russell, S. 1995. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, 346–351.

Kautz, H., and Allen, J. F. 1986. Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 32–38.

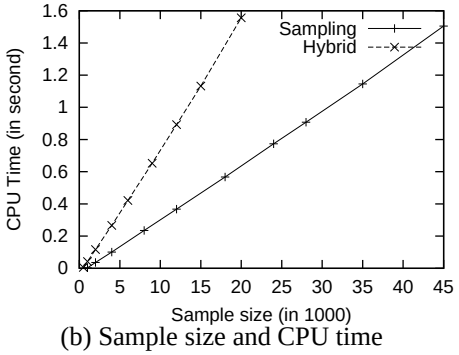
Nicholson, A. E., and Brady, J. M. 1992. The data association problem when monitoring robot vehicles using dynamic belief networks. In *Proceedings of the Tenth European Conference on Artificial Intelligence*, 689–693.

Parr, R., and Russell, S. 1997. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Sytems (NIPS-97)*.

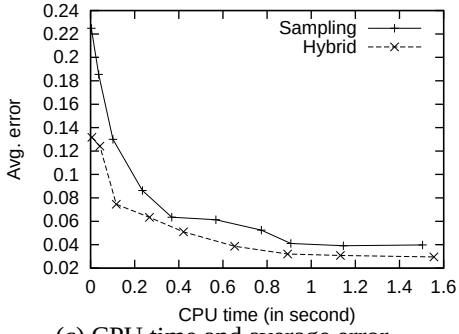
Pearl, J. 1988. *Probabilitic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.



(a) Sample size and average error



(b) Sample size and CPU time



(c) CPU time and average error

Figure 5: Sampling alone vs. hybrid inference

Pynadath, D. V., and Wellman, M. P. 1995. Accounting for context in plan recognition, with application to traffic monitoring. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*.

Rabiner, L. R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–286.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDP and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112:181–211.