Artificial Intelligence for Human-Robot Interaction
Papers from the AAAI 2015 Fall Symposium

Integration of Planning with Plan Recognition Using Classical Planners (Extended Abstract)

Richard G. Freedman
College of Information and Computer Sciences
University of Massachusetts Amherst
freedman@cs.umass.edu

Alex Fukunaga
Graduate School of Arts and Sciences
The University of Tokyo
fukunaga@idea.c.u-tokyo.ac.jp

Introduction

In order for robots to interact with humans in the world around them, it is important that they are not just aware of the presence of people, but also able to understand what those people are doing. In particular, interaction involves multiple agents which requires some form of coordination, and this cannot be achieved by acting blindly. The field of *plan recognition* (PR) studies methods for identifying an observed agent's task or goal given her action sequence. This is often regarded as the inverse of *planning* which, given a set of goal conditions, aims to derive a sequence of actions that will achieve the goals when performed from a given initial state. Ramírez and Geffner (2009; 2010) proposed a simple transformation of PR problems into classical planning problems for which off-the-shelf software is available for quick and efficient implementations. However, there is a reliance on the observed agent's optimality which makes this PR technique most useful as a *post-processing* step when some of the final actions are observed. In human-robot interaction (HRI), it is usually too late to interact once the humans are finished performing their tasks. In this paper, we describe ongoing work two extensions to make classical planning-based PR more applicable to the field of HRI. First, we introduce a modification to their algorithm that reduces the optimality bias's effect so that long-term goals may be recognized at earlier observations. This is then followed by methods for extracting information from these predictions so that the observing agent may run a second pass of the planner to determine its own actions to perform for a fully interactive system.

Plan Recognition for Long-Term Goals

A classical planning problem is defined by the tuple $\mathcal{P} = \langle F, I, A, G \rangle$ where F is the set of propositional statements which define the world, $I \subseteq F$ is the initial state, $G \subseteq F$ is the set of goal conditions which must be satisfied to solve the problem, and $a \in A$ is an action which the agent may perform to alter the world via add and delete lists $Add\left(a\right), Del\left(a\right) \subseteq F$. Each action also has a set of preconditions $Pre\left(a\right) \subseteq F$ which must be satisfied before the action is applicable, and it often has a cost $c\left(a\right) > 0$. Ramírez and Geffner (2010) define the PR problem as triplet $T = \langle \mathcal{P} \setminus G, \mathcal{G}, O \rangle$ where $\mathcal{P} \setminus G$ is the above planning problem without the specific goal. Instead, there are a set of possible goal sets $\mathcal{G}$ for which one is being approached via the strictly ordered (incomplete) observed action sequence O.

To transform T into a planning problem $\mathcal{P}_T$, we update the domain using a slightly different notation for clarity: $F_T = F \cup \{p_i \,| 0 \leq i \leq |O|\}$, $I_T = I \cup \{p_0\}$, and A_T is the same as A with modified $Add_T\left(a\right) = Add\left(a\right) \cup \{p_{i-1} \rightarrow p_i \,| a = o_i \in O\}$ enforcing the ordering of the observation sequence. For each goal $G \in \mathcal{G}$, a classical planner may now solve for $G + O = G \cup \left\{p_{|O|}\right\}$ to find the plan(s) which satisfy both the observations and the goal as well as $G + \overline{O} = G \cup \left\{\neg p_{|O|}\right\}$ for plans(s) which satisfy the goal and not all the observations. These goals are sufficient for generating weights which are then normalized for probabilities $P\left(G \,| O\right) = \alpha P\left(O | G\right) P\left(G\right)$. α is the normalizing constant, prior $P\left(G\right)$ is simply assumed to be uniform, and likelihood $P\left(O \,| G\right) = \alpha' exp\left(-\beta c\left(G + O\right)\right)$ assumes a Boltzman distribution where $\alpha' = P\left(O \,| G\right) + P\left(\overline{O} \,| G\right)$ and $c\left(G + O\right)$ is the cost of the optimal plan solving $G + O$.

Due to this design, the distribution strongly favors plans of shorter lengths. While this makes sense to assume in post-processing since an agent is more likely to focus on a goal without going too far out of the way, it *introduces a bias during the execution of the latent plan for short-term goals*. That is, a simpler task which requires fewer actions to complete is favored over a more complex task that the observed agent may just be starting to solve. We propose two approaches to address this issue. The first one is to use a *dynamic prior* $P_t\left(G\right) = Z^{-1} \cdot P\left(G\right) \cdot exp\left(\gamma \cdot f\left(G, t\right)\right)$ which favors goals with longer plans when time t is low and converges to the true prior as t increases. Z is the normalizing constant and we use $\gamma \in [0, 1]$ as a discount parameter to adjust the strength of foresight (as it does in the Bellman Equation). f is a decreasing nonnegative function over time that considers the cost of plans for solving goal G. Examples include thresholded polynomial $\max\left(0, c\left(G\right) - t^k\right)$ for some $k \in \mathbb{N}$ and exponential decay $c\left(G\right)^{-t}$; the function chosen plays a large role in the rate of convergence. Specifically, the rate at which this dynamic prior converges to the true prior may be derived from the derivative of the KL-Divergence:

$$\frac{dD_{KL}\left(P||P_t\right)}{dt} = \gamma \sum_{G \in \mathcal{G}} \frac{df\left(G, t\right)}{dt}\left(P_t\left(G\right) - P\left(G\right)\right)$$

This rate resembles an error-correcting feedback loop with step-size γ. In the case that an observed agent actually does favor performing simpler tasks with more short-term goals, we hypothesize that actually updating the Bayesian prior will eventually make the true prior large enough to counter the effect of the dynamic prior in favor of simpler goals.

The second method to consider is adjusting the likelihood to *consider the number of optimal plans* $P\left(O\,|G\right) = \alpha' \# \left(G + O\right) exp\left(-\beta c\left(G + O\right)\right)$. Ramírez and Geffner (2010) acknowledge ignoring this value even though it could have an impact on the distribution since the number of permutations of a partially ordered plan can yield more options for completing a task. The Boltzman distribution is still maintained since the count is a scalar multiple which may be written as a power of e. Algorithm 1 provides a method for counting the number of executable plans given a partially ordered plan in the form of a directed acyclic graph with causal edges indicating actions whose effects satisfy other actions' preconditions; the source is the initial state and the sink is the goal. This may be viewed as a compressed form of the planning graph (Blum and Furst 1997).

To avoid lag during runtime, we note that the algorithm may be run a priori for all $G \in \mathcal{G}$ regardless of O. As long as O is a subsequence of the optimal partial plan for some G, then we can find $\#\left(G + O\right)$ and $\#\left(G + \overline{O}\right)$ from the precomputations without needing to run the planner again since the optimal plan does not change. Even if O is not a subsequence of an optimal plan, then we only need to run the planner and Algorithm 1 for goal $G + O$ since the optimal plan for $G + \overline{O}$ (and thus $\#\left(G + \overline{O}\right)$) is precomputed.

Algorithm 1: Permutations of a Partially Ordered Plan

permutations(Node $from$ = SOURCE, Node to = SINK)
if $from == to$ **or** $|from.children| == 0$ **then**
  return 1 ▷ Base Case
else
  X = BFS($from, from.children \in X.ancestors$)
    ▷ Each child's branch rejoins at node X
  $totalNodes = 0$
  for *each Node* $c \in from.children$ **do**
    $nodesToXfrom[c] = nodesBetween(c, X)$
    $totalNodes += nodesToXfrom[c]$
  $perm$ = permutations(X, to) ▷ Count from rejoin
  for *each Node* $c \in from.children$ **do**
    $perm *= \binom{totalNodes}{nodesToXfrom[c]} *$ permutations(c, X)
    $totalNodes -= nodesToXfrom[c]$
  return $totalNodes$
nodesBetween(Node $from$, Node to)
if $to \in from.children$ **then**
  return 1
else
  return $1 + \sum_{c \in from.children}$ nodesBetween(c, to)

Deriving an Interactive Response

Following the probabilistic PR process modified above, the observing robot may plan its own behaviors with respect to its predictions of the human's plans. Given its own set of actions A_R and current state I_O based on the observed actions, we may define its planning problem as a centralized multiagent system $\mathcal{P}_R = \langle F, I_O, A \cup \{\text{no-op}\} \times A_R \cup \{\text{no-op}\}, G_R\rangle$ where G_R is derived from the goal probabilities and robot's purpose. We define the *necessity* of a proposition with respect to the set of goals as $N\left(p \in F\right) = \sum_{G \in \mathcal{G}} P\left(G\,|O\right) \cdot \mathbf{1}\left(p \in G\right)$ where $\mathbf{1}$ is the indicator function. A necessity of 1 implies that all goals with probability > 0 require p as a condition and 0 implies that no predicted goals require p as a condition. Then for some threshold T, we define three cases for G_R:

Assistive $G_R = \{p\,|N\left(p\right) \geq T\}$

Adversarial $G_R = \{\neg p\,|N\left(p\right) \geq T\}$

Independent $G_R = \{p\,|N\left(p\right) \geq T\} \cup G'$ where G' is a distinct task assigned to the robot

For each case, the robot may use the same off-the-shelf classical planner from the PR step to derive the joint plan it should perform alongside the human. However, this plan is optimistic since the human is acting independently and *is not guaranteed to follow the joint plan*. This introduces the need for replanning with new observations throughout the interaction; we leave identifying when to replan as future work.

To avoid replanning in mid-execution, we can also *generate local intermediate plans using landmarks* (Hoffmann, Porteous, and Sebastia 2004) for the most likely goal(s). This is similar to work by Levine and Williams (2014) where a robot recognizes local human activity at branching points in predefined plans and then executes its own plan so that later actions in the predefined plan may be performed. However, landmarks may be computed efficiently for any arbitrary plan (Keyder, Richter, and Helmert 2010) and do not have to be restricted to branching points. We can also perform local PR using these landmarks in place of $\mathcal{G}$ and then use the necessity of propositions over landmarks to derive G_R. While it seems better to just use landmarks with the initial PR process in lieu of all the proposed modifications, we note that many long-term plans have multiple parts that may be done in parallel so that the robot can drastically reduce the time to perform the entire task.

Future Work

Besides implementing and evaluating the methods proposed above, there are further directions in which this work may be extended. In particular, many real-world problems which use robots such as search and rescue (Jung, Takahashi, and Grupen 2014) rely on sensor data observations which do not provide the high-level action representations needed for classical planners. Freedman, Jung, and Zilberstein (2014) begins to address this issue by identifying clusters of sensor readings which may resemble actions, but connecting them to planners will be necessary to have interaction in practical domains. For specialized domains where we can represent the observations at this higher level, there appears to be an opportunity to use these together for specific domains of HRI.

References

Blum, A. L., and Furst, M. L. 1997. Fast planning through planning graph analysis. *Journal of Artificial Intelligence* 90(1-2):281–300.

Freedman, R. G.; Jung, H.-T.; and Zilberstein, S. 2014. Plan and activity recognition from a topic modeling perspective. In *Proc. of the 24th Int'l Conference on Automated Planning and Scheduling*, 360–364.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *Journal of Artificial Intelligence Research* 22(1):215–278.

Jung, H.-T.; Takahashi, T.; and Grupen, R. A. 2014. Human-robot emergency response - experimental platform and preliminary dataset. Technical report, University of Massachusetts Amherst.

Keren, S.; Gal, A.; and Karpas, E. 2014. Goal recognition design. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21-26, 2014*.

Keren, S.; Gal, A.; and Karpas, E. 2015. Goal recognition design for non-optimal agents. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*, 3298–3304.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and complete landmarks for and/or graphs. In *Proceedings of the Nineteenth European Conference on Artificial Intelligence*, 335–340. IOS Press.

Levine, S. J., and Williams, B. C. 2014. Concurrent plan recognition and execution for human-robot teams. In *Proceedings of the Twenty-Fourth Interntional Conference on Automated Planning and Scheduling*, 490–498.

Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 1778–1783.

Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.