Sequential Plan Recognition

Reuth Mirsky and **Roni Stern** and **Ya'akov (Kobi) Gal** and **Meir Kalech**
Department of Information Systems Engineering
Ben-Gurion University of the Negev, Israel
{dekelr,sternron,kobig,kalech}@bgu.ac.il

Abstract

Plan recognition algorithms infer agents' plans from their observed actions. Due to imperfect knowledge about the agent's behavior and the environment, it is often the case that there are multiple hypotheses about an agent's plans that are consistent with the observations, though only one of these hypotheses is correct. This paper addresses the problem of how to disambiguate between hypotheses, by querying the acting agent about whether a candidate plan in one of the hypotheses matches its intentions. This process is performed sequentially and used to update the set of possible hypotheses during the recognition process. The paper defines the sequential plan recognition process (SPRP), which seeks to reduce the number of hypotheses using a minimal number of queries. We propose a number of policies for the SPRP which use maximum likelihood and information gain to choose which plan to query. We show this approach works well in practice on two domains from the literature, significantly reducing the number of hypotheses using fewer queries than a baseline approach.

1 Introduction

Plan recognition (PR), the task of inferring agents' plans based on their observed actions, is a fundamental problem in AI, with a broad range of applications, such as inferring transportation routines [Liao *et al.*, 2007], advising in health care [Allen *et al.*, 2006], or recognizing activities in gaming and educational software [Uzan *et al.*, 2013].

A chief problem facing PR algorithms is how to disambiguate between multiple hypotheses (explanations) that are consistent with an observed agent's activities. A straightforward solution to this problem is to ask the observed agent to reveal the correct hypothesis, but soliciting complete hierarchies is time and information consuming and prone to error. Alternatively, querying whether a given hypothesis is correct will not contribute any information about the correct hypothesis should the answer be "false". Consider for example an e-learning software for chemistry laboratory experiments [Gal *et al.*, 2015; Yaron *et al.*, 2010]. There are many possible solution strategies that students can use to solve problems, and variations within each due to exploratory activities and mistakes carried out by the student. Given a set of actions performed by the student, one hypothesis may relate a given action to the solution of the problem, while another may relate this action to a failed attempt or a mistake. The space of possible hypotheses can become very large, even for a small number of observations. To illustrate, in the chemistry domain just seven observations produced over 11,000 hypotheses on average, with some instances producing over 32,000 hypotheses.

However, in many domains it is possible to query (for a cost) the observed agent itself or a domain expert about certain aspects of the correct hypothesis [Kamar *et al.*, 2013]. For example, the student may be asked questions about her solution strategy for a chemistry problem during her interaction with the educational software. Answers for such queries allow to reduce the set of possible hypotheses without removing the correct hypothesis. (e.g., interrupting students may disrupt their learning and incur a cognitive overhead).

The first contribution of this paper is to define the *sequential plan recognition process* (SPRP), in which we iteratively query whether a given part in one of the hypotheses is correct, and update all hypotheses in which this plan appears (or does not appear, depending on the answer to the query). We represent a hypothesis as a set of plans, one for each goal that the agent is pursuing. This allows to capture settings in which agents may pursue several goals at the same time and in which their actions may include mistakes (e.g., students performing exploratory activities in the lab and trial-and-error).

A key challenge in the SPRP is how to update the hypothesis space following the results of a query. Because recognition is performed in real-time, the hypothesis set may contain incomplete plans that describe only the observations seen thus far. Thus, for example, if the result of a query on a plan p is true (i.e., the agent plans to perform p), we cannot simply discard all hypotheses that do not contain p, because they may contain plans that will evolve to perform p in the future. To address this challenge we developed criteria for determining whether possibly incomplete plans appear in the correct hypothesis. We show that SPRP using these criteria is both sound and complete.

The second contribution of this paper is how to compute a good policy for the SPRP for choosing which plan in the current set of hypotheses to query. We consider queries that maximize the information-gain and the likelihood of the resulting hypotheses given the expected query result. The third contribution of this paper is to evaluate approaches for solving the sequential plan recognition problem in two domains from the plan recognition literature that exhibit varying degrees of ambiguity. One of the domains was synthetically generated [Kabanza *et al.*, 2013], while the other logs were taken from real students' traces when interacting with the aforementioned virtual chemistry lab [Amir and Gal, 2013; 2011]. In both domains, our approach significantly decreased the number of queries compared to a baseline technique. The number of queries performed by the information-gain approach was significantly smaller than the alternative approaches.

2 Related Work

Our work relates to different approaches in the PR literature on disambiguation of the hypothesis space during run-time. Most of the approaches admit all of the hypotheses that are consistent with the observed history and rank them [Geib and Goldman, 2009; Wiseman and Shieber, 2014].

Few works exist on interacting with the observed agent as means to disambiguate the hypothesis space during plan recognition: Bisson and Kabanza [2011] who "nudge" the agent to perform an action that will disambiguate between two possible goals. Fagundes et al. [2014] make a decision to query the observed agent if the expected time to disambiguate the hypothesis space does not exceed a predefined deadline to act in response to the recognized plan. They ask the observed agent directly about its intentions and do not prune the hypothesis space. They evaluate their approach in a simulated domain. We solve an orthogonal problem in which the observed agent can be queried repeatedly, and the hypothesis space is pruned based on the query response. We consider the cost of this query and evaluate the approach in a real-world domain.

Lastly, the deployment of probes, tests, and sensors to identify the correct diagnoses or the occurrence of events was inspired by work in sequential diagnosis [Feldman *et al.*, 2010; Siddiqi and Huang, 2011], active diagnosis [Sampath *et al.*, 1998; Haar *et al.*, 2013], and sensor minimization [Cassez and Tripakis, 2008; Debouk *et al.*, 2002]. [Keren *et al.*, 2014] suggested a metric that will allow an agent to recognize plans earlier.

3 Background

Before defining the SPRP we present some background about plans and PR. There are multiple ways to define a plan and the PR problem [Nau, 2007; Ramırez and Geffner, 2010, inter alia]. We follow the definitions used by Kabanaza et al. [2013] (simplified for brevity) in which the observing agent is given a *plan library* describing the expected behaviors of the observed agent.

Definition 1 *(Plan Library) A plan library is a tuple* $L = \langle B, C, R \rangle$*, where* B *is a set of basic actions,* C *is a set of complex actions, and* R *is a set of refinement methods of the form* $c \to (\tau, O)$*, where (1)* $c \in C$*; (2)* $\tau \in (B \cup C)^*$*; (3) and* O *is a partial order over* τ *representing ordering constraints over the actions in* τ*.*

The ordering constraints of each refinement method are used to enforce the order in which the method's constituents were executed [Geib and Goldman, 2009].

A plan is a labeled tree $p = (V, E, \mathcal{L})$, where V and E are the nodes and edges of the tree, respectively, and $\mathcal{L}$ is a labeling function $\mathcal{L} : V \to B \cup C$ mapping every node in the tree to either a basic or a complex action in the plan library. Each inner node is labeled with a complex action such that its children nodes are a decomposition of its complex action into constituent actions according to one of the refinement methods. The set of all leaves of a plan p is denoted by $leaves(p)$, and a plan is said to be *complete* iff all its leaf nodes are labeled basic actions, i.e., $\forall v \in leaves(p), \mathcal{L}(v) \in B$.

An *observation sequence* is an ordered set of basic actions that represents actions carried out by the observed agent. A plan p *describes* an observation sequence O iff every observation is mapped to a leaf in the tree. More formally, there exists an injective function $f : O \to leaves(p) \cap B$ such that $f(o) = v$. The observed agent is assumed to plan by choosing a subset of complex actions as intended goals and then carrying out a separate plan for completing each of these goals.

Importantly, an agent may pursue several goals at the same time. Therefore, a hypothesis can include a set of plans, as described in the following definition:

Definition 2 *(Hypothesis) A hypothesis for an observation sequence is a set of plans such that each plan describes a mutually exclusive subset of the observation sequence and taken together the plans describe all of the observations. We then say that the hypothesis describes the observation sequence.*

To illustrate these concepts we will use a running example from an open-ended educational software package for chemistry called VirtualLabs, which also comprises part of our empirical analysis. VirtualLabs allows students to design and carry out their own experiments for investigating chemical processes [Yaron *et al.*, 2010] by simulating the conditions and effects that characterize scientific inquiry in the physical laboratory. Such software is open-ended and flexible and is generally used in classes too large for teachers to monitor all students and provide assistance when needed. Thus, there is a need to develop recognition tools to support teachers' understanding of students' activities using the software.

We use the following problem as a running example: Given four substances $A; B; C$, and D that react in a

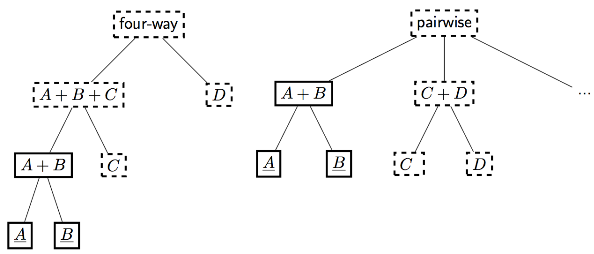

Figure 1: Two candidate hypotheses in VirtualLabs for observations A and B.

way that is unknown, design and perform virtual lab experiments to determine which of these substances react, including their stochiometric coefficients.

There are two classes of strategies used by students to solve the above problem in VirtualLabs. The most common strategy, called *pairwise*, is to mix pairs of solutions (A with B, A with C, etc.) in order to determine which solutions react with one another. In the *four-way* solution strategy, all substances are mixed in a single flask, which is sufficient to identify which solution pair were the reactants and which did not react, since the non-reactant are still observable after the reaction.

Now suppose that the student is observed to mix solutions A and B together in a single flask. Without receiving additional information, both the pairwise and four-way strategies are hypotheses that are consistent with the observations, and both include an *incomplete plan* describing the student's actions. Incomplete plans include nodes labeled with complex level actions that have not been decomposed using a refinement method. These *open frontier* nodes represent activities that the agent will carry out in future and have yet to be refined. (This is similar to the least commitment policies used by some planning approaches to delay variable bindings and commitments as much as possible [Tsuneto *et al.*, 1996; Avrahami-Zilberbrand and Kaminka, 2005; 2007].) This ambiguity is exemplified in Figure 1, showing one hypothesis for the four-way solution strategy (left) and one for the pairwise solution strategy (right). Each of these hypotheses contain a single incomplete plan. The nodes representing the observations A and B are underlined. The dashed nodes denote open frontier nodes.

We can now define the plan recognition problem.

Definition 3 *(Plan Recognition (PR)) A PR problem is defined by the tuple $\langle L, O\rangle$ where L is a plan library and O is an observation seqeuence. A PR algorithm accepts a PR problem and outputs a set of hypotheses H such that each hypothesis describes the observation sequence.*

Let h^* be the *correct* hypothesis, i.e., the set of plans the agent intends to follow (h^* is not known at recognition time). When recognition is performed in real-time, observations are collected over time, there is uncertainty about future activities, and the agent's plans may be incomplete (e.g., the agent may have not decided how to perform some of the planned complex actions). To address this challenge we require the following notion of *plan refinement*.

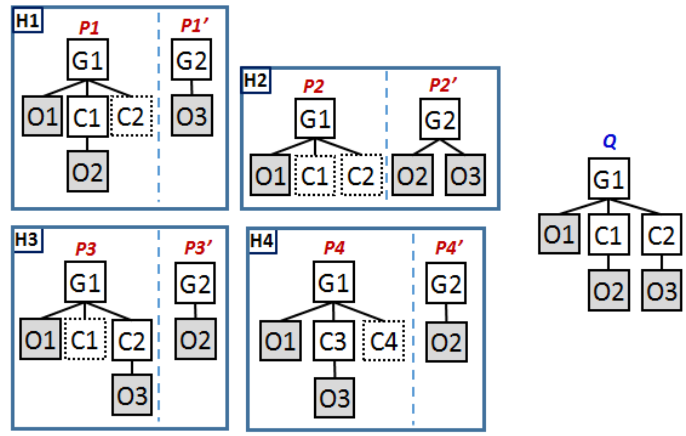

Figure 2: Four candidate hypotheses $H1$, $H2$, $H3$, and $H4$ for observations $O1$, $O2$, and $O3$, and one complete plan Q.

Definition 4 *(Refinement of a plan) A plan p is a refinement of a plan p', denoted by $p' \sim^r p$, if the plan p can be obtained by applying a (possibly empty) sequence of refinement methods from the plan library L to p'.*

The refinement criterion is asymmetric and transitive. Note that a plan can always refined from itself using an empty sequence of refinement methods. We extend the refinement criteria to hypotheses as follows. A hypothesis h is a refinement of a hypothesis h', denoted ($h' \sim^r h$), if there is a one-to-one mapping between every plan $p \in h$ and a plan $p' \in h'$ such that p is a refinement of p' ($p' \sim^r p$).

Using this definition, a PR algorithm is complete if it returns a hypothesis set H such that $h \sim^r h^* \rightarrow h \in H$, that is, H contains all possible hypotheses that can be refined to h^*.

To illustrate, the top part of Figure 2 shows part of a hypothesis set $H1, \ldots, H4$, each of these hypotheses explains the observations $O1$, $O2$, and $O3$. The figure shows the four possible hypotheses for the observation sequence. Each hypothesis Hi ($i = 1, \ldots, 4$) consists of two plans, P_i and P_i'. Nodes in gray represent the observations and nodes with dashed outline represent open-frontier actions.

The plans $P1$ and $P3$ are both refinements of $P2$ ($P2 \sim^r P1$ and $P2 \sim^r P3$), but they are not refinement of each other ($P1 \not\sim^r P3$ and $P3 \not\sim^r P1$). In addition, $H1$ is not a refinement of $H2$ ($H2 \not\sim^r H1$) because the plan $P1'$ is not a refinement of $P2'$ ($P2' \not\sim^r P1'$). Similarly, $H3$ is not a refinement of $H2$ ($H2 \not\sim^r H3$).

4 Sequential Plan Recognition

In this section we define the SPRP, beginning with the notion of a query function and a query policy.

Definition 5 *(Query Function) A query QA is a function that receives as input a plan p and outputs whether one of the plans in the correct hypothesis h^* can be re-*

Algorithm 1: Sequential Plan Recognition Process.

Input: H_0 is the initial set of hypotheses
Input: QA is a query function
Input: π is a query policy
1 $i \leftarrow 0$; $CLOSED \leftarrow \emptyset$
2 **while** $\bigcup_{h \in H_i} h \setminus CLOSED \neq \emptyset$ *or* $|H_i| = 1$ **do**
3 $\quad p \leftarrow \pi(H_i)$
4 $\quad H_{i+1} \leftarrow \text{Update}(QA(p), H_i, p)$
5 $\quad i \leftarrow i + 1$
6 $\quad$ Add p to $CLOSED$

fined from p.

$$QA(p) = \begin{cases} \text{True} & \text{if } \exists p' \in h^* \text{ s.t. } p \sim^r p' \\ \text{False} & \textit{otherwise} \end{cases} \tag{1}$$

A query policy selects which plan to query in a SPRP given the current set of hypotheses.

Definition 6 *(Query Policy) A query policy is a function $\pi : \mathcal{H} \rightarrow \mathcal{P}_{\mathcal{H}}$, where $\mathcal{H}$ is the set of all possible hypotheses and $\mathcal{P}_{\mathcal{H}}$ is the set of all plans in all possible hypotheses.*

Such a policy needs to trade off the immediate benefits of a query with the short and long term costs associated with disrupting the acting agent.

Given an initial hypothesis set H_0 (obtained by applying a PR algorithm), a query function $QA(\cdot)$, and a query policy π, the *Sequential Plan Recognition Process (SPRP)* is the iterative process shown in Algorithm 1. Starting from the initial iteration $i = 0$, in every iteration of SPRP, a candidate plan p is chosen from the set of hypotheses H_i and the result of the query is used to generate an updated hypothesis set H_{i+1} to be used in the following iteration. We maintain a $CLOSED$ list of the chosen hypotheses up to step i, and terminate when there are no more plans in the hypothesis set left to query or there is just a single hypothesis in the set H_i (line 3). The output of the algorithm is the set H_i at the last iteration.

The key step in the algorithm is how H_i should change after performing a query on a plan p (line 5). Suppose $QA(p) = \text{True}$. According to Definition 5, this means that there exists a plan $p^* \in h^*$ such that p^* is a refinement of p ($p \sim^r p^*$). If we knew p^* we could simply remove from H_i all hypotheses that do not contain a plan p' that can be refined to p^* ($p' \sim^r p^*$).

Since we do not know p^*, a natural option is to remove from H_i all the hypotheses that do not have any plan p' that can be refined from p. However, in certain situations this may lead us to discard the correct hypotheses. Consider the example of Figure 2 and assume that we query plan $P1$ which returns true (i.e., $QA(P1) = \text{True}$). If we remove all hypotheses that do not contain plans that are refinements of $P1$, then hypothesis $H3$ will be removed, since neither $P3$ nor $P3'$ are refinements of $P1$. However, it may be the case that one of the agent's intended plans is plan Q (right of Fig. 2). The query on $P1$ returned true because $P1 \sim^r Q$. However, note that $H3$ is a valid hypothesis and should not be discarded, since $P3 \sim^r Q$. Thus, we require a different pruning criteria for the hypotheses, given an outcome of query.

To handle this problem, we need to devise a new criteria for determining whether two plans can be used to refine a third plan. We will present this criteria and then show how it can be used to update the set of hypotheses for the next time step in a way that preserves the completeness of the PR process.

Definition 7 *(Matching of Plans) A pair of plans p and p' are said to match, denoted by $p' \sim^m p$ (or $p \sim^m p'$), if there exists a plan p'' that is a refinement of both plans p and p' ($p \sim^r p''$ and $p' \sim^r p''$).*

Note that the match criteria is symmetric. To illustrate this concept using the example in Figure 2, the plan $P1$ matches $P3$ ($P1 \sim^m P3$), even though they are not refinements of each other, since there is at least one plan which is a refinement of both. The complete plan Q is an example of such a plan, since $P1 \sim^r Q$ and $P3 \sim^r Q$.

Using both the *match* (Definition 7) and *refinement* (Definition 4) relations, we define the update rule (Algorithm 1, line 4) over the hypothesis set H which depends on whether the query $QA(p)$ returns True or False:

Case 1: $QA(p) = \text{True}$. For this case we define the set $\phi(H, p, \text{True})$ which includes only hypotheses in which at least one of the plans match p:

$$\phi(H, p, \text{True}) = \{h \mid h \in H \wedge \exists p' \in h \ \ p' \sim^m p\} \tag{2}$$

In our example in Figure 2, if $QA(P1) = \text{True}$ then we know that the correct hypothesis h^* will contain a complete plan that is a refinement of $P1$. In particular, Q is a possible refinement of $P1$. Thus, any hypothesis $h \in \{H1 \ldots, H4\}$ that has at least one plan p that can be refined to Q (or any other plan that is a refinement of $P1$) cannot be pruned. Therefore, the hypothesis $H2$ is not pruned, because Q is a refinement of $P2$ ($P2 \sim^r Q$). Similarly, the hypothesis $H3$ is not pruned, because Q is a refinement of $P3$ ($P3 \sim^r Q$). However, the hypothesis $H4$ is pruned since there is no plan in it that can be refined to a plan that is also a refinement of $P1$.

Case 2: $QA(p) = \text{False}$. This means that there is no plan $p^* \in h^*$ that is a refinement of p. The refinement operator is transitive, i.e., if p'' is a refinement of p' and p' is a refinement of p, then p'' is also a refinement of p. Therefore, if h^* does not contain any plan that is a refinement of p, we can safely remove from H every hypothesis that contains a plan p' such that p' is a refinement of p.

$$\phi(H, p, \text{False}) = H \setminus \{h \mid h \in H \wedge \exists p' \in h \ \ p \sim^r p'\} \tag{3}$$

In our example in Figure 2, if $QA(P2) = \text{False}$, there does not exist any plan in h^* that is a refinement of $P2$. Therefore, we can safely remove hypotheses $H1$, $H2$, and $H3$, because each of them has at least one plan that is a refinement of $P2$ (formally, $P2 \sim^r P1, P2 \sim^r P3$, and $P1 \sim^r P1$). If $QA(P1) = \text{False}$, then only $H1$ is pruned.

We can now define the update rule (line 4) for the Sequential Plan Recognition Process as follows:

$$\text{Update}(QA(p), H_i, p) = \begin{cases} \phi(H_i, p, \text{True}) & \text{QA(p)=True.} \\ \phi(H_i, p, \text{False}) & \text{otherwise} \end{cases}$$

We assume that the PR algorithm is complete and provides a set of probability-ranked hypotheses, as is common in the state-of-the art. We can now state that SPRP described in Algorithm 1 is both sound and complete:

Proposition 1 *The SPRP will necessarily terminate in a finite number of iterations k with a hypothesis set $H_k \subseteq H_0$ such that the following holds:*

Completeness *SPRP does not remove any hypothesis that can be refined to the correct hypothesis h^*. Formally, $\forall h \in H_0, \quad h \sim^r h^* \rightarrow h \in H_k$.*

Soundness *Every hypothesis SPRP keeps can be refined to the correct hypothesis h^*. Formally, $\forall h \in H_k, \quad h \sim^r h^*$.*

Termination First, we must show that after a finite number of iterations, the SPRP will terminate. This is immediate, since at each iteration we ask about a plan from the remaining set of plans. This means that at the worst case, if no hypothesis is removed, the process will terminate after $| T |$ iterations, where T is the set of all plans in all hypotheses.

Completeness We prove completeness by showing that every h that was removed from H_0, could not be refined to h^*. This reasoning follows from the update rule in each case of examining some plan p: If $QA(p) = \text{True}$, then

$$\begin{aligned} QA(p) = \text{True} &\Rightarrow \exists p^* \in h^* \;\; p \sim^r p^* \\ &\Rightarrow \forall h \in H \;\; h \sim^r h^* \rightarrow \exists p' \in h \;\; p' \sim^r p^* \end{aligned}$$

We can conclude that if $\forall p' \in h$ do not match the query plan p, we can safely remove the hypothesis h because h^* cannot be refined from h. If $QA(p) = \text{False}$, then the following holds:

$$\begin{aligned} QA(p) = \text{False} &\Rightarrow \forall p^* \in h^* \;\; \neg(p \sim^r p^*) \\ &\Rightarrow \forall h \in H \;\; \exists p' \in h \;\; p' \sim^r p \rightarrow \neg(p' \sim^r p^*) \\ &\Rightarrow \forall h \in H \;\; \exists p' \in h \;\; p' \sim^r p \rightarrow \neg(h \sim^r h^*) \end{aligned}$$

Thus, we can conclude that if the query plan p can be refined from $p' \in h$, we can safely remove the hypothesis h because h^* cannot be refined from h.

Soundness Let H_k be the set of all hypotheses after k iterations and h^* is the correct hypothesis. If there is still a hypothesis $h \in H_k$ such that $\neg(h \sim^r h^*)$, then $\exists p \in h \;\; \forall p^* \in h^* \neg(p \sim^r p^*)$. Thus, we can still query about p and k is not the final iteration of the algorithm. Hence, at the final iteration of the algorithm we have that $\forall h \;\; h \sim^r h^*$.

5 Probing Techniques

We propose several heuristic methods for generating a PR policy that aim to minimize the number of queries required to achieve the minimal set of hypotheses that are consistent with the observation. These methods rely on the standard assumption that each hypothesis h is associated with a lity $P(h)$ that is assigned by the PR algorithm (such as PHATT, DOPLAR and ELEXIR [Geib and Goldman, 2009; Kabanza *et al.*, 2013; Geib, 2009]).

Most Probable Hypothesis (MPH). Choose a plan from the hypothesis h that is associated with the highest probability and was not yet queried about, i.e., choose a plan t such that $t \in h = \text{argmax}_{h \in H_i} P(h)$.

Most Probable Plan (MPP). Choose the plan that is associated with the highest cumulative probability across all hypotheses: $\text{argmax}_{t \in T} P(t)$, where T is the union set of all plans in all of the hypotheses H, and $P(t)$ denotes the cumulative probability assigned to all hypotheses that contain the plan t, computed as follows:

$$P(t) = \sum_{h \in H | \exists p \in h, t \sim^r p} P(h) \tag{4}$$

Minimal Entropy (ME). Choose the plan with the maximal information gain (or minimal entropy) given the resulting hypothesis set. The information gain directly depends on Equations 2 and 3 for updating the hypothesis space following the results of the query $QA(p)$.

$$\begin{aligned} \min_{t \in T} & P(t) \cdot Ent(\phi(H_i, t, \text{True})) + \\ & (1 - P(t)) \cdot Ent(\phi(H_i, t, \text{False})) \end{aligned}$$

where $Ent(\cdot)$ is the standard entropy computation over the resulting hypothesis space [Shannon, 2001].

6 Empirical Evaluation

We evaluated the probing approaches described in the previous sections on two separate domains from the plan recognition literature. The first is the simulated domain used by Kabanza et al. [2013]. We used their same configuration which includes 100 instances with a fixed number of actions, five identified goals, and a branching factor of 3 for rules in the grammar. The second domain involves students' interactions with the VirtualLabs system when solving two different types of problems: the problem described in Section 2, and a problem which required students to determine the concentration level of an unknown acid solution by performing a chemical titration process. We sampled 35 logs of students' interactions in VirtualLabs to solve the above problems. In each of the logs, we used domain experts to tag the correct hypothesis. We used a plan-library representation which extended basic and complex actions to include parameters, and used the refinement methods from Amir and Gal [2013] which considered constraints over the parameter values.

We used the Most Probable Plan (MPP), the Most Probable Hypothesis (MPH) and the Minimal Entropy (Entropy) approaches, as well as a baseline approach that picked a plan to query at random. For both domains, we kept the PR algorithm constant as the PHATT algorithm [Geib and Goldman, 2009] and only varied the type of query mechanism used for the SPR.

We first show the number of hypotheses that were outputted by PHATT for the various approaches, without

Obs.	3	4	5	6	7
Hyp. (VL)	19	83	363	2,011	11,759
Hyp. (simulated)	12	25	28	32	25

Table 1: Number of hypotheses per observation.

probing interventions. As can be seen in Table 1, the number of hypotheses in the simulated domain grows linearly in the number of observations, but for the real-world domain, the number of hypotheses grows exponentially, reaching over 10,000 hypotheses after just 7 actions. Figure 3 shows the average percentage of hypotheses remaining from the initial hypothesis set (H_0) as a function of the number of queries performed. Before the first query, all algorithms start with 100% of the hypotheses in H_0, and this number decreases as more queries are performed. For both domains we used the plan recognition output after 7 observations.

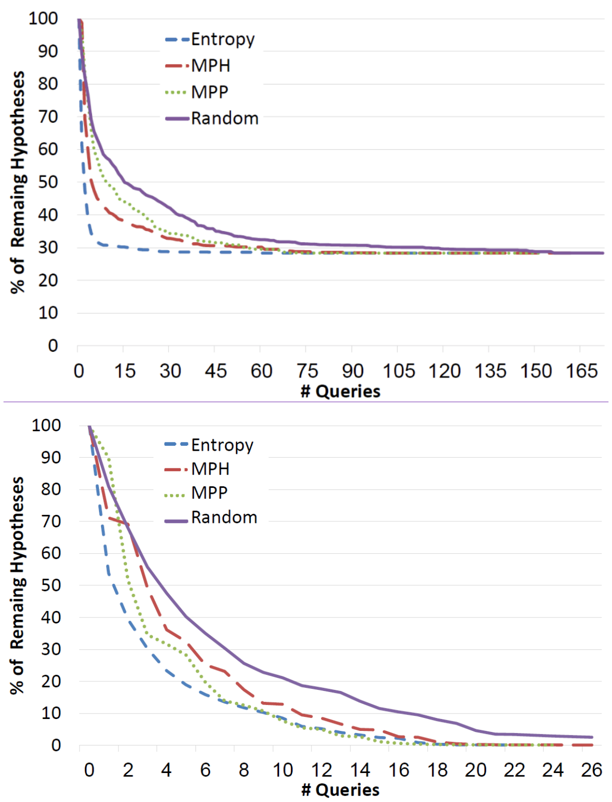

Figure 3: Decrease in the hypothesis set size after each query in the simulated domain (top) and VirtualLabs (bottom).

As seen in Figure 3, both in the simulated domain and in the VL domain, the Entropy probe performed better than all other probes. In general, all non-trivial probing techniques were able to reduce the number of hypotheses significantly compared to random, and Entropy outperformed all algorithms. As seen in the figures, although the PR process created more hypotheses for the VL domain, the convergence of SPRP is usually to a single hypothesis, while in the smaller simulated domain, all algorithms converge to a minimal hypothesis set of about 30% of the number of hypotheses in H_0. We attribute this to inherent ambiguity in this domain that cannot be resolved by making further queries.

Observations	3	4	5	6	7
Entropy-Sim	*7.3	**10.4	**15.6	**23.5	**18.4
MPP-Sim	*7.3	*10.8	*16.3	*25.2	*21.0
MPH-Sim	*7.6	*10.8	*16.3	*25.4	*19.9
Random-Sim	7.9	11.9	18.4	29.0	28.7
Entropy-VL	**7.6	*10.7	**13.4	*17.2	*18.3
MPP-VL	*8.2	*11.2	*14.7	*17.7	*18.9
MPH-VL	8.8	*12.2	*16.2	*19.8	21.6
Random-VL	9.5	14.9	24.0	36.3	27.8

* - significantly less queries compared to random,
** - significantly less queries compared to all other strategies ($p \leq 0.05$).

Table 2: Average Number of Queries until Convergence.

In general, the advantage of Entropy over all other approaches for the first five queries was statistically significant ($p \lesssim 0.01$). This is especially important since queries are costly and the the number of queries that can practically be asked is small. Thus an approach able to limit the hypotheses more with fewer queries is preferred. Lastly, Table 2 shows the average number of queries needed until reaching the minimal set of hypotheses, for each probing strategy. Notice that the number of hypotheses increase with each new observation. Although counter-intuitive, this is due to the fact that for each hypothesis, a new observation can initiate a new plan or complement an existing plan (or both), so the size of the hypothesis space will be at least the size of the original one. This table shows that the Entropy probe made significantly fewer queries than the other approaches.

7 Conclusion

This paper defined and studied SPRP, in which it is possible to query whether a chosen plan is part of the correct hypothesis, and subsequently remove all incorrect plans from the hypothesis space. The goal is to minimize the number of queries to converge to the minimal hypothesis set that is consistent with the observations. We presented a number of approaches for choosing a plan to query – the plan that maximizes the expected information gain, as well as the plan that is ranked highest in terms of likelihood by the PR algorithm. We evaluated these approaches on two domains from the literature, showing that both were able to converge to the correct hypothesis using significantly less queries than a random baseline, with the maximal information gain technique exhibiting a clear advantage over all approaches.

We are working on extending the heuristic approach described in the paper to using MDPs to allow for the probing policy to reason about future steps. To this end we are working on a compact representation of a state space to represent the set of possible hypotheses. We also intend to use our approach to augment existing educational software to intelligently query students about their solution strategy in a way that minimizes the disruption.

Acknowledgments

This research was funded in part by ISF grant numbers 363/12 and 1276/12, and by EU FP7 FET project no. 60085. R.M. is a recipient of the Pratt fellowship at the Ben-Gurion University of the Negev.

References

[Allen *et al.*, 2006] J. Allen, G. Ferguson, N. Blaylock, D. Byron, N. Chambers, M. Dzikovska, L. Galescu, and M. Swift. Chester: Towards a personal medication advisor. *Journal of Biomedical Informatics*, 39(5):500 – 513, 2006.

[Amir and Gal, 2011] O. Amir and Y. Gal. Plan recognition in virtual laboratories. In *IJCAI*, 2011.

[Amir and Gal, 2013] O. Amir and Y. Gal. Plan recognition and visualization in exploratory learning environments. *ACM Transactions on Interactive Intelligent Systems*, 3(3):16:1–23, 2013.

[Avrahami-Zilberbrand and Kaminka, 2005] D. Avrahami-Zilberbrand and G.A. Kaminka. Fast and complete symbolic plan recognition. In *IJCAI*, volume 14, 2005.

[Avrahami-Zilberbrand and Kaminka, 2007] D. Avrahami-Zilberbrand and G.A. Kaminka. Incorporating observer biases in keyhole plan recognition (efficiently!). pages 944–949, 2007.

[Bisson *et al.*, 2011] F. Bisson, F. Kabanza, A. R. Benaskeur, and H. Irandoust. Provoking opponents to facilitate the recognition of their intentions. In *AAAI*, 2011.

[Cassez and Tripakis, 2008] F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundam. Inform.*, 88(4):497–540, 2008.

[Debouk *et al.*, 2002] R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection*. *Discrete Event Dynamic Systems*, 12(4):417–445, 2002.

[Fagundes *et al.*, 2014] M. S. Fagundes, F. Meneguzzi, R. H. Bordini, and R. Vieira. Dealing with ambiguity in plan recognition under time constraints. In *AAMAS*, pages 389–396, 2014.

[Feldman *et al.*, 2010] A. Feldman, G. Provan, and A. van Gemund. A model-based active testing approach to sequential diagnosis. *Journal of Artificial Intelligence Research (JAIR)*, 39:301, 2010.

[Gal *et al.*, 2015] Y. Gal, O. Uzan, R. Belford, M. Karabinos, and D. Yaron. Making sense of students actions in an open-ended virtual laboratory environment. *Journal of Chemical Education*, 92(4):610–616, 2015.

[Geib and Goldman, 2009] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[Geib, 2009] C. W. Geib. Delaying commitment in plan recognition using combinatory categorial grammars. In *IJCAI*, pages 1702–1707, 2009.

[Haar *et al.*, 2013] S. Haar, S. Haddad, T. Melliti, and S. Schwoon. Optimal constructions for active diagnosis. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, pages 527–539, 2013.

[Kabanza *et al.*, 2013] F. Kabanza, J. Filion, A. R. Benaskeur, and H. Irandoust. Controlling the hypothesis space in probabilistic plan recognition. In *IJCAI*, pages 2306–2312, 2013.

[Kamar *et al.*, 2013] E. Kamar, Y. Kobi Gal, and B. J. Grosz. Modeling information exchange opportunities for effective human–computer teamwork. *Artificial Intelligence*, 195:528–550, 2013.

[Keren *et al.*, 2014] S. Keren, A. Gal, and E. Karpas. Goal recognition design. In *ICAPS Conference Proceedings*, 2014.

[Liao *et al.*, 2007] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Journal of Artificial Intelligence Research (JAIR)*, 171:311–331, 2007.

[Nau, 2007] D. S. Nau. Current Trends in Automated Planning. pages 1–16, December 2007.

[Ramırez and Geffner, 2010] M. Ramırez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI*, 2010.

[Sampath *et al.*, 1998] M. Sampath, S. Lafortune, and D. Teneketzis. Active diagnosis of discrete-event systems. *Automatic Control, IEEE Transactions on*, 43(7):908–929, 1998.

[Shannon, 2001] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[Siddiqi and Huang, 2011] S. A. Siddiqi and J. Huang. Sequential diagnosis by abstraction. *Journal of Artificial Intelligence Research (JAIR)*, pages 329–365, 2011.

[Tsuneto *et al.*, 1996] R. Tsuneto, K. Erol, J. A. Hendler, and D. S. Nau. Commitment Strategies in Hierarchical Task Network Planning. *AAAI/IAAI, Vol. 1*, pages 536–542, 1996.

[Uzan *et al.*, 2013] O. Uzan, R. Dekel, and Y. Gal. Plan recognition for exploratory domains using interleaved temporal search. In *Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED)*, 2013.

[Wiseman and Shieber, 2014] S. Wiseman and S. Shieber. Discriminatively reranking abductive proofs for plan recognition. In *ICAPS*, 2014.

[Yaron *et al.*, 2010] D. Yaron, M. Karabinos, D. Lange, J.G. Greeno, and G. Leinhardt. The ChemCollective–Virtual Labs for Introductory Chemistry Courses. *Science*, 328(5978):584, 2010.