# Goal Recognition over POMDPs: Inferring the Intention of a POMDP Agent

**Miquel Ramírez**
Universitat Pompeu Fabra
08018 Barcelona, SPAIN
miquel.ramirez@upf.edu

**Hector Geffner**
ICREA & Universitat Pompeu Fabra
08018 Barcelona, SPAIN
hector.geffner@upf.edu

## Abstract

Plan recognition is the problem of inferring the goals and plans of an agent from partial observations of her behavior. Recently, it has been shown that the problem can be formulated and solved using planners, reducing plan recognition to plan generation. In this work, we extend this model-based approach to plan recognition to the POMDP setting, where actions are stochastic and states are partially observable. The task is to infer a probability distribution over the possible goals of an agent whose behavior results from a POMDP model. The POMDP model is shared between agent and observer except for the true goal of the agent that is hidden to the observer. The observations are action sequences $O$ that may contain gaps as some or even most of the actions done by the agent may not be observed. We show that the posterior goal distribution $P(G|O)$ can be computed from the value function $V_G(b)$ over beliefs $b$ generated by the POMDP planner for each possible goal $G$. Some extensions of the basic framework are discussed, and a number of experiments are reported.

## 1 Introduction

Plan recognition is the problem of inferring the goals and plans of an agent from partial observations of her behavior [Cohen *et al.*, 1981; Yang, 2009]. The problem arises in a number of applications, and has been addressed using a variety of methods, including specialized methods [Avrahami-Zilberbrand and Kaminka, 2005], parsing procedures [Pynadath and Wellman, 2002; Geib and Goldman, 2009] and Bayesian networks algorithms [Bui, 2003; Liao *et al.*, 2007]. In almost all cases, the space of possible plans or activities to be recognized is assumed to be given by a suitable library of policies or plans.

Recently, two formulations have approached the plan recognition problem from a different perspective that replaces the need for a set of possible agent policies or plans, by an agent action model and a set of possible goals. The model expresses how the agent can go about achieving these goals and is used to interpret the observations. The result is a posterior probability distribution over the possible goals. In these approaches, the possible agent behaviors are encoded implicitly in the set of goals and action models, rather than explicitly as a library of plans. The advantage of these approaches to plan recognition is that they can leverage on model-based behavior generators; namely, planners. In [Ramirez and Geffner, 2009; 2010], the model is a classical planning model, namely, the initial state is fully known to agent and observer, and the actions have deterministic effects, while in [Baker *et al.*, 2009], the model is a Markov Decision Process (MDP), so that the states are fully observable, and actions have stochastic effects.

In this work, we extend the model-based approach to plan recognition over POMDP settings, where actions are stochastic and states are partially observable. The task is to infer a probability distribution over the possible goals of an agent whose behavior results from a POMDP (Partially Observable MDP) model. The model is shared between agent and observer except for the true goal of the agent that is hidden to the observer. The observations are action sequences that may contain gaps as some or even most of the actions done by the agent are not observed. We show that the posterior goal distribution can be computed from the value function over beliefs generated by a POMDP planner for each possible goal $G$. More precisely, executions are sampled from this value function assuming that the agent tends to select the actions that look best, and the likelihood of the observations $O$ given the goal $G$ is approximated from these samples. In analogy to the other cases, the goal recognition problem over POMDPs is solved using an off-the-shelf POMDP planner. While POMDP planners do not scale up as well as MDP planners, and certainly much worse than classical planners, we show that still a rich variety of recognition problems involving incomplete information can be effectively modeled and solved in this manner. The expressive power and computational feasibility of the approach will be illustrated through a number of experiments over several domains.

The paper is organized as follows. We start with an example (Sect. 2), and review previous approaches (Sect. 3) and POMDPs (Sect. 4). We then consider a preliminary formulation of POMDP goal recognition that assumes that all agent actions and observations are visible to the observer (Sect. 5), and a more general formulation that assumes neither (Sect. 6). We then report experiments (Sect. 7), discuss extensions (Sect. 8), and summarize the main results (Sect. 8).

## 2 Motivation

As an illustration of how a goal recognition problem can be naturally cast in the POMDP setting, consider an agent that is looking for an item $A$ or $B$, each of which can be in one of three drawers 1, 2, or 3, with probabilities $P(A@i)$ and $P(B@i)$ equal to:

$$P(A@1) = 0.6\ , P(A@2) = 0.4\ , P(A@3) = 0$$
$$P(B@1) = 0.1\ , P(B@2) = 0.6\ , P(B@3) = 0.3\ .$$

The actions available to the agent are to open and close the drawers, to look for an item inside an open drawer, and to grab an item from a drawer if it's known to be there. The agent is a male, and hence the probability that he doesn't observe the object in the drawer when the object is there is non-zero, say $0.2$, but the probability that he observes an object that is not there is $0$.

Let us assume that the possible goals $G_1$, $G_2$, and $G_3$ of the agent are to have item $A$, item $B$, or both, with priors $0.4$, $0.4$, and $0.2$. We want to find out the goal posterior probabilities when the behavior of the agent is partially observed. In our setting, the observer gets to see some of the actions done by the agent, but not necessarily all of them. The observer must then fill the gaps. Let us assume that it is observed that the agent opens drawer 1, then drawer 2, and then drawer 1 again; i.e.,

$$O = \{open(1), open(2), open(1)\}.$$

The most likely explanation of this observation trace is that the agent is looking for item $A$; else it wouldn't have started by looking in drawer 1 where the probability of finding $B$ is $0.1$. Then, it's likely that the agent didn't observe $A$ in that drawer, that it closed it, and then looked for $A$ in drawer 2. Then, probably the agent didn't find $A$ in drawer 2, and thus looked again in drawer 1.

Indeed, the algorithm that we will describe, concludes that the posterior probabilities for the three possible goals are $P(G_1|O) = 0.53$, $P(G_2|O) = 0.31$, and $P(G_3|O) = 0.16$, with $G_1$ as the most likely goal.

As far as we know, there are no other systems or formulations that can handle this type of scenarios where the agent gets partial observations from the environment, and the observer gets partial information about the actions of the agent, with some of the agent actions going possibly unnoticed.

## 3 Previous Approaches

As mentioned in the introduction, the problem of plan, goal, or activity recognition has been addressed in many ways, in most cases assuming that there is a library of possible plans or policies that represents the possible agent behaviors. The problem has been formulated in a variety of ways, as a deductive problem over a suitable logical theory [Kautz and Allen, 1986], a matching problem over a suitable AND/OR graph [Avrahami-Zilberbrand and Kaminka, 2005], a parsing problem over a grammar [Pynadath and Wellman, 2002; Geib and Goldman, 2009], and an inference task over a dynamic Bayesian network [Bui, 2003; Liao *et al.*, 2007].

Some recent approaches, however, attempt to map the *plan recognition problem* into *plan generation* to leverage on the performance of state-of-the-art planners. Ramirez and Geffner [2010] consider the problem of plan recognition over classical planning models where the goal of the agent is hidden to the observer. They show that the posterior distribution $P(G|O)$ over the possible agent goals $G$ given a sequence of observations $O$, can be defined from the costs $c(G, O)$ of the plans that achieve $G$ while complying with the observations $O$, and the costs $c(G, \overline{O})$ of the plans that achieve $G$ while not complying with $O$. Indeed, they define the likelihood $P(O|G)$ as a monotonic (sigmoid) function of the difference in costs $c(G, \overline{O}) - c(G, O)$. Thus, when the best plans for $G$ all comply with $O$, this difference is non-negative, and the larger the difference, the larger the likelihood $P(O|G)$. In order to compute the posterior probabilities $P(G|O)$ for a set $\mathcal{G}$ of possible goals $G$, the likelihoods $P(O|G)$ are derived in $2|\mathcal{G}|$ planner calls, and they are then plugged into Bayes rule along with the priors $P(G)$ to yield the posterior probabilities $P(G|O)$.

The other recent model-based approach to plan recognition is in the MDP setting where actions are assumed to have stochastic effects and states are fully observable [Baker *et al.*, 2009]. Baker *et. al.* show that from the value function $V_G(s)$ that captures the expected cost from state $s$ to the goal $G$, for every state $s$ and goal $G$, it is possible to define the probability that the agent will take a given action $a$ in $s$ if her goal is $G$. From this probability $P(a|s, G)$ and simple manipulations involving basic probability laws, they derive the likelihood that a sequence of actions and state observations results given that the agent starts in a state $s$ and pursues goal $G$. As before, from the likelihoods $P(O|G)$ and the goal priors $P(G)$, they derive the posterior probabilities $P(G|O)$ using Bayes rule. Once again, the main computational work is done by the planner, in this case an MDP planner, that must furnish the value function $V_G(s)$ for all goals $G$ and states $s$. Notice that in both the classical and MDP formulations, *probabilities* are inferred from *costs*; in the first case, the costs $c(G, O)$ and $c(G, \overline{O})$, in the second, the expected costs $V_G(s)$. The formulation that we develop below takes elements from both formulations while extending them to the POMDP setting where actions are stochastic and states are partially observable.

## 4 Background: Goal MDPs and POMDPs

*Shortest-path* MDPs provide a generalization of the state models traditionally used in heuristic search and planning in AI, accommodating stochastic actions and full state observability [Bertsekas, 1995]. They are given by

- a non-empty state space $S$,
- a non-empty set of goal states $S_G \subseteq S$,
- a set of actions $A$,
- probabilities $P_a(s'|s)$ for $a \in A$, $s, s' \in S$, and
- costs $c(a, s)$ for $a \in A$ and $s \in S$.

The goal states $t$ are assumed to be absorbing and cost-free; meaning $P_a(t|t) = 1$ and $c(a, t) = 0$ for all $a \in A$. Goal MDPs are shortest-path MDPs with a known initial state $s_0$ and *positive action costs* $c(a, s)$ for all $a$ and non-terminal states $s$. Shortest-path and Goal MDPs appear to be less expressive than *discounted reward* MDPs, where there is no

goal, rewards can be positive, negative, or zero, and a parameter $\gamma$, $0 < \gamma < 1$, is used to discount future rewards. Yet, the opposite is true: discounted reward MDPs can be transformed into equivalent Goal MDPs, but the opposite transformation is not possible [Bertsekas, 1995]. The same is true for discounted reward POMDPs and Goal POMDPs [Bonet and Geffner, 2009].

The solution to MDPs are functions $\pi$ mapping states into actions. The expression $V^\pi(s)$ denotes the *expected cost* that results from following the policy $\pi$ from the state $s$ to a goal state, and it can be computed by solving a system of $|S|$ linear equations. The optimal policies are well-defined if the goal is reachable from every state, and corresponds to the policies $\pi^*$ that minimize $V^\pi(s)$ over all states $s$. The optimal cost function $V^*(s) = V^\pi(s)$ for $\pi = \pi^*$, turns out to be the unique solution to the Bellman equation:

$$V(s) = \min_{a \in A} \left\{ c(a, s) + \sum_{s' \in S} P_a(s'|s) V(s') \right\} \quad (1)$$

for all $s \in S \setminus S_G$, and $V(s) = 0$ for $s \in S_G$. The Bellman equation can be solved by the *Value Iteration* (VI) method, where a value function $V$, initialized arbitrarily over non-goal states, is updated iteratively until convergence using the right-hand side of (1). The optimal policy $\pi^*$ is the policy $\pi_V$ that is *greedy* in the value function $V$

$$\pi_V(s) = \operatorname*{argmin}_{a \in A} \left\{ c(a, s) + \sum_{s' \in S} P_a(s'|s) V(s') \right\} \quad (2)$$

when $V = V^*$. Recent variants of value iteration aim to exploit the use of lower bound (admissible) cost or heuristic functions to make the updates more focused and achieve convergence on the states that are relevant. One of the first such methods is *Real-Time Dynamic Programming* (RTDP), that in each trial simulates the greedy policy $\pi_V$, updating the value function $V$ over the states that are visited [Barto *et al.*, 1995]. With a good initial lower bound $V$, RTDP and other recent heuristic search algorithms for MDPs, can deliver an optimal policy without considering many of the states in the problem.

POMDPs (Partially Observable MDPs) generalize MDPs by modeling agents that have incomplete state information [Kaelbling *et al.*, 1999] in the form of a prior belief $b_0$ that expresses a probability distribution over $S$, and a sensor model made up of a set of observation tokens $Obs$ and probabilities $Q_a(o|s)$ of observing $o \in Obs$ upon entering state $s$ after doing $a$. Formally, a *Goal POMDP* is a tuple given by:

- a non-empty state space $S$,
- an initial belief state $b_0$,
- a non-empty set of goal states $S_G \subseteq S$,
- a set of actions $A$,
- probabilities $P_a(s'|s)$ for $a \in A$, $s, s' \in S$,
- *positive* costs $c(a, s)$ for non-target states $s \in S$,
- a set of observations $Obs$, and
- probabilities $Q_a(o|s)$ for $a \in A$, $o \in Obs$, $s \in S$.

It is also assumed that goal states $t$ are cost-free, absorbing, and fully observable; i.e., $c(a, t) = 0$, $P_a(t|t) = 1$, and $t \in Obs$, so that $Q_a(t|s)$ is 1 if $s = t$ and 0 otherwise.

The most common way to solve POMDPs is by formulating them as completely observable MDPs over the *belief states* of the agent. Indeed, while the effects of actions on states cannot be predicted, the effects of actions on *belief states* can. More precisely, the belief $b_a$ that results from doing action $a$ in the belief $b$, and the belief $b_a^o$ that results from observing $o$ after doing $a$ in $b$, are:

$$b_a(s) = \sum_{s' \in S} P_a(s|s') b(s') \,, \quad (3)$$

$$b_a(o) = \sum_{s \in S} Q_a(o|s) b_a(s) \,, \quad (4)$$

$$b_a^o(s) = Q_a(o|s) b_a(s) / b_a(o) \quad \text{if } b_a(o) \neq 0. \quad (5)$$

As a result, the *partially observable* problem of going from an initial state to a goal state is transformed into the *completely observable* problem of going from one *initial belief state* into a *target belief state*. The Bellman equation for the resulting *belief* MDP is

$$V(b) = \min_{a \in A} \left\{ c(a, b) + \sum_{o \in Obs} b_a(o) V(b_a^o) \right\} \quad (6)$$

for non-target beliefs $b$ and $V^*(b_t) = 0$ otherwise, where $c(a, b)$ is the expected cost $\sum_{s \in S} c(a, s) b(s)$.

Like classical planning models, MDPs and POMDPs can be defined in compact form through the use of PDDL-like planning languages where states are valuations over a set of variables, and the goal states are the states where goal formulas (from now one, just goals) are true. If the goals are not observable, the mapping is slightly less direct and requires the use of dummy goals. Our experiments for goal recognition over POMDPs are carried on with the GPT planner [Bonet and Geffner, 2001] that allows for this type of compact representations, and solves Goal POMDP models using RTDP-Bel [Bonet and Geffner, 2000; 2009]; an adaptation of RTDP to Goal POMDPs.

## 5 Goal Recognition: Complete Observations

Our first formulation of goal recognition over POMDPs is a direct generalization of the MDP account [Baker *et al.*, 2009]. This account makes *two assumptions*. First, that the observation sequence is *complete*, meaning that $O$ contains *all the actions done by the agent*, and hence, that there are no gaps in the sequence. Second, that *the states of the MDP are fully observable not only to the agent, but also to the observer.* The assumptions are pretty restrictive but serve to reduce the goal recognition problem to a simple probabilistic inference problem. In the POMDP setting, the second assumption translates into the (partial) observations gathered by the agent being visible to the observer as well. Thus, in this setting, the observer gets two types of information: the *complete* sequence of actions $O$ done by the agent, and the corresponding sequence of POMDP observation tokens $o \in Obs$ that the agent received. In the next section, we will relax both of these assumptions.

The POMDP is assumed to be known by both the agent and the observer, except for the actual goal $G$ of the agent. Instead, the set $\mathcal{G}$ of possible goals is given along with the

priors $P(G)$. The posterior goal probabilities $P(G|O)$ can be obtained from Bayes rule:

$$P(G|O) = \alpha P(O|G)P(G) \tag{7}$$

where $\alpha$ is a normalizing constant that doesn't depend on $G$. The problem of inferring the posteriors $P(G|O)$ gets thus mapped into the problem of defining and computing the likelihoods $P(O|G)$. The key assumption is that if the agent is pursing goal $G$, the probability $P(a|b, G)$ that she will choose action $a$ in the belief state $b$ is given by the Boltzmann policy:

$$P(a|b, G) = \alpha' exp\{\beta\, Q_G(a, b)\} \tag{8}$$

where $\alpha'$ is a normalizing constant and $\beta$ captures a 'soft rationality' assumption [Baker *et al.*, 2009]: for large $\beta$, the agent acts greedily on $Q_G$ (optimally if $Q_G$ is optimal); for low $\beta$, the agent selects actions almost randomly.

The term $Q_G(a, b)$ expresses the expected cost to reach the goal $G$ from $b$ starting with the action $a$ in $b$; i.e.,

$$Q_G(a, b) = c(a, b) + \sum_{o \in O} b_a(o) V_G(b^o_a) \tag{9}$$

where $V_G$ is the value function for the POMDP assuming that the goal states are those in which $G$ is true, $c(a, b)$ is the expected cost of action $a$ in $b$, and $b_a(o)$ and $b^o_a$ as defined above, stand for the probability that the agent observes $o$ after doing action $a$ in $b$, and the probability distribution that results from doing $a$ in $b$ and observing $o$.

The observation sequence $O$ that the observer gets in this formulation is a sequence $O_{1,n}$ of pairs $(a_i, o_i)$, $i = 1, \ldots, n$, where $a_i$ is an action, and $o_i$ is the observation token that the agent obtains after doing action $a_i$. The likelihood that measures how well $G$ predicts the complete observation sequence $O$ for an agent starting in $b$, follows from the recursive decomposition

$$P(O_{i,n}|b, G) = P(a|b, G)\, b_a(o)\, P(O_{i+1,n}|b^o_a, G) \tag{10}$$

with $a = a_i$, $o = o_i$, and $P(O_{i,n}|b, G) = 1$ for $i > n$. The likelihood $P(O|G)$ is then $P(O_{1,n}|b_0, G)$, which plugged into Bayes rule (7) yields the desired posterior goal probabilities $P(G|O)$. The POMDP planner enters into this formulation by providing the expected costs $V_G(b)$ to reach $G$ from $b$ that are used via the factors $Q_G(a, b)$ for defining the probability that the agent will do action $a$ in $b$ (8).

## 6 Goal Recognition: Incomplete Observations

In the account above, the information available to the observer contains both the actions done by the agent, and the observation tokens that the agent receives from the environment. Moreover, this sequence of actions and tokens is assumed to be complete, meaning that no one is missed by the observer. In the account below, these two restrictions are removed.

As before, we assume a shared POMDP between agent and observer, except for the agent goal $G$ that is hidden to the observer but belongs to the set $\mathcal{G}$ of possible goals. The observation sequence $O$ is now a sequence of actions $a_1, \ldots, a_n$ which is not necessarily complete, meaning that some of the agent actions may have gone unnoticed to the observer, whom hence cannot assume a priori that action $a_{i+1}$ is the action that the agent did right after $a_i$. Still, as before, the posterior goal probabilities $P(G|O)$ can be derived using Bayes rule (7) from the priors $P(G)$ and the likelihoods $P(O|G)$ that can now be defined as

$$P(O|G) = \sum_{\tau} P(O|\tau) P(\tau|G) \tag{11}$$

where $\tau$ ranges over the possible executions of the agent given that she is pursuing goal $G$.

The executions $\tau$ contain the complete sequence of agent actions. We will say that *an execution $\tau$ complies with the observation sequence $O$* if the sequence $O$ is embedded in the sequence $\tau$. Taking then the probabilities $P(O|\tau)$ as 1 or 0 according to whether the execution $\tau$ complies with $O$ or not, the sum in (11) can be approximated by *sampling* as

$$P(O|G) \approx m_O/m \tag{12}$$

where $m$ is the total number of executions sampled for each goal $G$, and $m_O$ is the number of such executions that comply with $O$.

For this approximation to work, executions $\tau$ for the goal $G$ need to be sampled with probability $P(\tau|G)$. This can be accomplished by making the agent select the action $a$ in a belief $b$ with a probability $P(a|b, G)$ that results from the Boltzmann policy (8). As before, it is assumed that the POMDP planner furnishes the value function $V_G(b)$ that encodes the expected cost from $b$ to the goal.

Once the action $a$ is sampled with probability $P(a|b, G)$, the resulting observation $o$, that is no longer assumed to be available to the observer, is sampled with probability $b_a(o)$. The resulting traces $b_0, a_0, o_0, b_1, a_1, o_1, \ldots$ until the goal is reached are such that $b_{i+1} = b^o_a$ for $b = b_i$, $a = a_i$, and $o = o_i$, where $a_i$ is sampled with probability $P(a_i|b_i, G)$, and $o_i$ is sampled with probability $b_a(o_i)$ for $b = b_i$ and $a = a_i$.

The likelihoods $P(O|G)$ approximated through (12) are then plugged into Bayes rule (7) from which the posterior goal probabilities $P(G|O)$ are obtained. The key computational burden in this account results from the calculation of the value function $V_G$ over beliefs, that must be precomputed by the POMDP planner, and the simulated executions that need to be sampled for estimating the likelihoods $P(O|G)$ following (12).

## 7 Experiments

For testing the goal recognizer, we used the POMDP planner GPT built around the RTDP-BEL algorithm. The software for the goal recognition part was implemented on PYTHON, making calls to GPT when necessary. GPT supports a very expressive language for encoding POMDPs which have allowed us to test our approach over four non-trivial domains. Features of these domains are shown in Table 1. The software and example files are available from the authors.

DRAWERS is the domain described in Section 2. OFFICE is adapted from [Bui, 2003]. The agent is at the lab which consists of two rooms: one is her office, where she has a workstation and a cabinet to store her coffee cup and blank paper. The other is the club, where the coffee machine and

| Name | $\|S\|$ | $\|A\|$ | $\|Obs\|$ | $\|b_0\|$ | $\|\mathcal{G}\|$ | $\|T\|$ |
|---|---|---|---|---|---|---|
| OFFICE | 2,304 | 23 | 15 | 4 | 3 | 3.4 |
| DRAWERS | 3,072 | 16 | 16 | 6 | 3 | 4.5 |
| KITCHEN | 69,120 | 29 | 32 | 16 | 5 | 10.1 |

Table 1: Features of the four domains: number of states, actions, observation tokens, states in initial belief, and possible goals. $T$ is time in seconds to compute $V_G(b_0)$ for all goals $G$ in $\mathcal{G}$.

printer are placed. The two rooms are connected by a corridor. The agent is initially on the corridor. The printer can be out of paper, clogged, both or none. The agent goals are to print an article, have a cup of coffee or both. To print an article, the agent needs to get to her workstation, send the file to the printer queue and get to the printer. With probability 0.2 the printer gets clogged while printing. If the printer is out of paper, blank paper needs to be fetched from the cabinet. When the printer is clogged, the agent has to execute several actions to service it. To have coffee, the agent needs to pick the cup from the cabinet and go to the coffee machine. The prior of the joint goal is defined as the product of the priors for each of the individual goals, which are assumed to be equally likely.

In KITCHEN, the agent goal is to cook one of five possible dishes. There are ingredients $i1$, $i2$, $i3$, $i4$ which are placed at random on two cupboards. Each dish requires up to three different ingredients which are required to be mixed in a bowl. The agent can inspect the cupboards the find the ingredients it needs, having to approach them first. In addition, there are three objects needed – a tray, a pan and a pot – which are all located in a third cupboard. Whenever a recipe involves boiling or frying an ingredient, or a mix of them, the agent needs to place the required objects on the stove. The agent can navigate between locations, and all goals have the same prior.

The dataset is made up of a set of instances $\langle O, G\rangle$, where $O$ is an observation sequence obtained by randomly collecting either 30%, 50%, or 70% of the actions from greedy executions leading to $G$ obtained from the value function computed by GPT for each of the goals $G \in \mathcal{G}$. For each goal $G$ and each observation ratio, 10 instances $\langle O, G\rangle$ where generated in this way with $G$ as the hidden goal. Then, rather than reporting posteriors $P(G'|O)$ for all $G'$, we analyze the *binary goal classifier* that results from mapping the observation sequence $O$ into the set of *most likely* goals. These are the goals $G'$ that maximize the posterior probability $P(G'|O)$.[1] For this, we treat the pairs $\langle O, G\rangle$ as *positive* instances (P), and the pairs $\langle O, G'\rangle$ for all $G' \neq G$ as *negative* instances (N). A positive instance $\langle O, G\rangle$ is a then true positive (TP) if $G$ is found most likely given $O$ and is a false positive (FP) if not. Similarly, a negative instance $\langle O, G\rangle$ is a true negative (TN) if $G$ is not among the most likely goals given $O$, and else is a false negative (FN). We denote the resulting goal classifier as $\mathrm{GR}(m, \beta)$, where $m$ is the number of samples for each goal used in (12) and $\beta$ is the level of noise in the action selection (8). The classifiers $\mathrm{GR}(m, \beta)$ are evaluated according to the standard measures TPR, FPR, ACC, and PPV

[1] We consider two numbers $x$, $x'$ equal when $|x - x'| < 10^{-7}$.

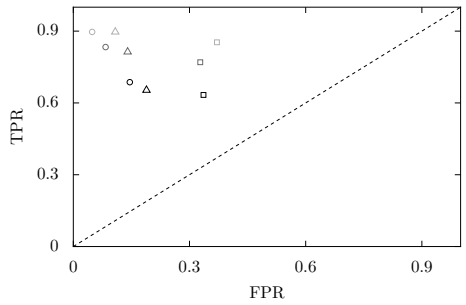


Figure 1: ROC graph showing aggregate results for goal classifier $\mathrm{GR}(m, \beta)$ for different $m$ and $\beta$ values (number of samples and noise level in action selection). Squares, triangles and circles denote different $m$ values: 100, 1000 and 10000. Black, gray, and light gray denote different $\beta$ values: 1, 10, 40. Results for random strategy represented by the dotted line.

| Domain | Obs % | L | T | ACC | PPV | TPR |
|---|---|---|---|---|---|---|
| office | 30 | 4.9 | 24.6 | 0.99 | 0.97 | 1.00 |
| | 50 | 7.6 | 24.7 | 1.00 | 1.00 | 1.00 |
| | 70 | 10.8 | 24.8 | 1.00 | 1.00 | 1.00 |
| kitchen | 30 | 3.8 | 95.2 | 0.86 | 0.73 | 0.73 |
| | 50 | 5.8 | 95.1 | 0.93 | 0.85 | 0.85 |
| | 70 | 8.3 | 95.2 | 0.98 | 0.95 | 0.95 |
| drawers | 30 | 2.9 | 38.8 | 0.84 | 0.77 | 0.77 |
| | 50 | 3.9 | 38.8 | 0.87 | 0.80 | 0.80 |
| | 70 | 6.0 | 38.8 | 0.96 | 0.93 | 0.93 |

Table 2: Performance of classifier $\mathrm{GR}(m = 10,000, \beta = 40)$: domains, observation ratio, average length of observation sequences (L), average time in seconds to process $O$ (T), average accuracy (ACC), precision (PPV) and True Positive rate (TPR).

that stand for the True Positive Rate, False Positive Rate, Accuracy, and Precision, defined by the ratios $TP/P$, $FP/N$, $TP + TN/P + N$, and $TP/TP + FP$ respectively.

Figure 1 shows the aggregate results of the goal recognizer $\mathrm{GR}(m, \beta)$ over the four domains in the form of a ROC graph [Fawcett, 2006] that plots TPR vs. FPR. As it can be seen, the performance of the goal classifier approaches the optimal vertex $(1, 0)$ as the number of samples $m$ becomes large. Performance is good – high TPR, low FPR – for $m \geq 1000$ and high $\beta$ values. Table 2 offers a detailed picture of the performance of the classifier for such values of $m$ and $\beta$. In all domains we see how the accuracy of goal recognition increases as more information is conveyed by the input observation sequence, with near optimal results in all cases as the ratio of observations approaches 70%.

## 8 Extensions

The model above for goal recognition over POMDPs is simple but quite expressive, and still there are a number of natural extensions, some of which we describe next.

• *Agent POMDP model partially known by observer:* in the above formulation, the agent POMDP model is known by the observer except for the hidden goal. Incomplete information about the *initial belief state* $b_0$ of the agent, however, can be accommodated as well. The simplest approach is to define

a set $B_0$ of possible initial belief states $b_0$ each with a probability $P(b_0)$. The formulation can then be generalized to deal with both hidden goals $G$ and hidden initial belief states $b_0$, and the posterior probabilities over the collection of such pairs can be computed in a similar manner.

• *Failure to observe and actions that must be observed:* as argued in [Geib and Goldman, 2009], information about actions $a$ that if done, must always be observed, is valuable, as the absence of such actions from $O$ imply that they were not done. This information can be used in a direct manner by adjusting the notion of when a sample execution $\tau$ complies with $O$. In the presence of *must-see* actions, executions $\tau$ comply with $O$ when $\tau$ embeds $O$, *and* every *must-see* action appears as many times in $\tau$ as in $O$.

• *Observing what the agent observes:* we have assumed that the observer gets a partial trace of the actions done by the agent and nothing else. Yet, if the observer gets to see some of the observation tokens $o \in Obs$ gathered by the agent, she can use this information as well. In particular, the number $m_O$ in (12) would then be set to the number of sampled executions for $G$ that comply with both $O$ and $Obs$.

• *Noise in the agent-observer channel:* if the observer gets to see the actions done by the agent through a noisy channel where actions can be mixed up, the problem of determining where a sample execution $\tau$ complies with the observations $O$ is no longer a *boolean* problem where $P(O|\tau)$ is either 0 or 1 but a probabilistic inference problem that can be solved in linear-time with Hidden Markov Model (HMM) algorithms, that would yield a probability $P(O|\tau)$ in the interval $[0, 1]$. For this, the model must be extended with probabilities $Q'(o|a)$ of observing token $o$ from the execution of action $a$, and hidden chain variables $t_i = j$ expressing that the observation token $o_i$ in $O = o_1, \ldots, o_n$ has been generated by action $a_j$ in the sample execution $\tau = a_1, \ldots, a_m$.

## 9 Summary

We have formulated and tested a new formulation of goal recognition for settings where the agent has partial information about the environment and the observer has partial information about the actions of the agent. The posterior goal probabilities $G$ for the hidden goals $G \in \mathcal{G}$ are computed from Bayes rule using given priors $P(G)$ and likelihoods $P(O|G)$ approximated in two steps: using first a POMDP planner to produce the expected costs $V_G$ from beliefs to goals, and using these costs to sample executions. A number of direct extensions have also been discussed and a number of experiments have been reported. The work assumes that there is a POMDP model shared by agent and observer. The ability to do goal and belief recognition when the two models do not match remains as an interesting challenge for future work.

### Acknowledgments

H. Geffner is partially supported by grants TIN2009-10232, MICINN, Spain, and EC-7PM-SpaceBook.

## References

[Avrahami-Zilberbrand and Kaminka, 2005] D. Avrahami-Zilberbrand and G. A. Kaminka. Fast and complete symbolic plan recognition. In *Proceedings of IJCAI*, pages 653–658, 2005.

[Baker *et al.*, 2009] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[Barto *et al.*, 1995] A. Barto, S. Bradtke, and S. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138, 1995.

[Bertsekas, 1995] D. Bertsekas. *Dynamic Programming and Optimal Control, Vols 1 and 2.* Athena Scientific, 1995.

[Bonet and Geffner, 2000] B. Bonet and H. Geffner. Planning with incomplete information as heuristic search in belief space. In *Proc. of AIPS-2000*, pages 52–61. AAAI Press, 2000.

[Bonet and Geffner, 2001] B. Bonet and H. Geffner. Gpt: A tool for planning with uncertainty and partial information. In *Proc. IJCAI Workshop on Planning with Uncertainty and Partial Information*, 2001.

[Bonet and Geffner, 2009] B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. Point-based algorithms. In *Proceedings IJCAI-09*, pages 1641–1646, 2009.

[Bui, 2003] H.H. Bui. A general model for online probabilistic plan recognition. In *Proc. IJCAI-03*, pages 1309–1318, 2003.

[Cohen *et al.*, 1981] P. R. Cohen, C. R. Perrault, and J. F. Allen. Beyond question answering. In W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing*. LEA, 1981.

[Fawcett, 2006] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, (27):861–874, 2006.

[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.

[Kaelbling *et al.*, 1999] L. P. Kaelbling, M. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1999.

[Kautz and Allen, 1986] H. Kautz and J. F. Allen. Generalized plan recognition. In *Proc. AAAI-86*, pages 32–37, 1986.

[Liao *et al.*, 2007] L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

[Pynadath and Wellman, 2002] D.V. Pynadath and M.P. Wellman. Generalized queries on probabilistic context-free grammars. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):65–77, 2002.

[Ramirez and Geffner, 2009] M. Ramirez and H. Geffner. Plan recognition as planning. In *Proc. 21st Intl. Joint Conf. on Artificial Intelligence*, pages 1778–1783. AAAI Press, 2009.

[Ramirez and Geffner, 2010] M. Ramirez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proc. AAAI-10*. AAAI Press, 2010.

[Yang, 2009] Q. Yang. Activity Recognition: Linking low-level sensors to high-level intelligence. In *Proc. IJCAI-09*, pages 20–26, 2009.