Sequential Plan Recognition

Reuth Mirsky, Ya'akov (Kobi) Gal, Roni Stern and Meir Kalech
Ben Gurion University

ABSTRACT

Plan recognition algorithms need to maintain all candidate explanations (hypotheses) which are consistent with the observations, even though there is only a single hypothesis that is the correct one. Unfortunately, the number of possible hypotheses can be exponentially large in practice. This paper addresses the problem of how to disambiguate between many possible hypotheses that are all consistent with the actions of the observed agent. One way to reduce the number of hypotheses is to consult a domain expert or the acting agent directly about its intentions. This process can be performed sequentially, updating the set of explanations during the recognition process. The paper specifically studies how to minimize the number of queries made that are required to find the correct hypothesis. It provides a formal mapping of this problem to the probing process in model-based diagnosis and adapt a number of probing techniques for choosing which plan to query, such as maximal information gain and maximum likelihood. These approaches were evaluated on two domains from the literature using a well known plan recognition algorithm. The results showed, on all domains, that the information gain approach was able to find the correct plan using significantly fewer queries than the maximum likelihood approach as well as a baseline approach choosing random plans. Our technique can inform the design of future plan recognition systems that interleave the recognition process with intelligent interventions of their users.

1. INTRODUCTION

Plan recognition, 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 [16], advising in health care [1], or recognizing activities in gaming and educational software [13, 21].

Many real world domains are ambiguous, in the sense that there are many possible hypotheses (explanations) that are consistent with an observed agent's activities. Consider for example an e-learning software for chemistry education. Students' interactions in the lab consist of building models of chemical reactions, running the models, and analyzing the results. There can be 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. In general, the size of the hypothesis space can be very large, even with a small number of observations. To illustrate, six observations produced (on average) over 4,000 hypotheses in the aforementioned domain in our study, while seven observations produced over 11,000 hypotheses.

We focus on real-world domains in which agents may pursue several goals at the same time. Thus a hypothesis includes a set of plans, one for each goal that the agent is pursuing.

In many domains it is possible to query the observed agent itself or a domain expert about certain aspects of the correct hypothesis. The query can be performed in real time (for example, the student may be asked about her solution strategy to a problem during her interaction with educational software), or offline after all observations were collected (for example, a system administrator observing suspicious behavior that can ask a cyber security expert for her opinion of these actions). Answers for such queries allow to reduce the set of possible hypotheses in a way that does not impede the completeness of the recognition process.

A challenge to this approach is that queries are associated with a monetary or cognitive cost (e.g., interrupting students may disrupt their learning). Our first contribution is to define the *sequential plan recognition* problem: how to minimize the set of queries required to reach the correct hypothesis. An obvious 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".

Our approach is to query whether a given plan 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).

Our second contribution is to present a mapping this problem and model-based sequential diagnosis, which probes system components to enable the correct diagnosis of an observed system failure [6, 5]. This mapping allows existing sequential diagnosis algorithms [15, 9, 19] to be adapted to solve sequential plan recognition problems.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), John Thangarajah, Karl Tuyls, Stacy Marsella, Catholijn Jonker (eds.), May 9–13, 2016, Singapore.*

The third contribution of this paper is to evaluate approaches for solving the sequential plan recognition problems in two domains from the literature which exhibit varying degrees of ambiguity. One of the domains was synthetically generated [14], while the other logs were taken from real students' traces when interacting with the aforementioned virtual chemistry lab [23]. We considered candidate plans to query that maximize the information gain as well as the likelihood of the resulting hypotheses given the expected query result. In both the domains, these approaches significantly decreased the number of queries compared to several baseline techniques. In addition, once the number of hypotheses is large enough, the number of queries performed by the information-gain approach was significantly smaller than the other approaches. The potential impact of this work is to show how existing recognition systems can be extended to disambiguate the hypothesis space by intelligently querying their potential users in a way that minimizes the disruption and overhead.

2. RELATED WORK

Our work relates to different approaches in the plan recognition literature for 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. Notable examples include Avrahami-Zilberband and Kaminka [2], who rank hypotheses based on the expected utility to the observer agent and probabilistic information in the grammar. The PHATT algorithm [10] which has been widely used in many applications uses and-or trees to represent hypotheses and maintains a probability distribution over this hypothesis space. Wiseman and Shieber [22] propose an abduction technique that discriminatively scores hypotheses based on features of the plan trees.

Other works control the hypothesis space by avoiding to generate certain hypotheses or by pruning the hypothesis space itself during run-time: Kabanza et al. [14] generates a hypothesis that is estimated to make the largest contribution for predicting the agent's goals. Some approaches use probabilistic constraints over the plan duration [7] as well as resource dependencies in the plan library to eliminate hypotheses [20].

Few works exist on interacting with the observed agent as means to disambiguate the hypothesis space during plan recognition: Bisson and Kabanza [4] who "nudge" the agent to perform an action that will disambiguate between two possible goals. Fagundes et al. [8] 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. Their approach is evaluated 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.

3. BACKGROUND AND PROBLEM DEFINITION

Our representation is based on Hierarchical Task Network (HTN) [12], which includes (1) a set of *basic actions* which

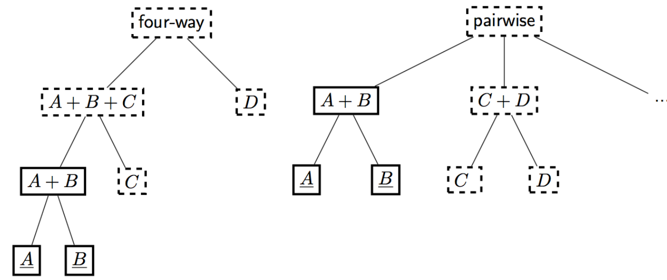

Figure 1: Two candidate hypotheses for observations A and B

which define primitive tasks carried out by the observed agent, (2) a set of *complex actions*, which describe higher-level tasks, and (3) a set of refinement methods for each complex action into constituent tasks which are themselves basic or complex actions. Planning proceeds by decomposing higher-level tasks into subtasks in a recursive manner until the planner reaches primitive tasks that are performed by the agent.

Definition 1. a plan is a tree such that each parent node is labeled with a complex action, its children nodes are a decomposition of the complex action into constituent actions according to the refinement methods for the complex action, and the leaves of the tree are labeled with basic actions.

An *observation sequence* is an ordered set of basic actions that represent actions carried out by the observed agent. A plan *describes* an observation sequence if each observation is mapped to a leaf node in the plan. We assume that the observed agent plans by choosing a subset of complex actions as intended goals and then carries out a separate plan for completing each of these goals. Importantly, we allow agents to pursue several (possibly interleaving) plans at the same time. Consequently a hypothesis includes a set of plan trees, as defined below:

Definition 2. A hypothesis for an observation sequence is a set of plan trees such that each plan tree describes a mutually exclusive subset of the observation sequence and taken together the plan trees describe all of the observations.

The input to the plan recognition algorithm PR is an observation sequence and an HTN representation. The output of the algorithm is a set of hypotheses such that each hypothesis describes the observation sequence. One of the hypotheses, denoted h^* is the *correct* hypothesis, and is unkown at recognition time. We assume that $h^* \in H$, that is the set of hypotheses outputted by the agorithm includes the correct hypothesis. A query function QA returns true if a given plan p matches the correct hypothesis $h^* \in H$. We will expand on the definition of the "matches" criteria in the next section.

Definition 3. A query/update process receives as input a set of hypothesis H, a candidate plan p, and a query function QA. If $QA(p)$ is true, then the updated hypothesis space will equal $\{h \mid h \in H, t \subseteq h, t \subseteq h^*\}$. Otherwise, the updated hypothesis space will equal $\{h \mid h \in H, t \not\subseteq h, t \not\subseteq h^*\}$.

We can now define an iterative process by which the query/update process is used to narrow the set of hypotheses for a selected

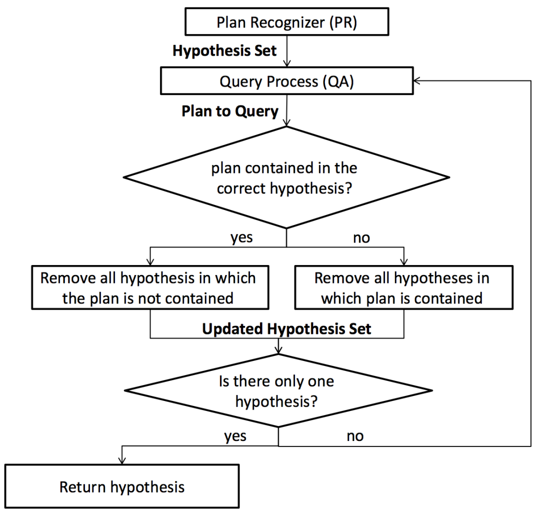

Figure 2: Flow chart of sequential plan recognition process

plan. Let PR be a plan recognition algorithm, H be the set of hypotheses outputted by PR for a given observation sequence, and let $h^* \in H$ be the correct hypothesis. Let $T = \{t \mid t \in h, h \in H\}$ be the set of plans in H.

Definition 4. A sequential plan recognition process is a pair $\langle PR, QA \rangle$, such that PR is a plan recognition algorithm and QA is a query function. At each iteration, one of the plans $p \in T$ is selected and the set of hypotheses is updated with the query process of Definition 3 with H, p and QA. This process together with the set of hypothesis H and the query function QA. The plan selection and query/update process is repeated until the set of hypotheses is reduced to a single one, as shown in Figure 2.

PROPOSITION 1. *Given that the correct hypothesis h^* exists in H, then the process described above will necessarily terminate in a finite number of iterations.*

The sequential plan recognition problem is defined as choosing the sequence of plans that minimize the number of queries in the above process to reduce the set of hypothesis to a single one.

3.1 Offline and Online SPR

In the offline case, all of the observations

To illustrate these concepts we will use 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 [23] 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 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.

Now suppose that the student is observed to mix solutions A and B together in a single flask. Without receiving additional information, both of these solution strategies are hypotheses that are consistent with the observations. 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). The nodes representing the observations A and B are underlined (action parameters are not shown for expository convenience). The dashed nodes in both hypotheses represent special type of nodes called *open frontier nodes*, which are basic or complex actions that serve as place holders for future activities that are expected to appear in order to complete the plan.

4. SEQUENTIAL DIAGNOSIS AND SEQUENTIAL PLAN RECOGNITION

In this section we present a mapping from sequential model-based diagnosis problem to sequential plan recognition. The diagnosis problem requires to identify which system components are faulty and caused a system's abnormal behavior. A diagnosis is a hypothesis about the health/fault of some or all of the system's components that, if correct, explains the observed system behavior. Automated diagnosis algorithms accept as input an observation of the system's abnormal behavior and outputs one or more diagnoses. While there is only one true diagnosis there can be multiple hypotheses that can explain a system's observed behavior, as the true internal state of the system is unknown.

Approaches to automated diagnosis abound in the AI literature, and include a range of approaches [3] such as case-based, rule-based, data-driven, and model-based. In this paper we focus on the latest approach. Model-based diagnosis (MBD) is a common approach to diagnosis in which the input to the diagnoser includes a formal model that describes the system behavior, and that model is used to infer diagnoses. This formal model, referred to as the *system description*, is a set of propositional Horn clauses, one per component, of the following form: $h(C_i) \rightarrow \varphi_{C_i}$, where C_i as a component of the system, $h(C_i)$ is a predicate stating that C_i is healthy and φ_{C_i} describes the normal behavior of C_i. More detailed system descriptions may also include information about how C_i behaves when abnormal.

Definition 5. A model-based diagnosis (MBD) problem is a tuple $\langle SD, COMPS, OBS \rangle$, where SD is the system description, $COMPS$ is the set of components that may be abnormal, and OBS is the observed behavior of the system. A solution to the MBD problem is a *diagnosis*, which is a set of components ω such that

$$SD \cup OBS \cup (\bigwedge_{C_i \in COMPS \setminus \omega} h(C_i)) \cup (\bigwedge_{C_i \in \omega} \neg h(C_i))$$

is consistent.

An MBD problem arises when $SD \wedge OBS$ is not consistent with the assumption that all components are healthy ($\bigwedge_{C_i \in COMPS} h(C_i)$). There may more than a single diagnosis for an MBD problem. Some MBD algorithms also provide an estimate of the likelihood of each diagnosis $\omega_i \in \Omega$ to be true, denoted by $p(\omega_i)$ [6].

4.1 From Model-Based Diagnosis to Grammar-Based Plan Recognition

The relation between plan recognition and diagnosis has long been recognized [17, inter alia]. The diagnosis task can be viewed as a special case of a plan recognition problem, where the plan to recognize is the set of components whose observed actions match the abnormal behavior that is exhibited by the system. We take this relation a step further, showing the relationship between *sequential diagnosis* and plan recognition with queries. Sequential diagnosis is the process of taking "a sequence of measurements of [components in] the system until the faults causing the abnormalities are identified" [19]. These measurements, referred to as *probes*, validate whether given components in a diagnosis are faulty. The most basic probe measures a single internal component in the system and reveals the value it outputs to its adjacent internal components.

Many works in the automated diagnosis literature deal with automated probe selection in sequential diagnosis are iterative [6, 19, 9, 24]. In every iteration several possible diagnoses that are consistent with the system behavior are generated. If a single diagnosis is found, the algorithm halts. Otherwise, a probe is decided upon and executed, providing a new observation for the diagnosis algorithm, with the intent of narrowing the set of possible diagnosis. The diagnosis algorithm and the probe selection algorithm can be decoupled.

Definition 6. A sequential diagnosis process is a tuple $\langle DA, PA \rangle$, where DA is a MBD problem and PA is a probe selection algorithm. The DA returns a set of diagnoses, each consists an assumption about which components are faulty. The PA accepts as input these set of diagnoses and outputs which component to probe. The observation obtained by executing the selected probe is then given to the DA, which returns a refined set of diagnoses, and so on until a refined enough set of diagnoses is returned.

We can now describe a mapping between the sequential diagnoistic process and the sequential plan recognition process described in Section 3.

- The grammar is mapped to the system description SD, which defines the possible behaviors of the system (when behaving normally).
- The sequence of observed events is mapped to the observed behavior of the system OBS.
- The set of hypotheses outputted by the plan recognition algorithm is mapped to the set of diagnoses Ω and each plan in the hypotheses corresponds to an assumption about a component being abnormal.
- The query in grammar-based plan recognition is mapped naturally to a probe, where diagnoses not consistent with the output of the probe are pruned from Ω.

In both domains the purpose is to limit the number of queries (or probes) until a single hypothesis (or diagnosis) is found, since queries (or probes) are costly. As shown in the next section, this mapping of probes to queries enables using algorithms for probe selection from the sequential model-based diagnosis literature to intelligently select queries for grammar-based plan recognition.

4.2 Probing Techniques

We adopted the following probing techniques from the MBD literature as processes for choosing which plan to query in sequential plan recognition [9, 19]. We do assume that the PR algorithm maintains a set of possible hypotheses H, and that each hypothesis h is associated with a likelihood $p(h)$, which is commonly available in state-of-the-art recognition algorithms such as PHATT, DOPLAR and ELEXIR [10, 14, 11]. Let $P(t)$ denote the cumulative probability assigned to all hypotheses that contain the plan t, computed as follows:

$$P(t) = \sum_{h \in H | t \in h} P(h)$$

- The *Most Probable Explanation (MPE)* approach sequentially chooses a plan to query from the hypothesis h that is associated with the highest probability and was not queried about:

$$h = \operatorname*{argmax}_{h' \in H} P(h')$$

- The *Most Probable Tree (MPT)* chooses the plan that is associated with the highest cumulative probability across all explanations:

$$P(t) = \sum_{h \in H | \exists p \in h : t \ {}^r p} P(h) \qquad (1)$$

- The *Minimal Entropy (ME)* approach chooses the plan with maximal information gain (or minimal entropy) given the resulting hypothesis set:

$$\min_{t \in T} P(t) \cdot E(\phi(H, t, \text{True})) + (1 - P(t)) \cdot E(\phi(H, t, \text{False})) \qquad (2)$$

where $Ent(\phi(H, t, \text{True}))$ is the entropy associated with all hypotheses which include plan t in the hypothesis space:

$$Ent(\phi(H, t, \text{True})) = \sum_{h \in H | t \ {}^m p | p \in h} -P(h) \cdot log_2 P(h) \qquad (3)$$

We can use a similar approach to compute $Ent(\phi(H, t, \text{False}))$, the entropy associated with not including plan t in the hypothesis space, by summing over all hypotheses $h \in H \mid \forall p \in h \rightarrow t \ /^r p$.

When updating the set of possible diagnoses in MBD following the query, it is sufficient to require that each diagnosis does not consist of invalid components. However, in sequential plan recognition, there is uncertainty about future observation, and we need a looser criteria to check whether a queried plan is a part of the correct hypothesis, which

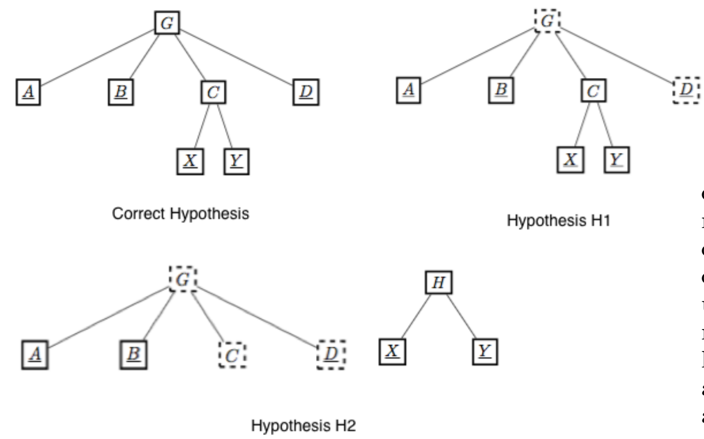

Figure 3: Two candidate hypotheses and the correct hypothesis

assumes full knowledge of the observation sequence. To illustrate, consider the example in Figure 3, which contains a correct hypothesis and two candidate hypotheses $H1$ and $H2$ that are consistent with the observations A, B, X, Y (underlined in the figure). Dashed nodes denote open frontier actions which represent future activities. Suppose that the plan chosen to query was the one rooted at G in $H1$. All of the instantiated nodes of this plan agree with those in the correct hypothesis. This plan should be contained in the correct hypothesis, despite not having seen the final observation D. In addition, this plan should not be contained in hypothesis $H2$. This is because the equivalent plan G in $H2$ did not instantiate the action represented by the C node. Therefore, hypothesis $H2$ will be eliminated from the hypothesis set according to the sequential process shown in Figure 2.

In contrast, suppose we chose to query plan rooted at G in $H2$. All of the instantiated nodes of this plan agree with those in the correct hypothesis (note that C and D are not instantiated). The plan G in $H2$ should be contained in the correct hypothesis (as we are expecting to see X, Y and D in future to realize the plan G in $H2$). This plan is also contained in hypothesis $H1$, because it agrees with G in $H1$ on all of the instantiated nodes of G in $H2$. As a result, no information gain will be incurred from choosing to query G in $H2$ because it does not lead to any change in the hypothesis space.

The following definition captures these issues: We say that a plan t_1 is *contained* in a plan t_2. denoted $t_1 \subseteq t_2$ if there is an injective mapping between every (possibly open frontier) node n_2 in t_2 to an equivalent node n_1 in t_1 such that n_1 and n_2 are instantiated with the same action and their parameter values do not conflict. We use this criteria both when choosing a plan to query and when updating the hypothesis space in the sequential plan recognition process that is shown in Figure 2.

5. EMPIRICAL EVALUATION

We evaluated the probing approaches described in the previous sections on two separate domains from the plan recognition. The first is the simulated domain used by Kabanza et al. [14]. This domain includes 100 instances with a fixed number of actions, 10 identified goals, and a branching factor of 3 for rules in the grammar. The second domain consisted of real students' traces solving the aforementioned problem using the VirtualLabs software described earlier. This domain includes 33 logs of various sizes as well as the correct hypothesis for each of these logs. We used the Most Probable Tree (MPT), the Most Probable Explanation (MPE) and the Minimal Entropy (Entropy) approaches, as well as a baseline approach that picked a tree to query at random. For both domains, we used the PHATT algorithm [10].

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

We first show the number of hypotheses that were outputted by PHATT for the various approaches, without 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, which is significantly more ambiguous, the number of hypotheses grows exponentially, reaching over 10,000 hypotheses after just 7 actions.

Figure 4 shows the average number of queries until converging to the single correct hypothesis in both of the domains, as a function of the number of hypotheses. As shown by the figure, in both domains, the random baseline generated significantly more queries than did the other approaches. In particular, the number of queries needed for convergence grew exponentially for the random approach in the VirtualLabs domain. As seen in Figure 4, at first there is not enough ambiguity between the different explanations to make the insights from the queries useful, thus no method is significantly better than the others. As we increase the number of explanations, we will see a difference between the more informed query approaches. As shown by the figure, although there is a monotonous increase in the number of queries required by all approaches, the growth rate of the Entropy query approach is the smallest. In total, the number of queries that were generated by the Entropy approach were significantly less than the number of queries generated by the MPT and MPE approaches (two-sided t-test $p < 0.05$).

The above pattern is also apparent when analyzing the number of queries as a function of the amount of observations which is shown in Figure 5 (we used the same range in both domains). The increase in the number of queries posed by the Entropy, MPE and MPT approaches is small and constant, while the increase in the random approach is much more pronounced. In particular, for 7 actions, this approach required over 30 queries to reach the correct hypothesis. These results suggest that the probing based approaches will be able to scale up to settings with high ambiguity (large number of hypotheses) without incurring a substantial increase in the number of queries needed to converge to the correct hypothesis.

6. DISCUSSION AND CONCLUSION

Many plan recognition approaches in the literature are based on a plan library or grammar which describe how complex level actions can be decomposed to constituent actions.

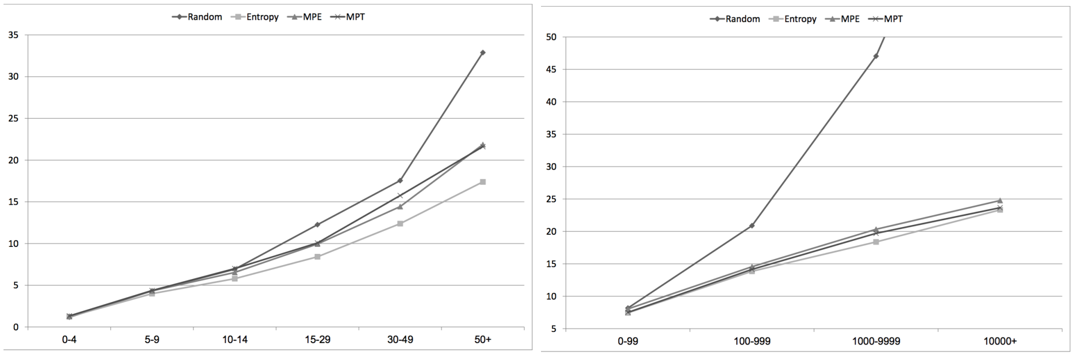

Figure 4: Queries until convergence given explanations for simulated domain (left) and VirtualLabs (right).

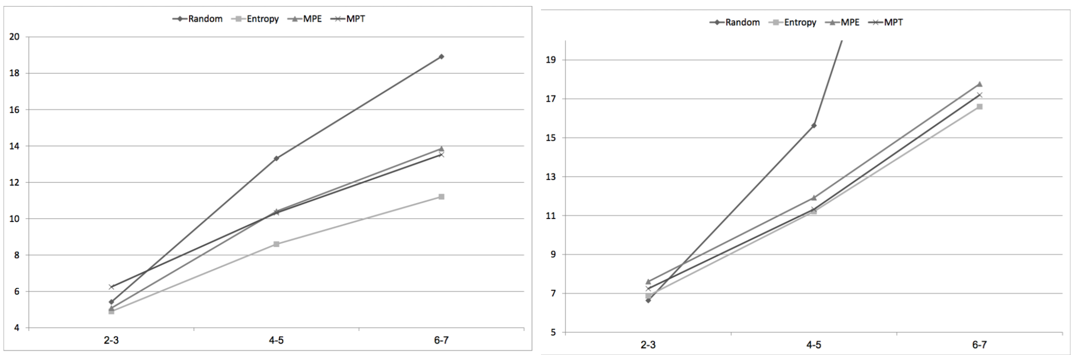

Figure 5: Queries until convergence given observations for simulated domain (left) and VirtualLabs (right).

Other approaches infer these rules using machine learning or other approaches. Our approch

Our approach does not explicitly require a plan-tree (sometimes referred to as a "grammar") that describes how abstract actions are decopmosed to their respective constituents. We expand on this point in the Discussion section.

This paper defined and studied the sequential plan recognition problem, in which it is possible to query, for a cost, 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 get to a single correct hypothesis. We provided a mapping of this problem to sequential model-based diagnosis, and presented a number of approaches for choosing a plan to query, including choosing the plan that maximizes the expected information gain, as well as the plan that is ranked highest in terms of likelihood by the plan recognition 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 currently extending this work in several directions. First, we are extending the myopic query approaches presented in the paper to employ look-ahead and consider the effects of a query on future time steps. Second, we are considering alternative models in which a query can include several plans and returns a probability about each component being correct. Lastly, we intend to deploy our approach in e-learning systems for science education, and query students about their intended activities.

REFERENCES

[1] 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. Dialog Systems for Health Communications.

[2] D. Avrahami-Zilberbrand and G. Kaminka. Incorporating observer biases in keyhole plan recognition (efficiently!). In *AAAI*, volume 7, pages 944–949, 2007.

[3] K. Balakrishnan and V. Honavar. Intelligent diagnosis systems. *Journal of Intelligent Systems*, 8(3/4):239–290, 1998.

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

[5] J. de Kleer and O. Raiman. Trading off the costs of inference vs. probing in diagnosis. In *IJCAI*, pages 1736–1741, 1995.

[6] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[7] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 838–845. IEEE, 2005.

[8] 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.

International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[9] 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.

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

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

[12] M. Ghallab, D. Nau, and P. Traverso. *Automated planning: theory & practice*. Elsevier, 2004.

[13] E. Harpstead, C. J. Maclellan, K. R. Koedinger, V. Aleven, S. P. Dow, and B. A. Myers. Investigating the Solution Space of an Open-Ended Educational Game Using Conceptual Feature Extraction. *EDM*, pages 51–58, 2013.

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

[15] L. D. Kuhn, B. Price, M. B. Do, J. Liu, R. Zhou, T. Schmidt, and J. de Kleer. Pervasive diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 40(5):932–944, 2010.

[16] 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.

[17] H. T. Ng and R. J. Mooney. Abductive plan recognition and diagnosis: A comprehensive empirical evaluation. *KR*, 92:499–508, 1992.

[18] M. Pollack. Plans as complex mental attitudes. *Intentions in communication*, 1990.

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

[20] G. Sukthankar and K. P. Sycara. Hypothesis pruning and ranking for large plan recognition problems. In *AAAI*, volume 8, pages 998–1003, 2008.

[21] 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.

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

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

[24] T. Zamir, R. Stern, and M. Kalech. Using model-based diagnosis to improve software testing. In *AAAI*, 2014.