ICT Technical Report ICT-TR-01-2004

Decision-Theoretic Approach to Plan Recognition

Wenji Mao Jonathan Gratch

Abstract In this report, first we give a survey of the work in plan recognition field, including the evolution of different approaches, their strength and weaknesses. Then we propose two decision-theoretic approaches to plan recognition problem, which explicitly take outcome utilities into consideration. One is an extension within the probabilistic reasoning framework, by adding utility nodes to belief nets. The other is based on maximizing the estimated expected utility of possible plan. Illustrative examples are given to explain the approaches. Finally, we compare the two approaches presented in the report and summarize the work.

1. Introduction

1.1 The Plan Recognition Problem

The plan recognition problem refers to the task of inferring the plan (or plans) from the observed behavior of the actor. The recognition process involves a mapping from the observed action sequence into some plan representation that identifies the goal of the plan. Since there might be multiple plans (i.e. hypotheses) available to explain the observations, the key challenge is to disambiguate among competing hypotheses.

Schmidt, Sridharan and Goodson [1978, 1976] are the first to identify plan recognition as a problem in its own right. Their psychological experiments, together with the experiment by Cohen, Perrault and Allen [1982] provide evidence that humans do infer the plans and goals of other agents and use these hypotheses in subsequent reasoning.

1.2 Assumptions

Clearly, the nature of the plan recognition process will depend on the assumptions of action sequence, plan representation and the role the observed agent plays in the recognition task.

Cohen, Perrault and Allen [1982] distinguish between two kinds of plan recognition, *keyhole* and *intended* recognition. In keyhole recognition, the observed agent does not attempt to impact the recognition process, as if the recognizer observes the agent through a "keyhole". In intended recognition, the observed agent deliberately performs actions to help the recognition. The latter is typical in cooperative environment. Geib and Goldman [2001] propose a third class of recognition, *adversarial* plan recognition. In adversarial recognition, the observed agent attempts to thwart the recognition, typically in competitive environment (e.g. computer games). Among the three kinds of plan recognition, keyhole recognition is the most common, without any assumption of the recognized agent's role in the recognition process.

There is also a wide range of assumptions regarding what the recognizer observes. Many approaches assume a *fully-observable* action sequence, starting from the first action in the plan. This assumption is typical in user modeling, where it is natural to assume that a software system can observe all the actions the user performs and the order they occur. More general approaches allow for holes in the sequence (*partially-observable* action sequence), where some actions may have gone unobserved, or some actions themselves are as unobservable, but with some observable effects [Pynadath and Wellman, 1995].

For generality, this report will focus on *keyhole* recognition with *partially-observable* action sequence.

1.3 Previous Work

Kautz and Allen [1986, 1990, 1991] present the first formal theory of plan recognition. In their theory, every observed action is part of one or more top-level plans, and the plan recognition task is to minimize the set of top-level actions sufficient to explain the observed actions. Hence the recognition problem becomes a nonmonotonic deduction of minimal top-level actions. They use McCarthy's circumscriptive theory for the formal description. Plans are represented as plan graphs, with top-level actions as root nodes and other actions as nodes depending on higher-level ones.

Kautz and Allen's work takes minimal covering set as a principle and ignores the priors of plans. Some plans are inherently more likely than others, given that they all correctly explain the observed actions. It is also not clear whether minimal covering set can be a principle of abduction, for example, in some domains, two common causes may better explain the evidence than a single uncommon one. Since plan recognition involves abduction, it would best be done using probabilistic inference and ranking different hypotheses by their probability values.

Charniak and Goldman [1989, 1991, 1993] build the first probabilistic model of plan recognition. They use a quantifier-free first-order language as representation and belief nets for plan inference. The random variables in the belief net are propositions, whereas the root nodes are hypotheses about agent's plan. The posterior probability of each hypothesis is computed by propagating the values from the evidence in the net. Charniak and Goldman apply their plan recognition system to understanding a charater's actions in a story.

However, the plan language Charniak and Goldman employ is a predicate-calculus-like representation based on collections of actions. The representation is suitable for modeling hierarchical action descriptions (i.e. part-of and is-a) as in a story, but it does not support many other features of plan representation (e.g. sequences of actions). Huber, Durfee and Wellman [1994] use PRS as a general-purpose language for plan specification, which supports sequencing, subgoaling, conditional plans, etc. This allows plan recognition reuses the same plan representation as that for planning. They give the mapping from PRS specification to belief nets, and apply the approach to coordinating multi-agent team.

More recently, Pynadath and Wellman [2000] proposed a plan recognition method that is both probabilistic and based on parsing. They represent plan libraries as a probabilistic-state-dependent grammars (PSDGs). The language they use is more restrictive than probabilistic context-free grammars, but they claim it is more efficient in mapping to dynamic belief nets.

Besides the Bayesian probabilistic models mentioned above, there are probabilistic approaches based on other theories, for example, Carberry [1990] and Bauer [1995] use Dempster-Shafer theory in plan recognition to distinguish lack of evidence for a proposition (i.e. unknown) from knowing evidence against the proposition.

Various plan representation has been utilized in these approaches, such as action taxonomies [Kautz & Allen], associative networks [Charniak & Goldman], PRS [Huber, Durfee & Wellman], context models [Carberry], etc. Among them, most rely on additional plan structures to specifically support the recognition task. These additional structures used are different from the plan structures normally used for planning.

For generality and reuse of knowledge, it is important that plan recognition and planning share the same representation.

1.4 Plan Recognition: the Missing Part

- Pure action-level recognition

Current plan recognition systems reason about plan probabilities in terms of observed actions. Discrimination among competing plan hypotheses is done by comparing plans at the level of agent's actions. World states are not considered in disambiguation. (Some work, e.g. [Pynadath & Wellman, Geib & Goldman], considers action effects when actions themselves are as unobservable, but the purpose is still to identify actions in use for the recognition process.)

- State desirability is not considered

Pure action-level recognition does not consider state desirability. In the simplest case, when each plan has only one desirable outcome (i.e. the goal itself in the plan), this implies that current plan recognition approaches treat all the goals/plans as of the same degree of desirability. In the more complex cases, there might be more than one desirable/undesirable outcome in a plan. Actions may also have non-deterministic effects, and different outcomes may occur with uncertainty.

- Outcome utilities are important in real-world applications

In many real-world applications, an agent who is planning for a course of actions usually takes into account that actions may have several different outcomes. Some outcomes are more desirable than the others, so the planning agent must balance between different possible outcomes to maximize the expected utility of goal attainment.

Plan recognition can be viewed as modeling the decision-making strategy of another agent. While current probabilistic approaches capture the fact of how well the observed evidence supports a particular plan, the missing part is the utility computation. This report is aiming at proposing possible solutions to refine the problem.

The remainder of the report is organized as follows. In section 2, we introduce the plan representation adopted in this work. Section 3 illustrates a motivating example from MRE virtual training scenario. In section 4, we propose two approaches to extend the probabilistic models by incorporating utility nodes into belief nets. The limitations of probabilistic approaches in general are discussed in section 5. Then in section 6, we present another view of plan recognition, which is based on maximizing the expected utility of hypothesized plan. Finally, in section 7 and 8, we show the interrelation of planning and plan recognition, and summarize the work.

2. Plan Representation

2.1 Relaxation of Classical STRIPS Representation

The plan representation we adopt is an extension of classical STRIPS representation, allowing probabilistic and conditional effects, and abstract actions.

- Probabilistic STRIPS operators

In classical STRIPS representation, each operator consists of a set of preconditions that must hold if the action is to be performed, and a set of effects that describe how the world would change if the action were executed. A probabilistic STRIPS operator extends the classical STRIPS representation in two ways. First, it allows for actions to have non-deterministic effects, and second, the effects of actions are not always known with certainty.

- Conditional non-deterministic effects

Since actions may have non-deterministic effects with uncertainty, in the case of action with conditional effects, when the operator is applied, based on the conditions that hold, action effects are represented in terms of conditional probability.

- Hierarchical decomposition

Hierarchical decomposition extends the STRIPS language to allow for abstract operators. An abstract operator can be decomposed into a set of steps, each of which is either a primitive operator (i.e. an action that can be directly executed by the agent) or another abstract operator. There may be more than one way to decompose an abstract action hierarchically, and each way of decomposition consists of a sequence of abstract or primitive sub-actions.

2.2 Utility Types

- State/Outcome utility

A (world) state can be viewed as a description of the world after an action is executed. The utility of the state (or outcome) is computed by summing the utilities of all individual action effects (which can be positive, negative or zero). In the case that several action effects have one utility value, we can partition the set of action effects into several groups. The utility of the state is the sum of the utility of each group.

- Plan utility

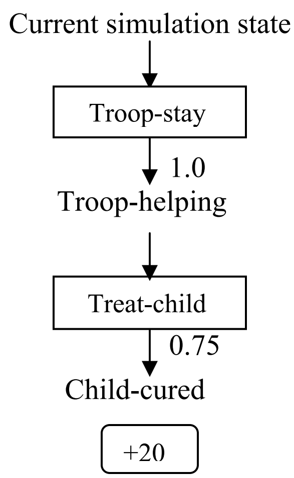
Similar to DRIPS [Haddawy & Suwandi, 1994], plan utility is computed based on an abstraction hierarchy of operators. For our purpose, since possible plans are given, we compute the exact value of plan utility, rather than estimate a range of utility values for searching the plan space.

3. Motivating Example

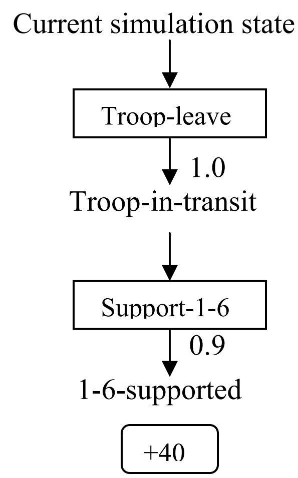
Here is an example from the MRE (Mission Rehearsal Exercise) system. In the scenario, after the accident happened, the child was wounded. The child's mother observed the troop's actions, trying to infer the troop's plan and subsequent actions.

The mother has a simplified model of the troop (comparing with the actual task model of the troop). She keeps two likely plans of the troop in her mind. Plan *Render-assistance* is composed of *Troop-stay* and *Treat-child*. Plan *support-inspection* consists of *Troop-leave* and *Support-eagle-1-6*. *Troop-stay* and *Troop-leave* are primitive actions (*Treat-child* and *Support-eagle-1-6* are actually abstract actions, but simplified here). *Troop-helping*, *Troop-in-transit*, *Child-cured* and *1-6-supported* are world states resulted from the action execution. For simplicity, the outcomes with zero utility values are omitted from the plan representation below.

Plan 1: **Render-assistance**:



Plan 2: **Support-inspection**:



The first step in creating the belief net is to create a random variable representing the top-level plan. The rest of the variables, dependency arcs and probability values all provide evidence for or against the proposition that this top plan is being pursued by the observed agent. We also need to create a random variable for each action in the plan. Because it is the adoption of the top plan that causes the execution of these actions, there is a dependency arc from the top plan to each of the action node in the net. Besides, we add evidence variables to represent the dependencies between actions and observed evidence.

Assume initially, plan *Render-assistance* and plan *support-inspection* have the same prior probability, but different outcome utilities and probabilities to achieve outcomes as shown in the plans. As the scenario proceeds, the mother observed half of the troop stayed and half left. Since the observed action equally support the two plans, Bayesian probabilistic reasoning shows that the two plans have the same posterior probability. The belief net computation is shown in *Figure 1* and *Figure 2*.

But in fact, the outcome of *Support-inspection* is more desirable to the troop, and this outcome is more likely to occur. Even if the two plans have the same prior and posterior probabilities, it is reasonable to infer that the troop is more likely to support inspection. The current probabilistic approaches to plan recognition could not make this distinction, because states and state desirability are not considered in the recognition process (note that neither outcome utility nor outcome probability is used).

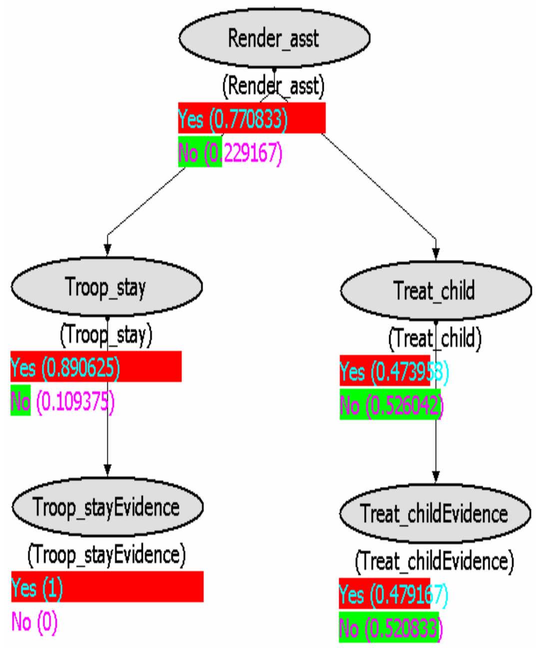


Figure 1. Belief Net of Plan "Render-assistance"

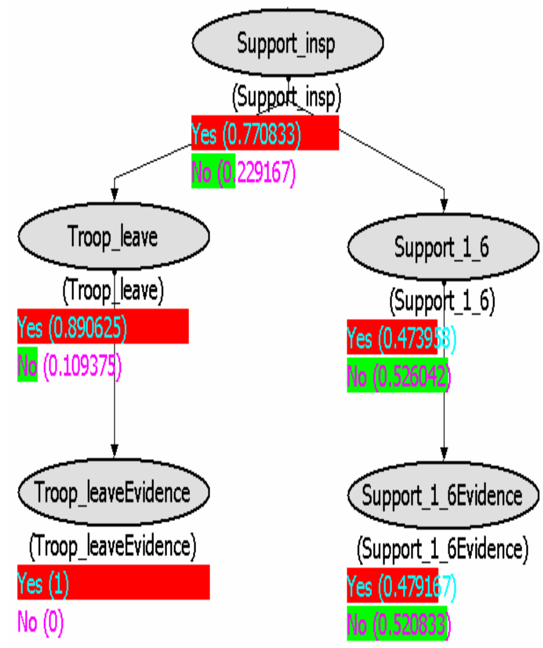


Figure 2. Belief Net of Plan "Support-inspection"

4. Extending Belief Nets to Incorporate Utility Nodes

Two types of utility nodes can be incorporated into belief nets, depending on what kind of utility we want to use to influence the probabilistic inference. One is plan utility, and the other is outcome utility. Some decision-theoretic planning techniques can be reused here, for example, computation of expected utility of a plan and computation of possibility of a plan/goal success. We can add new utility nodes to belief nets and use the computed values as evidence to adjust the probability distributions so as to take outcome desirability into consideration.

4.1 Plan Utility as Evidence

A straightforward solution is to directly incorporate plan utility node into the belief net. The process involves the following steps:

- Create a variable representing plan utility in the belief net, and compute the expected utility of the plan

- Add a dependency arc from the plan utility node to the top-level plan, and construct the CPT (conditional probability table) for the top plan
- Use the computed utility value of the plan as evidence in the belief net

The idea is to use the computed plan utility value as evidence to adjust the prior probability of the top-level plan in the belief net. For the example in *Section 3*, the *priors* of *Plan 1* and *Plan 2* are 0.5. After computation, the expected utilities of *Plan 1* and *Plan 2* are 15 (moderately high) and 36 (very high), respectively. We can construct a CPT for top-level plan as follows:

Plan-Utility (*U*)	P(*Plan*)	P(*Not*(*Plan*))
Very high (U>30)	0.9	0.1
Fairly high (30>=U>20)	0.8	0.2
Moderately high (20>=U>10)	0.7	0.3
Slightly high (10>=U>1)	0.6	0.4
Middle (1>=U>=-1)	0.5	0.5
Slightly low (-1>=U>-10)	0.4	0.6
Moderately low (-10>=U>-20)	0.3	0.7
Fairly low (-20>=U>-30)	0.2	0.8
Very low (-30>=U)	0.1	0.9

Let α be the adjustment ratio. ρ is the prior probability of plan. S is the scale used for discrimination, and n is the total number of probability values. In the table above, ρ=0.5, S=10, n=9 and α=0.1.

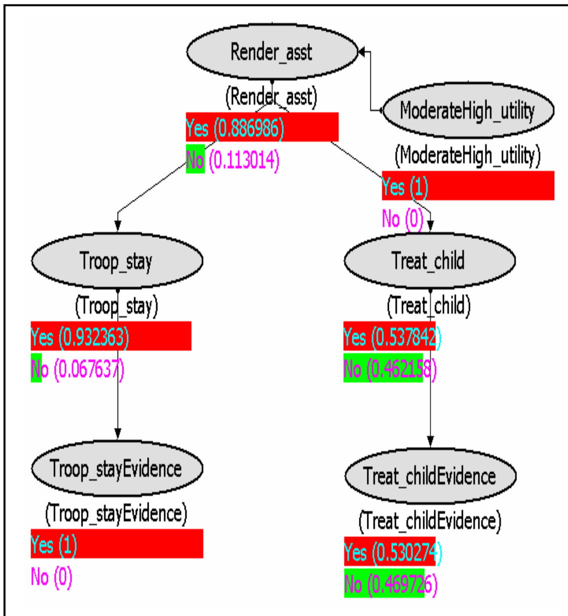


Figure 3. Belief Net of Plan 1 after Adjustment

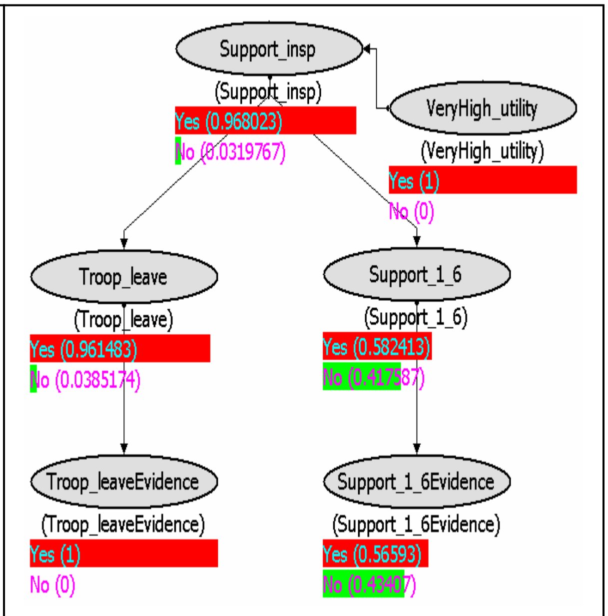


Figure 4. Belief Net of Plan 2 after Adjustment

The adjusted prior of plan is given by:

$$\rho' = \begin{cases} \rho \times (1 + \alpha \times \lceil U/S \rceil) & if \quad |U| \le S \times \lceil (n-1)/2 \rceil \\ \rho \times (1 + \alpha \times \lceil (n-1)/2 \rceil) & if \quad U \succ S \times \lceil (n-1)/2 \rceil \\ \rho \times (1 - \alpha \times \lceil (n-1)/2 \rceil) & if \quad U \prec -S \times \lceil (n-1)/2 \rceil \end{cases}$$

The higher the value of α, the more the probability of plan is influenced by plan utility. α is bounded by:

$$0 \prec \alpha \prec \frac{\min(\rho, 1-\rho)}{\lceil (n-1)/2 \rceil}$$

4.2 Incorporating Outcome Utility and Probability

Instead of adjusting the utility of top plan, a more detailed treatment is to associate outcome utility nodes with the relevant actions in the belief net, and use the positive/negative outcome utilities to increase/decrease the conditional probability of the action nodes. The adjusted probability values are then propagated to the top plan through the net.

The process involves the following steps:

- For each outcome with non-zero utility, create a variable representing outcome utility in the belief net, and compute the probability of outcome occurrence
- Add a dependency arc from the top plan to each outcome utility node, and use the computed probability value to construct the CPT for the node
- Add dependency arcs from each outcome utility node to each associated action node, and construct the CPT for each associated action
- Use the non-zero utility value of each outcome as evidence in the belief net

The idea of computing the probability of outcome occurrence is similar to that of computing the probability of plan/goal success in decision-theoretic planning [Blythe, 1999]. Typically, forward projection or more compact structure – belief nets are used for the computation.

Which action is associated with an outcome utility node? The primitive action that directly leads to the outcome should be associated. Besides, all the actions in the plan that causally support the execution of this primitive action should be associated. For example, in plan *Render-assistance*, action *Treat-child* is directly associated with the outcome *Child-cured*. Since the effect of action *Troop-stay* enables action *Treat-child*, *Troop-stay* should also be associated with the outcome. For actions with conditional effects, the treatment is similar.

The outcome utility node is used to adjust the conditional probability of actions. The formula is similar to those in plan utility case, except that we need an additional parameter *d*, to describe the distance between the action and the outcome. For example, in plan *Render-assistance*, the distance between action *Treat-child* and outcome *Child-cured* is

d=1, whereas the distance between *Troop-stay* and *Child-cured* is d=2. We use the adjustment ratio α/d, although one can think of other forms.

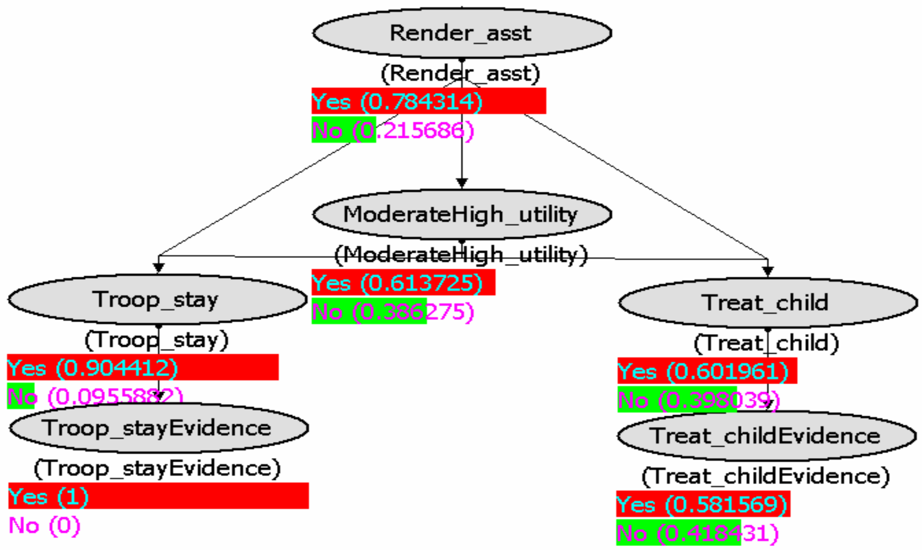


Figure 5. Belief Net of "Render-assistance" after adding outcome utility node

5. Discussion of Limitations

The approach of extending belief net to incorporate utility nodes forces the probabilistic reasoning to prefer plans with desirable outcomes and less prefer plans with undesirable outcomes. Though this is an improvement comparing with pure probabilistic approaches, the extension is still within the probabilistic reasoning framework. So it suffers from the common weaknesses of probabilistic approaches. As with any system based on Bayesian inference, it requires large number of prior and conditional probabilities. There is no good answer for where these numbers come from.

Using Bayesian inference, it seems that the result of probability computation is not sensitive to the order of observed actions. In other words, if the actions occur in a totally different order, belief net computation will draw exactly the same conclusion. This is a limitation in some applications where orders of actions are important. For incremental plan recognition, to accommodate new observations in the recognition, the update of belief net is known to be exponential of the number of entries in the CPTs.

Since the plan recognition problem is the modeling of decision-making strategy of another agent, it seems more reasonable to assume a rational agent will adopt a plan that maximizes the expected utility (given the evidence so far) rather than using probability as the only criterion.

Plan recognition based on maximizing expected utility has some key advantages. First, it is suitable for modeling the decision-making process of the observed agent, and compatible with the idea of decision-theoretic planning. Second, it eases the burden of defining large number of priors as in previous probabilistic models. Finally, it can make better use of action theory (i.e. knowledge about actions, their preconditions and effects), which is not fully explored in previous probabilistic approaches.

6. Alternative View of Plan Recognition: Maximizing Expected Utility

In order to apply the approach to the problem, besides a probabilistic action representation introduced in *Section 2*, the following information is also needed for recognition.

- Prior probabilities of preconditions of actions
- Execution probability of an action given all its preconditions are satisfied

The prior probabilities of ground literals can be obtained from calculating a large sampling set of the domain. The execution probability captures the likelihood of success/failure of action execution given preconditions are true. We denote the execution probability of an action A as $exec\_prob(A)$. In probabilistic action representation, actions can have deterministic or non-deterministic effects. We denote the probability of an effect e of an action A as $effect\_prob(A, e)$.

6.1 State Changes via Action Observation

E is the evidence. When an action A is observed, we can also infer the following state probabilities based on action theory ("rationality assumptions"):

- For $\forall x(x \in \text{preconditions}(A) \wedge x \notin \text{del\_effects}(A))$, $P(x|E) = 1.0$
- For $\forall x(x \in \text{add\_effects}(A))$, $P(x|E) = \text{effect\_prob}(A, x)$
- For $\forall x(x \in \text{del\_effect}(A))$, $P(x|E) = 1.0 - \text{effect\_prob}(A, \neg x)$

6.2 Expected Utility Computation

For each possible plan, the computation of expected utility is based on the evidence of observed actions as well as the evidence of state changes.

If an effect x is observed, then $P(x|E) = 1.0$. If an action A is observed, then $P(A|E) = 1.0$, else the probability of A given E is computed as

$$P(A \mid E) = \Big(\prod_{x \in preconditions(A)} P(x \mid E) \Big) \times exec\_prob(A)$$

Let O_i be the set of outcome of a plan P_i, $o_j \in O_i$. A_1, …, A_k is the action sequence in P_i that leads to o_j. The probability of o_j is computed as

$$P(o_j \mid E) = \Big(\prod_{i=1,\ldots,k} P(A_i \mid E) \Big) \times effect\_prob(A_k, o_j)$$

The estimated expected utility of plan P_i given E is computed as

$$EU(P_i \mid E) = \sum_{o_j \in O_i} (P(o_j \mid E) \times utility(o_j))$$

In hierarchical task representation, if the abstract action is an *And*-node (i.e. only one way to decompose the action), the expect-utility of the abstract action is the sum of the utilities of its sub-actions. If the abstract action is an *Or*-node (i.e. multiple way to decompose the task), the expected-utility of the abstract action is the maximum of the utilities of its sub-actions (because we assume the agent tries to maximize the expected utility). The utility of the root (action) node of the hierarchical plan structure is the expected utility of the abstract plan.

6.3 Illustrative Example

Consider again the plans in *Section 3*. Assume initial states are *Troop-at-aa* (i.e. *Troop-at-accident-area*) and *Child-at-aa* (i.e. *Child-at-accident-area*) (the effect of *Troop-leave* is actually a delete effect of *Troop-at-aa*). Initially, the prior probabilities and execution probabilities are as follows.

P(*Troop-at-aa*) = *P*(*Child-at-aa*) = 0.8
P(*Troop-helping*) = *P*(*Troop-in-transit*) = 0.5
exec_prob(*Troop-stay*) = *exec_prob*(*Troop-leave*) = 0.95
exec_prob(*Treat-child*) = *exec_prob*(*Support-1-6*) = 0.95

From the graphs in *Section 3*, we also know
effect_prob(*Troop-stay*, *Troop-helping*) = 1.0
effect_prob(*Troop-leave*, *Troop-in-transit*) = 1.0
effect_prob(*Treat-child*, *Child-cured*) = 0.75
effect_prob(*Support-1-6*, *1-6-supported*) = 0.9

Given that the observation equally supports two actions, *Troop-stay* and *Troop-leave*, i.e. *P*(*Troop-stay*|*E*) = *P*(*Troop-leave*|*E*) = 1.0, we have
P(*Child-at-aa*|*E*) = 1.0
P(*Troop-helping*|*E*) = 1.0
P(*Troop-in-transit*|*E*) = 1.0

Now compute the probabilities of *Treat-child* and *Support-1-6* given the evidence
P(*Treat-child*|*E*) = 0.95
P(*Support-1-6*|*E*) = 0.95

Now compute the outcome probabilities
P(*Child-cured*|*E*) = 0.7125
P(*1-6-supported*|*E*) = 0.855

Now compute the estimated expected utilities of $plan_1$ and $plan_2$ given current observation

$EU(Plan_1|E) = 14.25$
$EU(Plan_2|E) = 34.2$

So the algorithm recognizes that the troop is pursuing the plan *Support-eagle-1-6*.

Although the example here is oversimplified, for the more complex examples that include conditional probabilities, multiple outcomes, abstract actions, etc, the algorithm is applicable the same way.

6.4 Difference from Previous Probabilistic Approaches

- Using maximum expected utility as criterion

Previous probabilistic models view plan recognition as a kind of abduction, and use Bayesian rules to compute the best plan candidate. The approach here views plan recognition as modeling the decision-making process of another agent, takes agent's preferences into consideration, and uses maximum expected utility as criterion for disambiguation.

- Focusing on influence of actions on states

Previous probabilistic approaches are action-level recognition in general, focusing on how well the observed (or unobserved) actions support the hypothesis that a particular plan is being pursued by the observed agent. The approach here focuses on the influence of actions on state changes, making use of the knowledge about actions, preconditions and effects that is typically available in a plan-based system.

- Sensitive to the order of the observed actions

In the approach, observed actions are used as evidence to change the probabilities of action preconditions and effects. The state changes are closely coupled to the action theory. So the resulting computation is sensitive to the order of observed actions.

7. Planning and Plan Recognition

Plan recognition is a problem closely related to planning. In planning, the goal is given. The task of planning is to generate plans to achieve the goal. In plan recognition, possible plans are given. The task of plan recognition is to find the goal/plan of the agent. That's way plan recognition is generally perceived as the inverse problem of planning.

Planning helps construct the plan library used by plan recognition. Plan recognition helps multi-agent planning and coordination, especially when explicit communication is impossible or too expensive. Planning and plan recognition not only can share the same plan represent, they can also benefit each other by sharing some underlying techniques, for example, in this report, we utilize the ideas of decision-theoretic planning to help plan recognition task.

In applications like user modeling and user-adapted interaction, mixed-initiative systems, natural language understanding and some help systems, it is also interesting to explore how to combine both planning and plan recognition in the same system, so as to allow the system interleaving between planning and inferring plans, depending on the task at hand. In such situation, planning and recognition can also share some intermediate results. Some work on active acquisition of user models suggests combining plan recognition, domain planning and dialogue planning into the same system architecture [Wu, 1991].

8. Summary

The first part of the report is a survey of the work in plan recognition field, including the evolution of different approaches, the assumptions have been made, and strength and weaknesses of different approaches. The second part of the report is our extension to the problem, based on the known utilities of outcomes. Two solutions are proposed. One is the extension of belief nets, to explicitly incorporate utility nodes and allow them to influence Bayesian reasoning by adjusting prior or conditional probabilities in CPTs. This extension is within the probabilistic reasoning framework. Another solution is based on maximizing the expected utility of possible plan given current evidence. The observed actions are used to change the probabilities of states, which are then used to compute the probabilities of action execution and outcome occurrence. As a criterion for disambiguation, maximizing expected utility is a different view from choosing the plan with the highest probability as in previous probabilistic models.

Acknowledgement

This report was originally a course project report for *CSCI*541. The authors thank the three instructors Jim Blythe, Yolanda Gil and Jose-Luis Ambite for lecturing the course. Thank Jim Blythe and David Pynadath in particular for the valuable discussions.

References

J. F. Allen. Recognizing Intentions from Natural Language Utterances, *Computational Models of Discourse*, the MIT Press, 1983. (Also in *AI Journal* 15(3), 1980).

M. Bauer. A Dempster-Shafer Approach to Modeling Agent Preferences for Plan Recognition. *User Modeling and User-Adapted Interaction*, 1995.

J. Blythe. Decision-Theoretic Planning. *AI Magazine*, 20(2), 1999.

S. Carberry. *Plan Recognition on Natural Language Dialogue*, the MIT Press, 1990.

E. Charniak and R. Goldman. A Semantics for Probabilistic Quantifier-Free First-Order Languages, with Particular Application to Story Understanding. *IJCAI* 1989.

E. Charniak and R. Goldman. A Probabilistic Model of Plan Recognition. *AAAI* 1991.

E. Charniak and R. Goldman. A Bayesian Model of Plan Recognition. *AI Journal*, 64(1), 1993.

P. R. Cohen, C. R. Perrault and J. F. Allen. Beyond Question Answering. *Strategies for Natural Language Processing*, Lawrence Erlbaum Associates, 1982.

C. W. Geib and R. P. Goldman. Plan Recognition in Intrusion Detection Systems. *DISCEX* 2001.

P. Haddawy and M. Suwandi. Decision-Theoretic Refinement Planning using Inheritance Abstraction. *AIPS* 1994.

M. J. Huber, E. H. Durfee and M. P. Wellman. The Automated Mapping of Plans for Plan Recognition. *UAI* 1994.

H. A. Kautz and J. F. Allen. Generalized Plan Recognition. *AAAI* 1986.

H. A. Kautz. A Circumscriptive Theory of Plan Recognition. *Intentions in Communications*, the MIT Press, 1990.

H. A. Kautz. A Formal Theory of Plan Recognition and its Implementation. *Reasoning about Plans*, Morgan Kaufmann Publishers, 1991.

D. V. Pynadath and M. P. Wellman. Accounting for Context in Plan Recognition, with Application to Traffic Monitoring. *UAI* 1995.

D. V. Pynadath and M. P. Wellman. Probabilistic State-Dependent Grammars for Plan Recognitions. *UAI* 2000.

C. F. Schmidt, N. S. Sridharan and J. L. Goodson. The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence. *AI Journal*, 11(1,2), 1978.

C. F. Schmidt. Understanding Human Action: Recognizing the Plans and Motives of other Persons. *Cognition and Social Behavior*, Lawrence Erlbaum Associates, 1976.

D. Wu. Active Acquisition of User Models: Implications for Decision-Theoretic Dialog Planning and Plan Recognition. *User Modeling and User-Adapted Interaction*, 1(2), 1991.