Abstract Hidden Markov Models for Online Probabilistic Plan Recognition

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

Abstract

Abstract Markov Policy (AMP) is a model for representing the execution of an abstract plan in noisy and uncertain domains. Methods for recognising an abstract policy from a sequence of noisy observations thus can be used for online plan recognition under uncertainty. In this paper, we extend previous work on policy recognition and consider a general type of abstract policies, including those with non-deterministic terminating conditions and factored representations of the state space. We analyse the structure of the stochastic model representing the execution of the general AMP and provide an efficient hybrid Rao-Blackwellised sampling method for policy recognition that scales well with the number of levels in the plan hierarchy. This illustrates that while the stochastic models for plan execution can be complex, they exhibit special structures which, if exploited, can lead to efficient plan recognition algorithms.

Introduction

The plan recognition problem

Plan recognition is the problem of inferring an actor's plan by watching the actor's actions and their effects. Often, the actor's behaviour follows a hierarchical plan structure. Thus in plan recognition, the observer needs to infer about the actor's plans and sub-plans at different levels of abstraction in the plan hierarchy. The problem is complicated by the two sources of uncertainty inherent in the actor's planning process: (1) the stochastic nature of plan refinement (a plan can be non-deterministically refined into different sub-plans), and (2) the stochastic outcomes of actions (the same action can non-deterministically result in different outcomes). Furthermore, the observer has to deal with a third source of uncertainty arising from the noise and inaccuracy in its own observation about the actor's plan (partial observability). In addition, we would like our observer to be able to perform the plan recognition task "on-line", as the observations about the actor's plan streaming in. We refer to this type of problems as *on-line plan recognition under uncertainty*.

The seminal work in plan recognition (Kautz & Allen 1986) although considers a plan hierarchy does not deal with the uncertainty aspects of the problem. As a result, they can only postulate a set of possible plans for the actor, but unable to answer which plan is more probable. Since then, the important role of uncertainty reasoning in plan recognition has been recognised (Charniak & Goldman 1993; Bauer 1994; van Beek 1996), with Bayesian probability being argued as the appropriate model (Charniak & Goldman 1993; van Beek 1996). The plan recognition Bayesian network proposed in (Charniak & Goldman 1993) however is a static network and therefore is not suitable for online processing a stream of evidence about the plan. This dynamic, "on-line" aspect of plan recognition has only been recently considered (Pynadath & Wellman 1995; 2000; Goldmand, Geib, & Miller 1999; Huber, Durfee, & Wellman 1994; Albrecht, Zukerman, & Nicholson 1998). All of this recent work views online plan recognition as probabilistic inference in a stochastic process that models the execution of the actor's plan. While this view offers a general and coherent framework for modelling different sources of uncertainty, the stochastic process that we need to deal with can become quite complex, especially if we consider a large plan hierarchy. Thus, the main issue here is the computational complexity for dealing with this type of stochastic processes, and whether the complexity is scalable to more complex plan hierarchies.

These previous approaches have been limited in addressing this important issue. To achieve efficiency, some have opted for simplified models of plan execution (Huber, Durfee, & Wellman 1994; Albrecht, Zukerman, & Nicholson 1998). Those who consider a detailed model of plan execution (Pynadath & Wellman 2000; Goldmand, Geib, & Miller 1999) have not provided adequate computational techniques to deal with the resulting complex stochastic process. In particular, the work by (Goldmand, Geib, & Miller 1999) mainly serves as a representational framework. Pynadath and Wellman (2000) provide an algorithm for their Probabilistic State Dependent Grammar (PSDG) in the fully observable case, however have not adequately addressed the partially observable case.

Policy recognition

In a recent approach, Bui *et al.* (2000) has investigated online probabilistic plan recognition in the framework of policy recognition in a hierarchy of *abstract Markov policies*

(AMP). The AMP model originates from the abstract probabilistic planning literature (Sutton, Precup, & Singh 1999; Parr & Russell 1997; Forestier & Varaiya 1978) and is an extension of a policy in Markov Decision Processes (MDP) which enables an abstract policy to invoke other more refined policies and so on down the policy hierarchy. The AMP is thus similar to a contingent plan that prescribes which sub-plan should be invoked at each applicable state of the world to achieve its intended goal, except that it can represent both the uncertainty in the plan refinement and in the outcomes of actions.

The execution of an AMP leads to a special stochastic process called the *Abstract Markov Model* (AMM). The noisy observation about the environment state (e.g. the effects of action) can then be modelled by making the state "hidden", similar to the hidden state in the Hidden Markov Models (Rabiner 1989). The result is an interesting and novel stochastic process termed the *Abstract Hidden Markov Model*. Intuitively, the AHMM 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 observations. In the plan recognition task, an observer is given an AHMM corresponding to the actor's plan hierarchy, and is asked to infer about the current policy being executed by the actor at all levels of the hierarchy, taking into account the sequence of observations currently available.

Interestingly, the policy hierarchy considered in (Bui, Venkatesh, & West 2000) is equivalent to a restricted type of PSDG (Pynadath & Wellman 2000) where only production rules of the type $X \rightarrow Y\,X$ and $X \rightarrow \emptyset$ are allowed (the former rule represents a higher level policy X invokes a lower level policy Y, while the latter rule represents the termination of a policy X). Intuitively, this restriction means that control should always be returned to the higher level plan after an invoked sub-plan terminates. Although ruling out uninterrupted sequences of sub-plans, this is a reasonable assumption in a stochastic plan execution model since each component sub-plan might fail and thus need the intervention of the higher level plan to recover. The policy hierarchy however shares the same limitation with the PSDG which assumes a single top-level plan and does not consider interleaved concurrent plans.

Bui *et al*. (2000) have shown that the complexity of policy recognition scales reasonably well w.r.t. the number of abstraction levels in the policy hierarchy. However, the analysis is limited to a special type of policy hierarchy termed the state-space region-based decomposition (SRD) policy hierarchy (Dean & Lin 1995). In an SRD policy hierarchy, at each level of abstraction, the applicable regions of the policies form a partition of the state space. The boundaries between different applicable regions are clear cut, and each policy terminates deterministically if and only if the current state falls outside of the applicable region of the policy.

The main contribution of this paper is to extend the result in (Bui, Venkatesh, & West 2000) to the most general type of policy hierarchies, where the applicable regions of the policies can overlap, and non-deterministic policy termination is allowed. To achieve this aim, we first re-examine the Dynamic Bayesian Network representation of the AHMM to identify a set of context specific independence (CSI) properties (Boutilier *et al*. 1996) that are inherent in the execution model of an abstract policy. We then show that the hybrid inference method used in (Bui, Venkatesh, & West 2000), a variant of the Rao-Blackwellised sampling scheme (Casella & Robert 1996; Doucet *et al*. 2000), can take advantage of these CSI properties in the AHMM, leading to an efficient hybrid algorithm for policy recognition in the general case. In addition, we show that the AHMM representation and the hybrid policy recognition algorithm can also utilise a factored representation of the state space (Boutilier, Dearden, & Goldszmidt 2001). This further extends the applicability of the AHMM to the case where the state space is composed of many but relatively independent variables.

The main body of the paper is organised as follows. Section 2 formally introduces the AHMM as a general stochastic model for plan execution and the associated noisy observations. The new algorithm for policy recognition in the general AHMM is discussed in section 3. We finally conclude and discuss directions for further research in Section 4.

Abstract Hidden Markov Model

Policy hierarchy

Consider an MDP with the state space S and the set of actions A. At each state s, the agent has a set of applicable actions $A(s) \subset A$. Each action $a \in A(s)$ would cause the world to evolve to the next state s' with 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.

To model policies that select other more refined policies and so on down a number of abstraction levels, we need 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).

Definition 1 (Abstract Markov Policy). Let Π be a set of abstract policies, an abstract policy π^* over Π is defined as a tuple $\langle S_{\pi^*}, D_{\pi^*}, \beta_{\pi^*}, \sigma_{\pi^*} \rangle$ where:

- $S_{\pi^*} \subset \cup_{\pi \in \Pi} S_\pi$ is the set of applicable states.
- $D_{\pi^*} \subset \cup_{\pi \in \Pi} D_\pi$ is the set of destination states. $\beta_{\pi^*} : D_{\pi^*} \rightarrow (0, 1]$ is the set of stopping probabilities such that $\beta_{\pi^*}(d) = 1, \forall\, d \in D_{\pi^*} \setminus S_{\pi^*}$.
- $\sigma_{\pi^*} : S_{\pi^*} \times \Pi \rightarrow [0, 1]$ is the selection function where $\sigma_{\pi^*}(s, \pi)$ is the probability that π^* selects the policy π at the state s.

The set S_{π^*} models the local region over which the abstract policy is applicable. The stopping condition of the policy is modelled by a set of possible destination states D_{π^*} and stopping probabilities β_{π^*}, where $\beta_{\pi^*}(d)$ is the probability that the policy will terminate when the current state is d. It is possible to allow the policy to stop at

some state outside of its applicable region, however, for all $d \in D_{\pi^*} \setminus S_{\pi^*}$ we enforce the condition that $\beta_{\pi^*}(d) = 1$, i.e. d is a terminal destination state. If we consider only policies with deterministic stopping condition as in (Bui, Venkatesh, & West 2000), every destination is a terminal destination: $\beta(d) = 1 \, \forall d \in D$. In this special case, we can ignore the redundant parameter β and need only specify the set of destinations D. The selection function σ_{π^*} models how π^* selects the policies at the lower level. If π is selected by π^*, π will be executed until termination before control is returned to π^*.

Using abstract policies as the building blocks, a hierarchy of abstract policies can be constructed as follows. A policy hierarchy is a sequence $\mathcal{H} = (\Pi_0, \Pi_1, \ldots, \Pi_K)$ where K is the number of levels in the hierarchy, 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. The dynamical process of executing a top-level abstract policy π^K is termed the *Abstract Markov Model* (AMM). When the states are only partially observable, the noisy observation can be modelled by the usual observation model $\Pr(o_t \mid s_t) = \omega(s_t, o_t)$. The resulting process is termed the *Abstract Hidden Markov Model* (AHMM) since the states are hidden as in the Hidden Markov Model (Rabiner 1989).

DBN representation

The DBN representation of the AHMM can be constructed similar to the network structure in (Bui, Venkatesh, & West 2000). At time t, the current time slice includes the current state s_t and the current policies at different levels of abstraction π_t^k, $k = 0, \ldots, K$. Since policies are allowed to terminate non-deterministically here, we use the boolean variable e_t^k to represent the ending status of the policy π_t^k: e_t^k is true if π_t^k terminates at the current time and false otherwise. The full network for the AHMM is given in Fig. 1. All the links pointing to an ending status node e_t^k represent the policy stopping model $\beta_{\pi_t^k}$, while all the links pointing to a policy node π_t^k represent the policy selection function $\sigma_{\pi_t^{k+1}}$.

Context specific independence properties

The AHMM exhibits other interesting conditional independence properties that are not represented by the DBN above. These properties can be described as a type of context specific independence (CSI), i.e. conditional independence statements which hold only under a specific instantiation of the conditioning variables (Boutilier *et al.* 1996). There are two sources of CSI in the AHMM: from the termination of the current policy, and from the selection of a new policy. We describe these CSI properties below.

Policy termination Consider the links pointing to a policy ending status node e_t^k in the DBN shown in Fig. 1. Among the three parents of e_t^k, the node e_t^{k-1} plays a special role as the "context" variable. If $e_t^{k-1} = T$, meaning the lower level policy terminates at the current time, $\Pr(e_t^k = T \mid \pi_t^k, s_t) = \beta_{\pi_t^k}(s_t)$ which gives the conditional probability of e_t^k given the other two parent variables. However, if $e_t^{k-1} = F$, π_t^k should not terminate and so $e_t^k = F$. Therefore, given that $e_t^{k-1} = F$, e_t^k is deterministically determined and is independent of the other two parent variables π_t^k and s_t. From the network manipulation rule for context-specific independence (Boutilier *et al.* 1996), we can then safely remove the links from the other two parents to e_t^k in the context that e_t^{k-1} is false (Fig. 2(a)).

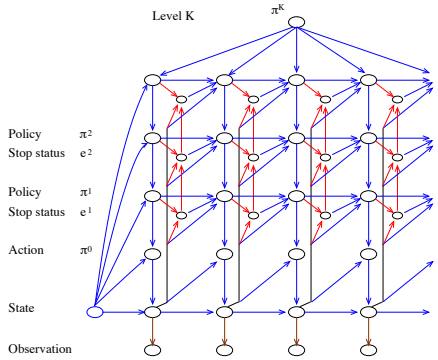

Figure 1: The DBN representation of the Abstract Hidden Markov Model

At the bottom level, since the primitive action always terminates immediately, $e_t^0 = T$ for all t. Since we are modelling the execution of a single top-level policy π^K, we can assume that the top-level policy does not terminate and remains unchanged: $e_t^K = F$ and $\pi_t^K = \pi^K$ for all t. Also, note that $e_t^l = T \Rightarrow e_t^k = T$ for all $k \leq l$, and $e_t^l = F \Rightarrow e_t^k = F$ for all $k \geq l$. Thus, at each time t, there exists $0 \leq l_t < K$ such that $e_t^k = T$ for all $k \leq l_t$, and $e_t^k = F$ for all $k > l_t$. The variable l_t is termed the highest level of termination at time t. Knowing the value of l_t is equivalent to knowing the terminating status of all the current policies.

Policy selection Similarly, consider the links pointing to a policy node π_t^k. Among the four parents, the node e_{t-1}^k can be considered as a context variable. If the previous policy has not terminated ($e_{t-1}^k = F$), the current policy is the same as the previous one: $\pi_t^k = \pi_{t-1}^k$ and the variable π_t^k is thus independent of π_t^{k+1} and s_{t-1}. Therefore, in the context $e_{t-1}^k = F$, the two links from π_t^{k+1} and s_{t-1} to the current policy can be removed, and the two nodes π_t^k and π_{t-1}^k can be merged together (Fig. 2(b)). If the previous policy has terminated ($e_{t-1}^k = T$), the current policy is selected by the higher level policy with probability $\Pr(\pi_t^k \mid \pi_t^{k+1}, s_{t-1}) = \sigma_{\pi_t^{k+1}}(s_{t-1}, \pi_t^k)$. In this context, π_t^k is independent of π_{t-1}^k and the corresponding link in the Bayesian network can be removed (Fig. 2(c)).

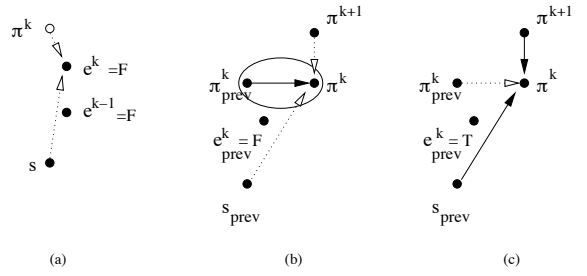

Figure 2: Context specific independence in the AHMM

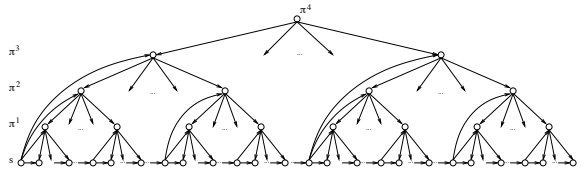

Figure 3: Simplified network if the duration of each policy is known (action nodes are omitted for clarity)

Consequences of CSI in the AHMM The rules for network manipulation corresponding to the CSI properties described above provide a systematic way to modify the network structure of the AHMM when conditioned on the policy ending status variables. This allows us to derive useful conditional independence properties about the AHMM which are not obvious under the full DBN representation.

For example, if all the policy ending status nodes are known, i.e. we know the exact duration of each policy in the network, we can modify the full DBN to obtain a more intuitive tree-shaped network shown in Fig. 3. In this network, all the policy nodes corresponding to the same policy in its entire duration are grouped into one. Furthermore, if the state sequence is also known (full observability), the remaining network becomes singly-connected and is therefore amenable to very efficient exact inference algorithms.

The CSI also provides an alternative and more intuitive proof of Theorem 1 in (Bui, Venkatesh, & West 2000), which states that the current higher level policies are independent of the current lower level ones given the current level-k policy π_t^k, its starting state and its starting time. The idea is to proceed by first modifying the full DBN under the context that we know the starting time of π_t^k. At the starting time of π_t^k, the previous level-k policy has just terminated, and since then the current level-k policy has not terminated. Conditioning on these policy ending status will give us a new network structure in which the higher level policies and the lower level ones are d-separated by π_t^k and its starting state. The theorem then follows immediately.

Policy recognition in the AHMM

In policy recognition, we assume that a policy hierarchy is given and is modelled by an AHMM, however the top level policy and the details of its execution are unknown. The problem is then to determine the top level policy and other current policies at the lower levels given the current sequence of observations. Thus we need to compute[1] $\Pr(\pi_t^{all} \mid \tilde{o}_{t-1})$ where $\pi_t^{all} = (\pi_t^K, \ldots, \pi_t^0)$, $\tilde{o}_{t-1} = (o_0, \ldots, o_{t-1})$, and especially the marginals $\Pr(\pi_t^k \mid \tilde{o}_{t-1})$ for all levels k.

To compute these probabilities "online", as each new observation becomes available, we need to update the belief state (filtering distribution) of the AHMM at each time point t. Here, the belief state is a joint distribution of $K+3$ discrete variables: $\Pr(\pi_t^K, \ldots, \pi_t^0, s_t, l_t \mid \tilde{o}_t)$. Thus updating the belief state is generally intractable, especially when K is large.

Rao-Blackwellisation

To cope with this complexity, one generally has to resort to some form of approximation to trade off accuracy for computational resources. The key to an efficient approximation, as pointed out by (Bui, Venkatesh, & West 2000), is a hybrid method that combines both exact and sampling-based inference to utilise the special structure of the AHMM. This hybrid inference turns out to be a variant of a general method for combining exact and sampling-based inference known as Rao-Blackwellisation (Casella & Robert 1996; Doucet *et al*. 2000).

When applied to general DBN inference, Rao-Blackwellisation splits the network into two sets of variables: the set of variables that need to be sampled (termed the Rao-Blackwellising (RB) variables), and the set of remaining variables whose belief state conditioned on the RB variables need to be maintained (termed the Rao-Blackwellised (RB) belief state). The RB variables thus play a similar role to the cut-set variables in cut-set inference (Pearl 1988). The difference is that instead of summing over all the possible values of the cut-set variables which can be intractable, only a number of representative sampled values are used.

Similar to cut-set variables, RB variables should be chosen to simplify the network structure of the remaining variables. Thus, context variables in CSI properties, and variables which help to cut loop in the network structure are the potential candidates. In the case of the AHMM, the policy ending status variables $\tilde{l}_t$ are the context variables conditioning on which simplifies the DBN to a more manageable structure in Fig. 3. The state sequence $\tilde{s}_t$ then helps break all the loops on this structure. We thus choose $\tilde{l}_t$ and $\tilde{s}_t$ to be the RB variables in the AHMM. Note that if only region-based decomposition policy hierarchies are considered, the RB variables contain only the state sequence $\tilde{s}_t$ as in (Bui, Venkatesh, & West 2000), due to the fact that the policy ending status $\tilde{l}_t$ can be deterministically derived from the state sequence $\tilde{s}_t$.

[1] We can also use similar methods to compute the smoothed probability $\Pr(\pi_t^k \mid \tilde{o}_{t+r})$, with a fixed lag $r \geq 0$.

Using $\tilde{l}_t$ and $\tilde{s}_t$ as the RB variables, the required probability $\Pr(\pi_t^k \mid \tilde{o}_{t-1})$ can be rewritten as $\bar{h} = \mathrm{E}_{\tilde{l}_{t-1},\tilde{s}_{t-1}|\tilde{o}_{t-1}} h(\tilde{l}_{t-1}, \tilde{s}_{t-1})$, where the h function is $\Pr(\pi_t^k \mid \tilde{l}_{t-1}, \tilde{s}_{t-1})$. The expectation $\bar{h}$ can then be approximated by performing sequential importance sampling [2] (SIS) (Doucet, Godsill, & Andrieu 2000) for the RB variables $(\tilde{l}_t, \tilde{s}_t)$.

There are two main steps in the overall algorithm. In the exact step, we need to maintain the RB belief state, which is a Bayesian network representation of the distribution $\mathcal{B}_t = \Pr(\pi_t^{all}, l_t, s_t, o_t \mid \tilde{l}_{t-1}, \tilde{s}_{t-1})$. From this, the function h can be computed as the marginal at the node π_t^k. In the sampling step, we need to sample the new RB variables l_t and s_t from the distribution $\Pr(l_t, s_t \mid \tilde{l}_{t-1}, \tilde{s}_{t-1}, o_t)$. This again can be done using the network representation of $\mathcal{B}_t$. We describe these two steps briefly below. Space restriction prevents a full detailed presentation.

The exact step We note that the RB belief state $\mathcal{B}_t$ can be factorised into $\Pr(o_t \mid s_t) \Pr(l_t \mid \pi_t^{all}, s_t)\mathcal{C}_t$, where $\mathcal{C}_t = \Pr(\pi_t^{all}, s_t \mid \tilde{l}_{t-1}, \tilde{s}_{t-1})$ is the belief chain, a chain structure representing the joint distribution of the current policies and the current state (Bui, Venkatesh, & West 2000). Thus, the RB belief state can be obtained by attaching the observation node and the policy termination status nodes to the belief chain $\mathcal{C}_t$ (Fig. 4(a) and (b)). The problem of maintaining the belief state is then reduced to maintaining the belief chain $\mathcal{C}_t$. In updating the belief chain from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$, we note that the conditioning variables include the current state s_t and all the policy termination nodes e_t^k, which cut all the loops in the transition network from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$. Furthermore, by definition, the policies from level $l_t + 1$ and higher do not terminate. Thus only the lower part of the chain $\mathcal{C}_t$ (from level l_t and below) needs to be updated. Consequently, the complexity of updating the belief chain is linear to l_t, and since $\Pr(l_t = l)$ is exponentially small w.r.t. l, the updating complexity on average is $O(\sum_l l/exp(l))$ and does not depend on the depth of the policy hierarchy K. In terms of the number of policies, the complexity is linear to the size of a policy node in the belief chain, i.e. to the maximum number of policies at one level which are applicable at the current state. It does not depend on the size of the state space since full observability is assumed.

The sampling step In sampling the RB variables, we note that in our case, the sequence of RB variables are not Markov. Thus, an extension of the existing Rao-Blackwellisation method (Doucet *et al.* 2000) is required to handle the non-Markov RB variables. We first need perform evidence reversal (ER) on the structure of $\mathcal{B}_t$ to obtain the network $\mathcal{B}_t^{er} = \Pr(\pi_t^{all}, s_t, l_t \mid \tilde{l}_{t-1}, \tilde{s}_{t-1}, o_t)$ (Fig 4(c)). The samples for s_t and l_t can then be obtained by forward sampling on $\mathcal{B}_t^{er}$. This will give us sampled values for all the nodes of $\mathcal{B}_t^{er}$. Since l_t by definition is the highest level of policy termination, the sampling can stop at the first level k such that $e_t^k = F$. We can then assign l_t the value $k-1$. All the unnecessary samples for the policy nodes obtained along the way are discarded. The weight of the new sample is $\Pr(o_t \mid \tilde{l}_{t-1}, \tilde{s}_{t-1})$ and is obtained as a by-product of the ER step.

[2] By sequential importance sampling, we refer to the SIS algorithm and its improvements which include an extra re-sampling and Monte-Carlo sampling step. These improvements of the SIS algorithm can all be used in conjunction with the Rao-Blackwellisation technique.

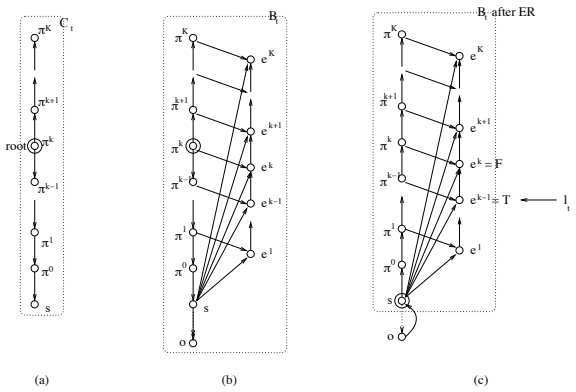

Figure 4: Manipulating the belief state

Begin
For $t = 0, 1, \ldots$
  For each sample $i = 1, \ldots, N$
    /* sampling step */
    Sample $s_t^{(i)}, l_t^{(i)}$ from $\mathcal{B}_t^{(i)}(s_t, l_t \mid o_t)$
    Update weight $w^{(i)} = w^{(i)} \mathcal{B}_t^{(i)}(o_t)$
    /* exact step */
    Update the belief chain $\mathcal{C}_{t+1}^{(i)}$ from $\mathcal{C}_t^{(i)}$
    Compute the new belief state $\mathcal{B}_{t+1}^{(i)}$ from $\mathcal{C}_{t+1}^{(i)}$
    Compute $h^{(i)} = \mathcal{C}_{t+1}^{(i)}(\pi_{t+1}^k)$
  Compute the estimator $\Pr(\pi_{t+1}^k \mid \tilde{o}_t) \approx \hat{f} = \sum_{i=1}^N h^{(i)} \tilde{w}^{(i)}$
End

Figure 5: RB-SIS for policy recognition

The overall algorithm is given in Fig. 5. The complexity at each time t is $O(NK)$, where N is the number of samples used. In comparison with SIS methods such as the likelihood weighting with ER (Kanazawa, Koller, & Russell 1995), the RB-SIS has the same order of computational complexity. However, while the SIS methods need to sample every layers of the AHMM, the RB-SIS method only needs to sample two sequences of variables $\tilde{s}_t, \tilde{l}_t$, and avoid having to sample the K policy sequences $\{\tilde{\pi}_t^k\}$. After Rao-Blackwellisation, the dimension of the sample space becomes much smaller, and more importantly, does not grow with K. As a result, RB-SIS avoids the the problems of sampling in high dimensional space of SIS, and the accuracy of the approximation by the RB-SIS method does not degrade when K is large.

Factored representation of the state space

In many cases, the state space S is the Catersian product of many state variables representing independent properties of

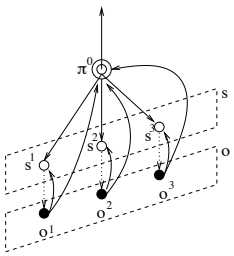

Figure 6: Evidence reversal with a factored state space

a state: $s = (s^1, s^2, \ldots, s^M)$. In this case, factored representation (Boutilier, Dearden, & Goldszmidt 2001) can be used to represent any probability distribution conditioned on the state, e.g. the policy selection function σ_{π^k}, the policy termination model β, and the transition probability of an action σ_{π^0}. This helps to get around the exponential dependency on M in the specification of a policy hierarchy.

Similarly, the representation of the RB belief state $\mathcal{B}_t$ can also make use of a factored representation of the state space. Care must be taken, however, when performing evidence reversal on $\mathcal{B}_t$. In the ER step discussed previously (Fig. 4(c)), we need to compute the marginal distribution $\Pr(s_t)$. Without conditioning on the current action π_t^0, the factored representation cannot be utilised since the individual state variables $\{s_t^m\}$ are no longer independent. We thus need to always keep the specification of the distribution of the current state conditioned on the current action, not vice versa. This can be achieved by first positioning the root of the chain $\mathcal{C}_t$ at π_t^0, and then reverse the evidence from o_t to both π_t^0 and s_t. Fig. 6 illustrates this evidence reversal procedure, assuming that at time t we have an observation node o_t^m for each state variable s_t^m. All the parameters on the new links and the marginal $\Pr(o_t)$ (which gives the sample weight) can be computed efficiently without the exponential dependency on M.

Conclusion and Discussion

In summary, we have presented an approach for on-line plan recognition under uncertainty using the AHMM as the model for the execution of a stochastic plan hierarchy and its noisy observation. By considering a general policy hierarchy, including policies with non-deterministic terminating conditions and a factored representation of the state space, we have generalised the result in (Bui, Venkatesh, & West 2000) and significantly widen the applicability of the AHMM model. We show that the hybrid algorithm in (Bui, Venkatesh, & West 2000), a variant of the Rao-Blackwellised Sequential Importance Sampling (RB-SIS) method, can be extended to work with a general policy hierarchy. The complexity of RB-SIS for policy recognition only depends linearly on the number of levels K in the policy hierarchy, while the sampling error does not depend on K.

These results show that while the stochastic process for representing the execution of a plan hierarchy can be complex, they exhibit certain conditional independence properties that are inherent in the dynamics of the planning and acting process. These independence properties, if exploited, can help to reduce the complexity of inferencing on the plan execution stochastic model, leading to feasible and scalable algorithms for on-line plan recognition in noisy and uncertain domains. The key to this exploitation, as we have shown in the paper, is a combination of recently developed techniques: compact representations for Bayesian networks (context-sensitive independence, factored representations), and hybrid DBN inference which can take advantage of these compact representations (Rao-Blackwellisation).

In our current work, we are investigating the use of Joint AHMMs to the represent the plan execution model for a team of agents. In the simplest joint model, one could simply replace joint action for action, joint policy for policy, joint state for individual state to obtain a team policy execution model. However this method would quickly become intractable when we have many agents in the team since the size of a joint variable (policy, action or state) would grow exponentially. It also fails to utilise the fact that once a joint plan is agreed upon, most of the time, the individual plans are carried out by the agents independently. For example, consider a joint plan involving agent A opening the door to let agent B in. At first both A and B would need to be at the door. While A is walking towards the door, the current location of B becomes completely independent. It is only relevant when A is already at the door so that A can decide whether it should open the door next. Fortunately, this type of independency can be captured as a form of CSI similarly to the type of CSI present in the AHMM. It can help to separate the execution model for each individual agent, and only joining them at the appropriate state. We expect that a Rao-Blackwellised sampling inference method can be used for the Joint AHMMs to take advantage of these CSI properties, making the inference task tractable.

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.

Bauer, M. 1994. Integrating probabilistic reasoning into plan recognition. In *Proceedings of the Eleventh European Conference on Artificial Intelligence*.

Boutilier, C.; Friedman, N.; Goldszmidt, M.; and Koller, D. 1996. Context-specific independence in Bayesian networks. In *Proceedings of the Twelveth Annual Conference on Uncertainty in Artificial Intelligence*.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2001. Stochastic dynamic programming with factored representations. *Artificial Intelligence*. to appear.

Bui, H. H.; Venkatesh, S.; and West, G. 2000. On the recognition of abstract Markov policies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-2000)*.

Casella, G., and Robert, C. P. 1996. Rao-Blackwellisation of sampling schemes. *Biometrika* 81–94.

Charniak, E., and Goldman, R. P. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* 64:53–79.

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)*.

Doucet, A.; de Freitas, N.; Murphy, K.; and Russell, S. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*.

Doucet, A.; Godsill, S.; and Andrieu, C. 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*.

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*.

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.

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.

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*.

Pynadath, D. V., and Wellman, M. P. 2000. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Sixteenth 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.

van Beek, P. 1996. An investigation of probabilistic interpretations of heuristics in plan recognition. In *Proceedings of the Fifth International Conference on User Modeling*, 113–120.