Incorporating Observer Biases in Keyhole Plan Recognition (Efficiently!)

Dorit Avrahami-Zilberbrand and **Gal A. Kaminka**
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{avrahad1,galk,}@cs.biu.ac.il

Abstract

Plan recognition is the process of inferring other agents' plans and goals based on their observable actions. Essentially all previous work in plan recognition has focused on the recognition process itself, with no regard to the use of the information in the recognizing agent. As a result, low-likelihood recognition hypotheses that may imply significant meaning to the observer, are ignored in existing work. In this paper, we present novel efficient algorithms that allows the observer to incorporate her own biases and preferences—in the form of a utility function—into the plan recognition process. This allows choosing recognition hypotheses based on their expected utility to the observer. We call this Utility-based Plan Recognition (UPR). While reasoning about such expected utilities is intractable in the general case, we present a hybrid symbolic/decision-theoretic plan recognizer, whose complexity is $O(NDT)$, where N is the plan library size, D is the depth of the library and T is the number of observations. We demonstrate the efficacy of this approach with experimental results in several challenging recognition tasks.

Introduction

Keyhole plan recognition (Charniak & Goldman 1993; Duong *et al.* 2005; Geib 2004) focuses on mechanisms for recognizing the unobservable state of an agent, given observations of its interaction with its environment. Most approaches to plan recognition utilize a plan library, which encodes the behavioral repertoire of observed agents. Observations are matched against this plan library in sequence.

Essentially all plan recognition techniques ignore the decision processes of the recognizing agent. Thus existing work focuses on probabilistic or heuristic ranking of recognition hypotheses, with no regards to the task for which knowledge of the plans of others is needed. As a result, low-likelihood recognition hypotheses that may carry significant gains or costs to the observer, might be ignored.

For instance, suppose we observe a sequence of Unix commands that can be explained by for some intention I or for a more common intention L. Most plan recognition systems will prefer the most likely hypothesis L, and ignore I. Yet, if the expected cost (risk) of I for the observer is high (e.g., if I is a plan to take down the computer system), then that hypothesis should be preferred when trying to recognize suspicious behavior.

We advocate a novel plan recognition approach, *utility-based plan recognition* (UPR), in which the observer folds its biases and preferences—in the form of a utility function—into the plan recognition process itself. Using UPR, the recognition process ranks recognition hypotheses based on their *expected utility to the observer*. This allows the observer, for instance, to select hypotheses based on their expected costs (e.g., in the case of a risk-averse observer), or expected gains. Unfortunately, while in principle UPR can be carried out via influence diagrams or other means, such reasoning about interactions with others is intractable in the general case (Howard & Matheson 1984; Noh & Gmytrasiewicz 2005).

We present an efficient UPR recognizer, able to carry out plan recognition in worst-case complexity of $O(NDT)$, where N is the size of a hierarchical plan library, D is the depth of the library, and T is the number of observations. This complexity is achieved by using an hybrid approach that combines an efficient symbolic plan recognizer (Avrahami-Zilberbrand & Kaminka 2005; Avrahami-Zilberbrand, Kaminka, & Zarosim 2005), with a decision-theoretic inference mechanism. We restrict these algorithms to the case of *keyhole recognition*, where the observed agent does not modify its behavior based on the knowledge that it is being observed.

We evaluate the use of the algorithms in several scenarios requiring a differentiation between neutral recognition (in which the recognition hypotheses are only ranked based on their expected likelihood), and utility-based plan recognition (UPR). Finally, we show that previous work, which introduced heuristic ranking functions for selecting hypotheses in adversarial settings, can now be recast in terms of UPR, in a principled manner.

Related Work and Motivation

There has been considerable research exploring plan recognition algorithms. Almost all of it ignores the use of utilities; we leave those aside for lack of space. Similarly, we leave aside investigations of (sequential) multi-agent decision making which take into account the utility of other's actions, but deemphasize the recognition process—which is complex in itself—necessary to establish the hypotheses underlying the decisions. Here we only address those efforts that are closely related.

Most existing work addresses utilities of the other agent's actions to itself, in contrast to our work. (Mao & Gratch 2004) has explored explicit modeling of observed agents' utilities as part of ranking recognition hypotheses. Here, equally-likely hypotheses are ranked based on the preferences of the observed agent, as expressed in its own utilities, and under the assumption of rationality. Similarly, Suzic (Suzic 2005) proposes a generic framework for tactical plan recognition using Multi-Entity Bayesian Networks (MEBN). MEBN also take into account a-priori knowledge of the utility, given plans. Our work differs from (Suzic 2005; Mao & Gratch 2004) in that we consider the impact of recognition hypotheses on the observer, not the observed.

(Sukthankar & Sycara 2005) present a cost minimization approach, in which the recognizer uses behavior transition cost function to select the most parsimonious, minimal-cost, recognition hypothesis. However, the cost is to the observed agent, transitioning between behaviors, and is intended to increase recognition coherence.

More closely-related work examined reasoning about the utility of recognition hypotheses *for the observer*. (Tambe & Rosenbloom 1995) have examined the use of reactive plan recognition in simulated air-combat domains. Here, the observing agent may face ambiguous observations, where some hypotheses imply extreme danger (a missile being fired towards the observer), and other hypotheses imply gains (the opponent running away). RESC takes a heuristic approach that prefers hypotheses that imply significant costs to the observer (e.g., potential destruction). The relative likelihood of such hypotheses is ignored. While we are inspired by this work, we take a principled, decision-theoretic, approach. In the algorithms we present, the likelihood of hypotheses is combined with their utilities, to calculate the expected impact on the observer. We show that this subsumes the earlier, heuristic work.

In general, a UPR recognizer could be implemented by extending the use of plan-recognition Bayesian Networks (Charniak & Goldman 1993) to influence diagrams (Howard & Matheson 1984) or similar representations. However, the run-time complexity of inference in such representations is inhibitory for real-world cases.

A Hybrid UPR Technique

This section presents an efficient hybrid UPR technique. Here, a highly efficient symbolic plan recognizer (Avrahami-Zilberbrand & Kaminka 2005) is used to filter through hypotheses, maintaining only those that are consistent with the observations (but not ranking the hypotheses in any way). We then add a decision-theoretic layer which is run on top of the symbolic recognizer.

Efficient Symbolic Plan Recognition

We exploit SBR, a highly-efficient symbolic plan recognizer, briefly described below. The reader is referred to (Avrahami-Zilberbrand & Kaminka 2005) for details.

SBR's plan library is a single-root directed graph, where vertices denote *plan steps*, and edges can be of two types: Decomposition edges decompose plan steps into sub-steps, and sequential edges specify the temporal order of execution. The graph is acyclic along decomposition transitions.

Each plan has an associated set of conditions on observable features of the agent and its actions. When these conditions hold, the observations are said to match the plan. At any given time, the observed agent is assumed to be executing a *plan decomposition path*, root-to-leaf through decomposition edges. An observed agent is assumed to change its internal state in two ways. First, it may follow a sequential edge to the next plan step. Second, it may reactively interrupt plan execution at any time, and select a new (first) plan.

The recognizer operates as follow: First, it matches observations to specific plan steps in the library according to the plan step's conditions. Then, after matching plan steps are found, they are tagged by the time-stamp of the observation. These tags are then propagated up the plan library, so that complete plan-paths (root to leaf) are tagged to indicate they constitute hypotheses as to the internal state of the observed agent when the observations were made. The propagation process tags paths along decomposition edges. However, the propagation process is not a simple matter of following from child to parent. A plan may match the current observation, yet be *temporally inconsistent*, when a history of observations is considered. SBR is able to quickly determine the temporal consistency of a hypothesized recognized plan (Avrahami-Zilberbrand & Kaminka 2005).

At the end of the SBR process we are left with a set of *current-state hypotheses*, i.e., a set of paths through the hierarchy, that the observed agent may have executed at the time of the last observation. The overall worst-case run-time complexity of this process is $O(LD)$ (Avrahami-Zilberbrand & Kaminka 2005). Here, L is the number of plan-steps that directly match the observations; D is depth of a degenerate plan-library (i.e., a linked list). Extensions to this model address interleaved plans and limits on durations (Avrahami-Zilberbrand, Kaminka, & Zarosim 2005).

Computing the Expected Utility of an Hypothesis

After getting all *current state hypotheses* from the symbolic recognizer, the next step is to compute the expected utility of each hypothesis. This is done by multiplying the posterior probability of a hypothesis, by its utility to the observer.

We follow in the footsteps of *Hierarchical Hidden Markov Model* (HHMM) (Fine, Singer, & Tishby 1998) in representing probabilistic information in the plan library. We denote plan-steps in the plan library by q_i^d, where i is the plan-step index and d is its hierarchy depth, $1 \leq d \leq D$. For each plan step, there are three probabilities maintained:

Sequential transition. For each internal state q_i^d, there is a state transition probability matrix denoted by $A^{q^d} = (a_{i,j}^{q^d})$, where $a_{i,j}^{q^d} = P(q_j^d | q_i^d)$ is the probability of making a sequential transition from the i^{th} plan-step to the j^{th} plan-step. Note that self-cycle transitions are also included in A^{q^d}.

Interruption. We denote by $a_{i,end}^{q^d}$ a transition to a special plan step end^d which signifies an interruption of the sequence of current plan step q_i^d, and immediate return of control to its parent, q^{d-1}.

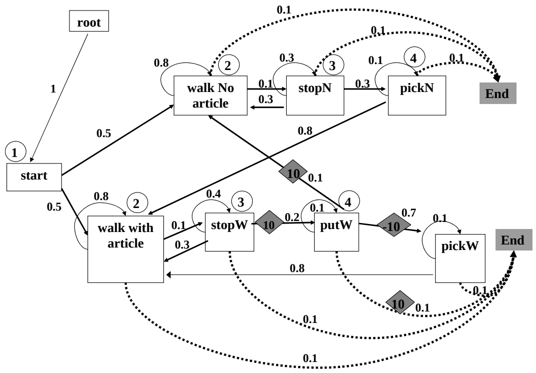

Figure 1: **An example plan library. Recognition time-stamps (example in text) appear in circles. Costs appear in diamonds.**

Decomposition transition. When the observed agent first selects a decomposable plan step q_i^d, it must select between its (first) children for execution. The decomposition transition probability is denoted $\Pi^{q^d} = \pi^{q^d}(q^{d+1}) = P(q_k^{d+1}|q_i^d)$, the probability that plan-step q_i^d will initially activate the plan-step q_k^{d+1}.

Observation Probabilities. Each leaf has an output emission probability vector $B^{q^d} = (b^{q^d}(o))$. This is the probability of observing o when the observed agent is in plan-step q^d. For presentation clarity, we treat observations as children of leaves, and use the decomposition transition Π^{q^d} for the leaves as B^{q^d}.

In addition to transition and interruption probabilities, we add utility information on the edges in the plan library. The utilities on the edges represent the cost or gains to the observer, given that the observed agent selects the edge. For the remainder of the paper, we use the term cost to refer to a positive value associated with an edge or node. As in the probabilistic reasoning process, for each node we have three kinds of utilities: (a) E^{q^d} is the sequential transition utility (cost) to the observer, conditioned on the observed agent transitioning to the next plan-step, paralleling A^{q^d}; (b) $e_{i,end}^{q^d}$ is the interruption utility; and (c) Ψ^{q^d} is the decomposition utility to the observer, paralleling Π^{q^d}.

Figure 1 shows portion of the plan library of an agent walking with or without a suitcase in the airport, occasionally putting it up and picking it up again, an example discussed below. Note the *end* plan step at each level, and the transition from each plan-step to this end plan step. This edge represent the probability to interrupt. The utilities are shown in diamonds (we omitted zero utilities, for clarity). The transitions allowing an agent to leave a suitcase without picking it up are associated with large positive costs, since they signify danger to the observer.

We use these probabilities and utilities to rank the hypotheses selected by the SBR. First, we determine all paths from each hypothesized leaf in time-stamp $t-1$, to the leaf of each of the current state hypotheses in time stamp t. Then, we traverse these paths multiplying the transition probabilities on edges by the transition utilities, and accumulating the utilities along the paths. If there is more then one way to get from the leaf of the previous hypothesis to the leaf of the current hypothesis, then it should be accounted for in the accumulation. Finally, we can determine the *most costly* current plan-step (the current-state hypothesis with maximum expected cost). Identically, we can also find the *most likely* current plan-step, for comparison.

Formally, let us denote hypotheses at time $t-1$ (each a path from root to leaf) as $W = \{W_1, W_2, ..., W_r\}$, and the hypotheses at time t as $X = \{X_1, X_2, ..., X_l\}$. To calculate the maximum expected-utility (most costly) hypothesis, we need to calculate for each current hypothesis X_i its expected cost to the observer, $U(X_i|O)$, where O is the sequence of observations thus far. Due to the use of SBR to filter hypotheses, we know that the $t-1$ observations in O have resulted in hypotheses W, and that observation t results in new hypotheses X. Therefore, under assumption of Markovian plan-step selection, $U(X_i|O) = U(X_i|W)$.

The most costly hypothesis is computed in Equation 1. We use $P(W_k)$, calculated in the previous time-stamp, and multiply it by the probability and the cost to the observer of taking this step from W_k to X_i. This is done for all i, k.

$$\hat{X}_i = \underset{X_i}{\operatorname{argmax}} \sum_{W_k \in W} P(W_k) \cdot P(X_i|W_k) \cdot U(X_i|W_k) \quad (1)$$

To calculate the expected utility $E(X_i|W_k) = P(X_i|W_k) \cdot U(X_i|W_k)$, let X_i be composed of plan steps $\{x_i^1, ..., x_i^m\}$ and W_k be composed of $\{w_k^1, ..., w_k^n\}$ (the upper index denotes depth). There are two ways in which the observed agent could have gone from executing the leaf $w^n \in W_k$ to executing the leaf $x^m \in X_i$: First, there may exist $w \in W_k$, $x \in X_i$ such that x and w have a common parent, and x is a direct decomposition of this common parent. Then, the expected utility is accumulated by climbing up vertices in W_k (by taking *interrupt* edges) until we hit the common parent, and then climbing down (by taking first child decomposition edges) to x^m. Or, in the second case, x^m is reached by following a sequential edge from a vertex w to a vertex x.

In both cases, the probability of climbing up from a leaf w^n at depth n, to a parent w^j (where $j < n$) is given by

$$\alpha_{w^n}^j = \prod_{d=n}^{j} a_{w,\text{end}}^d \quad (2)$$

and the utility is given by

$$\gamma_{w^n}^j = \sum_{d=n}^{j} e_{w,\text{end}}^d \quad (3)$$

The probability of climbing down from a parent x^j to a leaf x^m is given by

$$\beta_{x^m}^j = \prod_{d=j}^{m} \pi^{x^d}(x^{d+1}) \quad (4)$$

and the utility is given by

$$\delta_{x^m}^j = \sum_{d=j}^{m} \psi^{x^d}(x^{d+1}) \quad (5)$$

Note that we omitted the plan-step index, and left only the depth index, for presentation clarity.

Using α_w^j, β_x^j, γ_w^j and δ_x^j, and summing over all possible j's, we can calculate the expected utility (Equation 6) for the two cases in which a move from w_n to x_m is possible .

$$\begin{aligned} E(X_i|W_k) &= P(X_i|W_k) \times U(X_i|W_k) \\ &= \sum_{j=n-1}^{1} [(\alpha_w^j \cdot \beta_x^j) \times (\gamma_w^j + \delta_x^j) \times \text{Eq}(x^j, w^j)] \\ &+ \sum_{j=n-1}^{1} [\alpha_w^j \cdot a_{w,x}^j \cdot \beta_x^j] \times (\gamma_w^j + e_{w,x}^j + \delta_x^j) \end{aligned} \quad (6)$$

The first term covers the first case (transition via interruption to a common parent). Let $\text{Eq}(x^j, w^j)$ return 1 if $x^j = w^j$, and 0 otherwise. The summation over j accumulates the probability multiplying the utility of all ways of interrupting a plan w^n, climbing up from w^n to the common parent $x^j = w^j$, and following decompositions down to x^m.

The second term covers the second case, where a sequential transition is taken. $a_{w,x}^j$ is the probability of taking a sequential edge from w^j to x^j, given that such an edge exists ($a_{w,x}^j > 0$), and that the observed agent is done in w_j. To calculate the expected utility, we first multiply the probability of climbing up to a plan-step that has a sequential transition to a parent of x^m, then we multiply in the probability of taking the transition, and then we multiply in the probability of climbing down again to x^m. Then, we multiply in the utility summation along this path.

A naive algorithm for computing the expected costs of hypotheses at time t can be expensive to run. It would go over all leaves of the paths in $t-1$ and for each of these, traverse the plan library until getting to all leaves of paths we got in time-stamp t. The worst-case complexity of this process is $O(N^2T)$, where N is the plan library size, and T is the number of observations.

Efficient UPR Algorithms

We developed a set of algorithms that calculates the expected utilities of hypotheses (Equation 1) in worst-case runtime complexity $O(NDT)$, where D is the depth of the plan library (N,T as above). The algorithms are based on the observation that the structural constraints on the plan library are such, that all the paths from any path (hypothesis) true at time $t-1$, to a given hypothesis X_i, true at time t, must necessarily go through a single node S that is a part of X_i. Moreover, S is necessarily a common node to X_i and one or more paths at time $t-1$. If we can compute α_S and γ_S up to this node S, then we could propagate from it to all paths X in which it is a part, that are true at time t. In other words, we can reuse the summation α_S and γ_S for all hypotheses in which S participates.

This translates into the following procedure. We begin with the leaves of all $t-1$ hypotheses W_k ($1 \le k \le n$). We sum the utilities and multiply the probabilities while climbing up from the leaves along the hierarchy, all the way to the root, storing intermediate results in the internal nodes w^j

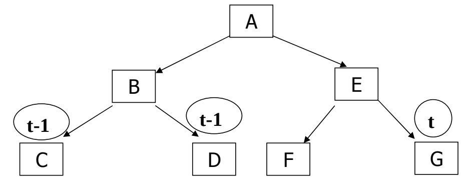

Figure 2: **Efficient UPR Example**

(plan-steps) of the hierarchy. We then look for internal nodes (i) that have a child marked at time t (i.e., w^j is a common parent, $\text{Eq}(w^j, x^j)$ is true); or (ii) that have a sequential transition to an internal node marked at time t (i.e., $a_{w,x}^j > 0$). A node (marked time t) that satisfies either of these cases is one through which one or more time t hypotheses X_i pass, i.e., a node S as above. We then propagate down the calculated probability and utility downward (β_S and δ_S).

To illustrate the efficient UPR algorithm, let us examine a portion of a plan library at figure 2. Suppose that in time $t-1$ the SBR had returned that the two plan-steps C and D are matching, and G in time-stamp t. To calculate the expected utility with the naive algorithm, we would traverse the plan library in the following manner: $E(G|C,D) = (\alpha_C^A \cdot \beta_A^G + \alpha_D^A \cdot \beta_A^G) \times (\gamma_C^A + \delta_A^G + \gamma_D^A + \delta_A^G)$. With the efficient UPR: $E(G|C,D) = [(\alpha_C^B + \alpha_D^B) \times \alpha_B^A \times \beta_A^G] \times (\gamma_C^B + \gamma_D^B + \gamma_B^A + \delta_A^G)$. Meaning that the probabilities and utilities of C and D are stored in B, so we are not traversing the plan library more then necessary.

Complexity Analysis. The run-time complexity of the algorithm is $O(NDT)$: We first propagate the $t-1$ expected utilities up the hierarchy, not visiting plans that already been visited, in worst-case time $O(N)$. Then, calculating β for different depths, for paths tagged with t, is $O(ND)$. We do this for every observation, of which there are T, thus the overall complexity is $O(NDT)$.

Note the reliance on the underlying SBR: Since the symbolic recognizer provides the possible paths at times $t-1, t$, we do not need to consider all possible paths, and can begin the propagation process directly at the leaves of paths. Hopefully, many paths are disqualified by the symbolic algorithm, due to temporal coherence; in that case, we expect performance in practice to improve significantly over the worst case complexity.

Experiments

To demonstrate the novel capabilities of UPR, and its efficient implementation as described above, we tested the capabilities of our algorithms in three different recognition tasks. The domain for the first task consisted of recognizing passengers that leave articles unattended, as in the example above. In the second task we will show how our algorithms can catch a dangerous driver that cuts between two lanes repeatedly. The last experiment intends to show how previous work, which has used costs heuristically (Tambe & Rosenbloom 1995), can now be recast in a principled manner. All of these experiments show that we should not ignore the observer biases, since the most probable hypothesis sometimes mask hypotheses that are important for the observer.

Leaving unattended articles

It is important to track a person that leave her articles unattended in the airport. It is difficult, if not impossible, to catch

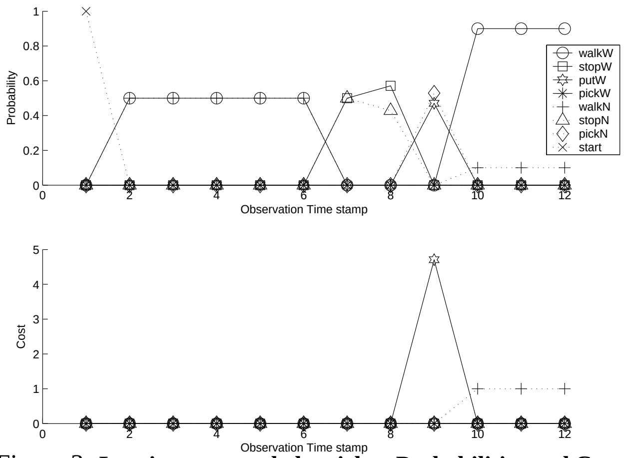

Figure 3: **Leaving unattended articles: Probabilities and Costs**

this behavior using only probabilistic information. We examine the instantaneous recognition of costly hypotheses.

We demonstrate the process using the plan library in Figure 1. This plan library is used to track simulated passengers in an airport that walk about carrying articles, which they may put down and pick up again. The recognizer's task is to recognize passengers that put something down, and then continue to walk without it. Note that the task is diffi cult because the plan-steps are hidden (e.g., we see a passenger bending, but cannot decide whether it pick something up, put something down, or neither; we cannot decide whether a person has an article when they walk).

For the purposes of a short example, suppose that in time $t = 2$, the SBR had returned that the two plan-steps marked *walk* match the observations (*walkN* means walking with no article, *walkW* signifi es walking with an article); in time $t = 3$ the two $stop$ plan steps match ($stopN$ and $stopW$), and in time $t = 4$ the plan step $pickN$ and plan step $putW$, match (e.g., we saw that the observed agent was bending). The probability in $t = 4$ will be $P(putW|stopW) = 0.5 \times 0.2 = 0.1$ (the probability of $stopW$ in previous timestamp is 0.5, then following sequential link to $putW$), and in the same way $P(pickN|stopN) = 0.5 \times 0.3 = 0.15$. Normalizing the probabilities for the current time $t = 4$, $P(putW|stopW) = 0.4$ and $P(pickN|stopN) = 0.6$. The expected utility in time $t = 4$ is $U(putW|stopW) = P(putW|stopW) \times E(putW|stopW) = 0.4 \times 10 = 4$. The expected utility of $pickN$ is zero. The expected costs, rather than likelihoods, raise suspicions of a passenger putting down an article (perhaps not picking it up).

Let us examine a more detailed example. We generated the following observations based on Figure 1: In time stamps $t = \{1 - 5\}$ the simulated passenger walks in an airport, but we can not tell whether she has an dangerous article in her possession. In time-stamps $t = \{6-7\}$ she stops, then at time $t = \{8\}$ we see her bending but can not tell whether to put or to pick something. In time-stamps $t = \{10 - 12\}$, she walks again.

Figure 3 shows the results from the recognition process. The X-axis measures the sequence of observations in time. The probability of different leaves (corresponding to hypotheses) is shown on the Y-axis in the upper graph. The

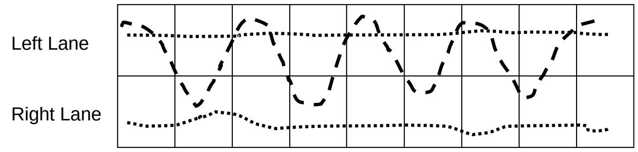

Figure 4: **Simulated trajectories for drivers.**

expected costs are shown in the lower graph. In both, the top-ranking hypothesis (after each observation), is the one whose value on the Y-axis is maximal for the observation.

In the probabilistic version (upper graph) we can see that the probabilities, in time $t = \{1 - 5\}$, are 0.5 since we have two possible hypotheses of walking. with or without an article ($walkW$ and $walkN$). Later when the person stops there are again two hypotheses $stopW$ and $stopN$. Then, in $t = \{7\}$ two plan steps match the observations: $pickW$ and $putN$, where the prior probability of $pickN$ is greater than $putN$ (after all, most passengers do not leave items unattended). As a result, the most likely hypothesis for the remainder of the sequence is that the passenger is currently walking with her article in hand *walkW*.

In the lower graph we can see a plot of the hypotheses, ranked by expected cost. At time $t = 8$ when the agent pick or put something, the cost is high (equal to 5), then in time stamp $t = \{9 - 12\}$ the top-ranking hypothesis is *walkN*, signifying that the passenger might have left an article unattended. Note that the prior probabilities on the behavior of the passenger have not changed. What is different here is the importance (cost) we attribute to observed actions.

Catching a dangerous driver

Some behavior becomes increasingly costly, or increasingly gainful, if repeated. For example, a driver switching a lane once or twice is not necessarily acting suspiciously. But a driver zigzagging across two lanes is dangerous. We demonstrate here the ability to accumulate costs of the most costly hypotheses, in order to capture behavior whose expected costs are prohibitive *over time*.

Figure shows two lanes left and right in a continuous area, divided by a gird. There are 2 straight trajectories and one zigzag trajectory from left to right lane. From each position, the driver can begin moving to the next cell in the row (straight), or to one of the diagonal cells. We emphasize that the area and movements are continuous—the grid is only used to create a discrete state-space for the plan library. Moreover, the state-space is hidden: A car in the left lane may be mistakenly observed (with small probability) to be in the right lane, and vice versa.

Each grid-cell is a plan-step in the plan library. The associated probabilities and utilities are as follows: The probability for remaining in a plan-step (for all nodes) is 0.4. The probability of continuing in the same lane is 0.4. The probability of moving to either diagonal is 0.2. All costs are zero, except when moving diagonally, where the cost is 10.

We generated 100 observation sequences (each of 20 observations) of a zigzagging driver, and 100 sequences of a safe driver. The observations were sampled (with noise) from the trajectories; with 0.1 probability, an observation would incorrectly report on the driver being in a given lane.

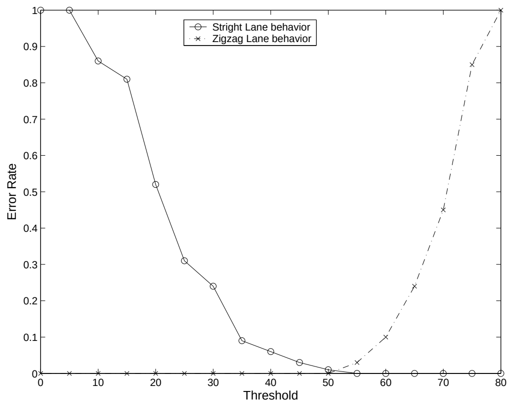

Figure 5: **Confusion error rates for different thresholds for dangerous and safe drivers.**

For each sequence of observations we accumulated the cost of the most costly hypothesis, along the 20 observations. We now have 100 samples of the accumulated costs for a dangerous driver, and 100 samples of the costs for a safe driver. Depending on a chosen threshold value, a safe driver may be declared dangerous (if its accumulated cost is greater than the threshold), and a dangerous driver might be declared safe (if its accumulated cost is smaller than the threshold).

Figure 5 shows the confusion error rate as a function of the threshold. The error rate measures the percentage of cases (out of 100) incorrectly identified. The figure shows that a trade-off exists in setting the threshold, in order to improve accuracy. Choosing a cost threshold at 50 will result in high accuracy, in this particular case: All dangerous drivers will be identified as dangerous, and yet 99 percent of safe drivers will be correctly identified as safe.

Air-Combat Environment

(Tambe & Rosenbloom 1995) used an example of agents in a simulated air-combat environment to demonstrate the RESC plan recognition algorithm. RESC heuristically prefers a single worst-case hypothesis, since an opponent is likely to engage in the most harmful maneuver. The example used was of a air-combat maneuver, in (Tambe & Rosenbloom 1995) showed this heuristic in action in a simulated air-combat, where the turning actions of the opponent could be interpreted as either leading to it running away, or to its shooting a missile. RESC prefers the hypothesis that the opponent is shooting. However, unlike UPR, RESC will *always* prefer this hypothesis, regardless of its likelihood, and this has proven problematic (Tambe & Rosenbloom 1995). Moreover, given several worst-case hypotheses, RESC will choose arbitrarily a single hypothesis to commit to, again regardless of its likelihood. Additional heuristics were therefore devised to control RESC's worst-case strategy (Tambe & Rosenbloom 1995).

In contrast, UPR incorporates the biases of an *observing* pilot much more cleanly. Because it takes the likelihood of hypotheses into account in computing the expected cost, it can ignore sufficiently improbable (but still possible) worst-case hypotheses, in a principled manner. Moreover, UPR also allows modeling optimistic observers, who prefer best-case hypotheses.

Summary and Future Work

This paper presents a utility-based plan recognition (UPR) approach, for incorporating biases and preferences of the observer into keyhole plan recognition. This allows choosing recognition hypotheses based on their expected utility to the observer. While reasoning about such expected utilities is intractable in the general case, we present a hybrid symbolic decision-theoretic plan recognizer, whose complexity is $O(NDT)$, where N is the plan library size, D is the depth of the library and T is the number of observations. We demonstrated the efficacy of this approach with experimental results in several challenging recognition tasks. We plan to further explore the use of UPR algorithms in additional queries and cases such as *intended recognition*, where the observed agent may modify its behavior based on the knowledge that it is being observed.

Acknowledgments. We thank Jacob Goldberger for invaluable discussions. This research was supported in part by ISF grant #1211/04. Special thanks to Nadav Zilberbrand and K.Ushi.

References

Avrahami-Zilberbrand, D., and Kaminka, G. A. 2005. Fast and complete symbolic plan recognition. In *IJCAI-05*.

Avrahami-Zilberbrand, D.; Kaminka, G. A.; and Zarosim, H. 2005. Fast and complete plan recognition: Allowing for duration, interleaved execution, and lossy observations. In *Proceedings of the MOO-05 Workshop*.

Charniak, E., and Goldman, R. P. 1993. A Bayesian model of plan recognition. *AIJ* 64(1):53–79.

Duong, T. V.; Bui, H. H.; Phung, D. Q.; and Venkatesh, S. 2005. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR*.

Fine, S.; Singer, Y.; and Tishby, N. 1998. The hierarchical hidden markov model: Analysis and applications. *Machine Learning* 32(1):41–62.

Geib, C. W. 2004. Assessing the complexity of plan recognition. In *AAAI-04*.

Howard, R., and Matheson, J. 1984. Influence diagrams. In Howard, R., and Matheson, J., eds., *Readings on the Principles and Applications of Decision Analysis*. Strategic Decisions Group.

Mao, W., and Gratch, J. 2004. A utility-based approach to intention recognition. In *Proceedings of the MOO-04 Workshop*.

Noh, S., and Gmytrasiewicz, P. 2005. Flexible multi-agent decision-making under time pressure. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 35(5):697–707.

Sukthankar, G., and Sycara, K. 2005. A cost minimization approach to human behavior recognition. In *AAMAS-05*.

Suzic, R. 2005. A generic model of tactical plan recognition for threat assesment. In Dasarathy, B. V., ed., *Proceedings of SPIE Multisensor*, volume 5813, 105–116.

Tambe, M., and Rosenbloom, P. S. 1995. RESC: An approach to agent tracking in a real-time, dynamic environment. In *IJCAI-95*.