Goal Recognition as Planning with Unknown Domain Models

Diego Aineto[1] , **Sergio Jiménez**[1] , **Eva Onaindia**[1] and , **Miquel Ramírez**[2]

[1]Departamento de Sistemas Informáticos y Computación. Universitat Politècnica de València. Valencia, Spain
[2]School of Computing and Information Systems. The University of Melbourne. Melbourne, Victoria. Australia

{dieaigar,serjice,onaindia}@dsic.upv.es, miquel.ramirez@unimelb.edu.au

Abstract

The paper shows how to relax one key assumption of the *plan recognition as planning* approach for *goal recognition* that is knowing the action model of the observed agent. The paper introduces a novel formulation that fits together the *learning of planning action models* with *plan recognition as planning*. The empirical evaluation evidences that this novel formulation allows to solve standard goal recognition benchmarks without *a priori* knowing the action model of the observed agent.

1 Introduction

Goal recognition is a particular classification task in which each class represents a different goal and each example is an observation of an agent acting to achieve one of that goals. Despite there exists a wide range of diffenrent approaches for *goal recognition*, *plan recognition as planning* [Ramírez and Geffner, 2009; Ramírez, 2012] is one of the most appealing since it is at the core of various activity recognition tasks such as, *goal recognition design* [Keren *et al.*, 2014], *deceptive planning* [Masters and Sardina, 2017], *planning for transparency* [MacNally *et al.*, 2018] or *counter-planning* [Pozanco *et al.*, 2018].

Plan recognition as planning leverages the action model of the observed agent and an off-the-shelf classical planner to compute the most likely goal of that agent. In this paper we show that we can relax the key assumption of the *plan recognition as planning* approach for *goal recognition* that is having an action model of the observed agent. In particular, the paper introduces a novel formulation that fits together the *learning of planning action models* with the *plan recognition as planning* approach. The evaluation of our formulation evidences that it allows to solve goal recognition tasks, even when the action model of the observed is unknown, using an off-the-shelf classical planner.

2 Background

This section formalizes the *planning model* we follow as well as the kind of *observations* that are given as input to the *goal recognition* task.

2.1 Classical planning with conditional effects

Let F be the set of propositional state variables (*fluents*) describing a state. A *literal* l is a valuation of a fluent $f \in F$; i.e. either $l = f$ or $l = \neg f$. A set of literals L represents a partial assignment of values to fluents (without loss of generality, we will assume that L does not contain conflicting values). Given L, let $\neg L = \{\neg l : l \in L\}$ be its complement. We use $\mathcal{L}(F)$ to denote the set of all literal sets on F; i.e. all partial assignments of values to fluents. A *state* s is a full assignment of values to fluents; $|s| = |F|$.

A *classical planning frame* is a tuple $\Phi = \langle F, A\rangle$, where F is a set of fluents and A is a set of *actions*. Each classical planning action $a \in A$ has a precondition $\mathsf{pre}(a) \in \mathcal{L}(F)$, a set of effects $\mathsf{eff}(a) \in \mathcal{L}(F)$, and a positive action cost $cost(a)$. The semantics of actions $a \in A$ is specified with two functions: $\rho(s, a)$ denotes whether action a is *applicable* in a state s and $\theta(s, a)$ denotes the *successor state* that results of applying action a in a state s. Then, $\rho(s, a)$ holds iff $\mathsf{pre}(a) \subseteq s$, i.e. if its precondition holds in s. The result of executing an applicable action $a \in A$ in a state s is a new state $\theta(s, a) = (s \setminus \neg\mathsf{eff}(a)) \cup \mathsf{eff}(a)$. Subtracting the complement of $\mathsf{eff}(a)$ from s ensures that $\theta(s, a)$ remains a well-defined state. The subset of action effects that assign a positive value to a state fluent is called *positive effects* and denoted by $\mathsf{eff}^+(a) \in \mathsf{eff}(a)$ while $\mathsf{eff}^-(a) \in \mathsf{eff}(a)$ denotes the *negative effects* of an action $a \in A$.

A *classical planning problem* is a tuple $P = \langle F, A, I, G\rangle$, where I is the initial state and $G \in \mathcal{L}(F)$ is the set of goal conditions over the state variables. A *plan* π is an action sequence $\pi = \langle a_1, \ldots, a_n\rangle$, with $|\pi| = n$ denoting its *plan length* and $cost(\pi) = \sum_{a\in\pi} cost(a)$ its *plan cost*. The execution of π on the initial state I of P induces a *trajectory* $\tau(\pi, s_0) = \langle s_0, a_1, s_1, \ldots, a_n, s_n\rangle$ such that $s_0 = I$ and, for each $1 \leq i \leq n$, it holds $\rho(s_{i-1}, a_i)$ and $s_i = \theta(s_{i-1}, a_i)$. A plan π solves P iff the induced *trajectory* $\tau(\pi, s_0)$ reaches a final state $G \subseteq s_n$, where all goal conditions are met. A solution plan is *optimal* iff its cost is minimal.

An *action with conditional effects* $a_c \in A$ is defined as a set of preconditions $\mathsf{pre}(a_c) \in \mathcal{L}(F)$ and a set of *conditional effects* $\mathsf{cond}(a_c)$. Each conditional effect $C \rhd E \in \mathsf{cond}(a_c)$ is composed of two sets of literals: $C \in \mathcal{L}(F)$, the *condition*, and $E \in \mathcal{L}(F)$, the *effect*. An action a_c is applicable in a state s if $\rho(s, a_c)$ is true, and the result of applying action a_c in state s is $\theta(s, a_c) = \{s \setminus \neg\mathsf{eff}_c(s, a) \cup \mathsf{eff}_c(s, a)\}$ where

$\mathsf{eff}_c(s,a)$ are the *triggered effects* resulting from the action application (conditional effects whose conditions hold in s):

$$\mathsf{eff}_c(s,a) = \bigcup_{C \rhd E \in \mathsf{cond}(a_c), C \subseteq s} E,$$

2.2 The observation model

Given a planning problem $P = \langle F, A, I, G \rangle$, a plan π and a trajectory $\tau(\pi, P)$, we define the *observation of the trajectory* as an interleaved combination of actions and states that represents the observation from the execution of π in P. Formally, $\mathcal{O}(\tau) = \langle s_0^o, a_1^o, s_1^o \dots, a_l^o, s_m^o \rangle$, $s_0^o = I$, and:

- The **observed actions** are consistent with π, which means that $\langle a_1^o, \dots, a_l^o \rangle$ is a sub-sequence of π. Specifically, the number of observed actions, l, can range from 0 (fully unobservable action sequence) to $|\pi|$ (fully observable action sequence).
- The **observed states** $\langle s_0^o, s_1^o, \dots, s_m^o \rangle$ is a sequence of possibly *partially observable states*, except for the initial state s_0^o, which is fully observable. A partially observable state s_i^o is one in which $|s_i^o| < |F|$; i.e., a state in which at least a fluent of F is not observable. Note that this definition also comprises the case $|s_i^o| = 0$, when the state is fully unobservable. Whatever the sequence of observed states of $\mathcal{O}(\tau)$ is, it must be consistent with the sequence of states of $\tau(\pi, P)$, meaning that $\forall i, s_i^o \subseteq s_i$. In practice, the number of observed states, m, range from 1 (the initial state, at least), to $|\pi|+1$, and the observed intermediate states will comprise a number of fluents between $[1, |F|]$.

We assume a bijective monotone mapping between actions/states of trajectories and observations [Ramírez and Geffner, 2009], thus also granting the inverse consistency relationship (the trajectory is a superset of the observation). Therefore, transiting between two consecutive observed states in $\mathcal{O}(\tau)$ may require the execution of more than a single action ($\theta(s_i^o, \langle a_1, \dots, a_k \rangle) = s_{i+1}^o$, where $k \geq 1$ is unknown but finite. In other words, having $\mathcal{O}(\tau)$ does not imply knowing the actual length of π.

2.3 Goal recognition as classical planning

Goal recognition is a particular classification task in which each class represents a different goal $g \in G[\cdot]$ and each example is an $\mathcal{O}(\tau)$ observation of an agent acting to achieve one of the input goals in $G[\cdot]$. Following the *naive Bayes classifier*, the *solution* to the *goal recognition* task is the subset of goals in $G[\cdot]$ that maximizes this expression.

$$argmax_{g \in G[\cdot]} P(\mathcal{O}|g)P(g). \quad (1)$$

The *plan recognition as planning* approach shows how to compute estimates of the $P(\mathcal{O}|g)$ likelihood leveraging the action model of the observed agent and an off-the-shelf classical planner. More precisely, given a *classical planning problem* $P = \langle F, A, I, G[\cdot] \rangle$, where $G[\cdot]$ represents the set of possible goals, then the *plan recognition as planning* approach estimates the $P(\mathcal{O}|g)$ by computing the cost difference between of solution plans to these two different classical planning problems:

- $P'_\top$, that constrains problem P to achieve $g \in G[\cdot]$ through a plan $\pi_\top$ *consistent* with the input observation $\mathcal{O}(\tau)$.
- $P'_\bot$, that constrains problem P to achieve $g \in G[\cdot]$ through a plan $\pi_\bot$ *inconsistent* with $\mathcal{O}(\tau)$.

The higher the value of this cost difference $\Delta(cost(\pi_\top), cost(\pi_\bot))$, the better $g \in G[\cdot]$ predicts $\mathcal{O}(\tau)$ and hence, the higher $P(\mathcal{O}|g)$ likelihood. The function used by the *plan recognition as planning* approach for mapping the previous cost difference into likelihoods is the sigmoid function:

$$P(\mathcal{O}|g) = \frac{1}{1 + e^{-\beta\Delta(cost(\pi_\top), cost(\pi_\bot))}} \quad (2)$$

This expression is derived from the assumption that while the observed agent is not perfectly rational, he is more likely to follow cheaper plans, according to a Boltzmann distribution. The larger the value of β, the more rational the agent, and the less likely that he will follow suboptimal plans.

Recent works show that faster estimates of this $P(\mathcal{O}|g)$ likelihood can also be computed using relaxations of the $P'_\top$ and $P'_\bot$ classical planning tasks [Pereira *et al.*, 2017].

3 Goal Recognition as Planning with Unknown Domain Models

This section shows that $cost(\pi_\top)$ and $cost(\pi_\bot)$, and hence an approximation to the $P(\mathcal{O}|g)$ likelihood, can also be computed with classical planing when the action model of the observed agent is *unknown*.

4 Evaluation

5 Conclusions

References

[Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *International Conference on Automated Planning and Scheduling, (ICAPS-14)*, pages 154–162, 2014.

[MacNally *et al.*, 2018] Aleck M MacNally, Nir Lipovetzky, Miquel Ramirez, and Adrian R Pearce. Action selection for transparent planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1327–1335. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[Masters and Sardina, 2017] Peta Masters and Sebastian Sardina. Deceptive path-planning. In *IJCAI 2017*, pages 4368–4375. AAAI Press, 2017.

[Pereira *et al.*, 2017] Ramon Fraga Pereira, Nir Oren, and Felipe Meneguzzi. Landmark-based heuristics for goal recognition. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. AAAI Press, 2017.

[Pozanco *et al.*, 2018] Alberto Pozanco, Yolanda E.-Martín, Susana Fernández, and Daniel Borrajo. Counterplanning using goal recognition and landmarks. In *International Joint Conference on Artificial Intelligence, (IJCAI-18)*, pages 4808–4814, 2018.

[Ramírez and Geffner, 2009] Miquel Ramírez and Hector Geffner. Plan recognition as planning. In *International Joint conference on Artifical Intelligence, (IJCAI-09)*, pages 1778–1783. AAAI Press, 2009.

[Ramírez, 2012] Miquel Ramírez. *Plan recognition as planning*. PhD thesis, Universitat Pompeu Fabra, 2012.