The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Inferring Agent's Goals from Observing Successful Traces

Guillaume Lorthioir*1*2 Katsumi Inoue*1*2 Gauvain Bourgne*3

*1National Institute of Informatics
*2Department of Informatics, SOKENDAI (The Graduate University for Advanced Studies)
*3CNRS & Sorbonne Universités, UMR 7606, LIP6

Intention recognition is the task of inferring the intentions and goals of an agent. Intention recognition has many applications. Especially, it can be very useful in the context of intelligent personal assistants like robots or mobile applications, smart environments, and monitoring user needs. We present a method to infer the possible goals of an agent by observing him in a series of successful attempts to reach them. We model this problem as a case of concept learning and propose an algorithm to produce concise hypotheses. However, this first proposal does not take into account the sequential nature of our observations and we discuss how we can infer better hypotheses when we can make some assumption about the behavior of the agents and use background knowledge about the dynamics of the environment. Then we talk about future work to improve our method.

1. Introduction

We are entering an era where new technologies are increasingly present, and soon artificial intelligence will be everywhere. Home automation, robotics, intelligent personal assistants, they will soon be interacting with humans on a daily basis. But for this, we need an artificial intelligence able to understand and interpret human's intention, which is not trivial. Even currently, there are many cases where the knowledge of an agent's intentions is useful, for cooperation or competition in a multi-agent system or even for a smart-phone application which tries to help the user to do something. The ISS-CAD problem [E-Martin et al., 2015] is a good illustration of this problem, where a free-flying robot is observing an astronaut performing a task in the International Space Station (ISS) and he has to help him.

This problem is called "Plan recognition" or "Intention recognition" and has been investigated in AI research and has many applications. [Schmidt et al., 1978] were the first ones to introduce the problem and treat it from a psychological point of view. [Charniak and Goldman, 1993] use Bayesian models and [Geib and Goldman, 2009] use also a probabilistic algorithm for the plan recognition. [Singla and Mooney, 2011, Ha et al., 2011] mix probabilistic and logic approaches, they found a very interesting field of application for plan recognition, which is digital games. [Carberry, 2001] describes the plan recognition problem and surveyed the current ways to tackle this problem. In this paper, we focus on a sub-field of intention recognition which is called "Goal-recognition" and which concern understanding of the goals of an agent several related recent work [Cardona-Rivera and Michael Young, 2017, E-Martin and Smith, 2017, Mirsky et al., 2017, Goldman et al., 2018, Vered et al., 2018] show that goal recognition is growing of interest.

However, there are only few approaches using propositional logic for goal recognition. It is this lack of literature that has led us to focus our work on this aspect of the goal recognition problem. [Hong, 2001] use propositional logic but he combines this to a graph representation, we use a different approach which consist of combining propositional logic and concept learning. We are among the first to do it. We try to guess the goal of the agent, the state of things that the agent is trying to achieve. Indeed, many previous work try to solve the plan recognition problem by using a set of possible goals for the agent and try to guess which one is more likely, as it is the case for example in the work of [Lang, 2004]. But often they assume that this set of possible goals is given, which is usually not the case.

In this paper, we will talk about the method to infer the possible goals of an agent by using concept learning that we introduced in our previous work [Lorthioir et al., 2018]. We review our method and describe the future extensions of this one. First, Section 2. will explain how to formalize the problem of inferring goals from the observation of successful scenarios as a concept learning task, we will explain how to use some assumptions on the agent decision process and the environment to improve the results of our algorithm in Section 3. We will detail a way to take into account the agents' preferences regarding their goals, and those provide additional information about the agents' goals in Section 4. Finally we will conclude this paper with Section 5.

Contact: Guillaume Lorthioir, National Institute of Informatics, 2 Chome-1-2 Hitotsubashi, Chiyoda, Tokyo, Japan, 101-8430, lorthioir@nii.ac.jp

2. Problem Formalisation

What is our problem exactly? Our objective is to infer the possible goals of an agent by observing him in a series of successful attempts to reach them. We thus assume some training process in which we observe the agent in a series of scenarios in which he performs actions in some environment until he satisfies one of the goals we are trying to guess. These observed scenarios will be modelled as a set of observed traces describing the successive states of the environment and actions of the agent. We consider environments with discrete time and no exogenous events: each action performed by the agent thus corresponds to a change of state. Here we model the state of the environment as a series of discrete-valued attributes : N variables var_i with $i \in \{1,\dots,N\}$ taking their values in variable domains $D_j = \{val_1^i,\dots,val_{N_i}^i\}$. We build an atomic representation by converting all the couples $var_i = val_j^i$ into atoms $var_i^{val_j^i}$, denoting by $\mathcal{L}$ the set of all these atoms. A *state* S is then defined as a set of atoms $var_i^{val_j^i}$ from $\mathcal{L}$ where each var_i appears once and a *trace* is defined as a sequence of couples (S_i, a_i) where $S_i \subset \mathcal{L}$ is a state and a_i an action (taken from a finite set of actions $\mathcal{A}$). We do not go into the detail of the environment's dynamics, but they can be abstracted away by some function $next$ from $2^{\mathcal{L}} \times \mathcal{A}$ to $2^{2^{\mathcal{L}}}$ which, given a state S and an action a gives the set of possible states that can be reached from S by performing a. When the environment is deterministic, $next(S,a)$ corresponds to a single state. Since they come from observations, traces are assumed to respect this dynamic, meaning that if i is not the last index of the trace, $S_{i+1} \in next(S_i, a_i)$.

To define successful traces we consider a special action `success` without effects ($\forall S, next(S, \texttt{success}) = \{S\}$), which the agent performs whenever he reaches his goal. A successful trace is thus a trace $T = (S_0,a_0),\dots,(S_k,a_k)$ where $a_k =$`success` and for $i < k$, $a_i \neq$`success`. This means that a successful trace is a trace which ends in the first state where the goal of the agent is satisfied. Given a trace $T = (S_0,a_0),\dots,(S_k,a_k)$, we denote by $endS(T)$ the last state S_k of a trace and by $intS(T)$ the set of intermediate states $\{S_0,\dots,S_{k-1}\}$. The input of our problem is a set of successful traces $\Sigma = \{T_0,\dots,T_l\}$ and our objective is to infer from that some hypothesis about the agent's goal, which will be expressed as a propositional formula over $\mathcal{L}$ which should be satisfied by some state (by interpreting states as the conjunction of their atoms) if and only if the goal is reached. We assume here that the goal depends only on the state and not on the way to reach it. The agent just needs to reach a state where the atoms composing the state satisfy his goal, no matter how he reaches it. We want the hypothesis written in the disjunctive normal form since an agent can have several goals. More precisely we want hypotheses of the form $H = C_0 \vee C_1 \vee ... \vee C_m$ where each $C_i = x_0 \wedge ... \wedge x_n$ is a conjunction of atoms of $\mathcal{L}$. Then, given $H = C_0 \vee C_1 \vee ... \vee C_m$, a state $S = \{y_0,\dots,y_n\}$ satisfies H if and only if $\bigwedge_{y_i \in S} y_i \models H$, that is, if and only if there exists $i \in \{0,\dots,m\}$ such that $C_i \subseteq S$.

Even without knowing anything about the behavior of the agents or the dynamics of the system, these observations give a series of states in which we know whether the goal is reached or not. Namely, we can build the set $S_{positive}$ of successful states by including in it all end-states of successful traces from Σ, that is, $S_{positive} = \{endS(T) | T \in \Sigma\}$. Likewise, we can build the set $S_{negative}$ of unsuccessful states by taking the union of all intermediate states, that is, $S_{negative} = \bigcup_{T\in\Sigma} intS(T)$. Given the definition of a successful trace, it means that the agent's goal is satisfied only by the elements of $S_{positive}$ and under no circumstances by an element of $S_{negative}$. The problem of inferring the goals of the agent is then equivalent to a concept learning problem where the states that we put in the set $S_{positive}$ are the positive examples and the states that we put in $S_{negative}$ are the negative ones. A hypothesis H will be said to be consistent with our data if it is satisfied by all the elements of $S_{positive}$ and by none of the elements of $S_{negative}$. We want to obtain such a hypothesis as an output of our problem. We created an algorithm to treat this problem and produce such a hypothesis, the algorithm can be found in [Lorthioir et al., 2018].

3. Inferring the Agent's Model and Environment Rules from Data

In previous sections, our input Σ, a set of successful traces, is reduced to two sets of states $S_{positive}$ and $S_{negative}$. By doing so, we do not use the information contained in Σ about the order of the explored states and the actions performed by the agents at each step. The advantage of ignoring these aspects is that we can infer possible goals without assuming more about the agent than what is induced by the definition of successful traces, that is, the agent stops (with a `success` action) as soon as his goal is reached and this goal is dependent only on the current state. However, if we know the dynamics of the environment, it seems sensible to derive some information based on what the agent chose to do given what it could have done. We explain in [Lorthioir et al., 2018] how to use such knowledge to improve the results of our algorithm.

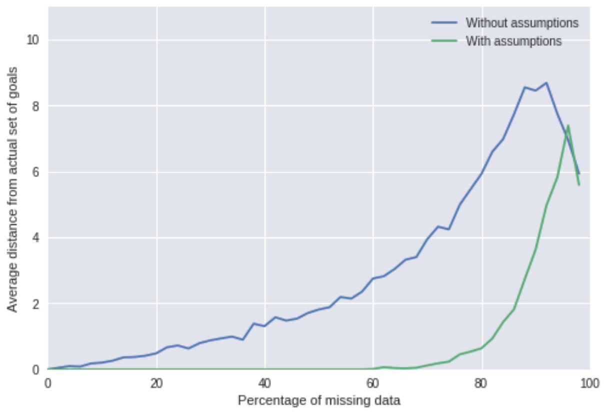

Figure 1: Comparison of the average syntactic distance from the actual goals in function of the percentage of missing data with and without assumptions on the agent.

Figure 1 allows us to see the point to have this last assumptions about the agent. On this figure the curves represent a distance similar to the Hamming one, between the hypothesis about the agent's goals generated by our algorithm and the actual agent's goals according to the percentage of data about the agent traces. More details about this distance can be found in [Lorthioir et al., 2018]. For this experiment we use the same amount of data for the two curve at the beginning. Then we make two assumptions on the agent which are that we know the action model of the agent and that we know that if he can reach his goal

The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

with one move at the step $t-1$ then at the step t he will reach it. These are not very strong assumptions. These assumptions allow us to generate more data from the existing one. In total, we obtain almost three times more data with the assumptions about the agent than without the assumptions. As we can see in figure 1 it allows us to generate a hypothesis closer to the actual goals than when we use no assumptions.

Actually, we have two interesting methods that we wish to exploit for the deduction of the rules of action of agents and the environment. The first one is to use LFIT [Inoue et al., 2014] which is a framework for learning normal logic programs from transitions of interpretations, which means that, since we use a logical representation of the world, we will be able to infer the possible action model of the agent and the rules of the environment. Effectively, to use LFIT we just need a trace of the evolution of the states of the world in a chronological order. Which are in fact the same data that we use to infer the agent's goals. Which means that we do not need an important modification of the data collecting process to incorporate LFIT to our method, which is a good point. After processing these data with LFIT we will obtain a set of rules that correspond to the possible transition from a state of the world to another. We will then have for each state of the world (observed previously in the collected data), the different actions that the agent did when he was in this state and the states of the world that he reached after these actions. So we can infer a potential action model of the agent and the dynamic of the environment based on the observed transitions. But unfortunately LFIT is not really robust against the lack of data and the noise, especially the data need to cover as many transitions as possible between all the different possible states of the world. Which is not very convenient because usually the agent that we observe is not acting randomly and so, many possible transitions will never been observe. This is where our second method comes in. SMILE [Bourgne et al., 2007], another method that we can use to learn the action model of the agent. This method uses the same kind of data than LFIT so here also the modification of the data collecting process to use SMILE and our method are not too cumbersome. Even if SMILE is usually used for a multi-agent system because it is, in fact, the different agents who will learn information about the other agents of the system, we can just create an agent "Observer" who will learn the action model of the observed agent. SMILE is more robust to the lack of data than LFIT but not as effective in some cases. So we will choose which method to use based on the quantity of data obtained during the observation phase. We already tried SMILE and LFIT with our data to extract the action model of the agent and the rules of the environment. We obtained promising results and so, we want to continue in this way. But we do not have combined our method and these two algorithms yet.

4. Incorporating Agents' Preferences

To improve our method and provide some more interesting hypothesis about the agents' goals, we thought about incorporate a preferences function about the goals of these ones. Which means that our hypothesis will provide us with a set of possible goals for an agent and ordered these goals in function of the preferences of the agent. We want to start first with a simple model of preference, by simple we mean that if the agent prefers to reach the goal A rather than the goal B his preferences will not take into account the difficulty to reach these two goals. In other words, if for example the goal A is preferred to the goal B but to reach the goal A the agent needs twenty actions more than to reach the goal B, in this case, we might think that in function of the cost of the actions, the agent could finally prefer to reach B. But since we do not take into account the cost of the actions in our model we are not going to deal with this case right now. However, it's not really difficult to integrate this to our model because we can represent the cost of the agents' actions by some atoms that we will add to our representation of the world, but the computation time of our algorithm is likely to increase drastically. The process of inferring agents' preferences will take place after the goals inference in our method. Because, of course, we need to know the agents' goals to be able to order them. So once we have the agent's goals, we also need to know the agent's action model, then, for each intermediate state of the trace of the agent (see Section 2.) we can see if there is other reachable goals that the one reached in the final state of the trace and if so, compare the number of time that a goal has been preferred to another one. If between two goals one has always been preferred and if it happened several times, we can assume a strict preference between them $A > B$. Otherwise the preference will be more moderate $A \succeq B$.

Given an order of priority $>_p$ about the goals of an agent, we can translate the fact $A >_p B$ in a logic formula by assuming that if the agent finally reaches the goal B it is because the goal A was not reachable. Then we can translate this in the logic formula $A \vee \neg A \wedge B$ (where A and B are atoms conjunctions) by using our language $\mathcal{L}$ described in section 2. Likewise, if we have the preference $A >_p B >_p C$ on the agent's goals we can write this $A \vee \neg A \wedge B \vee \neg A \wedge \neg B \wedge C$. With such a translation we could write all the possible strict orders of preference on the agent's goals, and since the writing is still in disjunctive normal form (DNF) we can modify a bit our algorithm and return a disjunction of such DNF as a result and thus obtain the possible goals of the agent and his preferences for these goals. If the results are promising, we will then try to take into account the fact stated earlier, which is that under some special conditions the agents can change their preferences (especially considering the actions' cost).

5. Conclusion

We saw that the Intention Recognition and Goal Recognition have a wide variety of applications, especially in the coming years with the development of artificial intelligence, robotics, and smart assistance. Unfortunately, several serious problems still slow down the use of plan recognition in large-scale real-world situations. [Carberry, 2001] describes these problems at the end of her paper. But we can also add another problem more related to our work, which is that in reality we do not really know when an agent reaches his goal, but we will come back on this problem later. We reminded our method and showed that it is quite different than the previous works on intention or goal recognition. Because few work are focused on using propositional logic for goal recognition and our formalisation of the problem allows us to use concept learning. This is very useful because the concept learning is well known and pretty easy to use, several algorithms already exist to treat concept learning. However, we made our own algorithm to control some generalisation bias and we have shown is efficiency in [Lorthioir et al., 2018]. We also showed that making some assumptions about the agent and his environment can drastically improve the goals deduction. This is why we plan to use LFIT or SMILE to infer the agents' models and the environment's rules to exploit such assumptions. This and the integration of the agents' goals preferences in our model will allow us to infer more refined hypotheses about the agent's goals. We were talking about the fact that in reality, we could not know when an agent reaches his goal. This is actually the main weakness of our method and we need to overcome this weakness. For this, a dynamic learning throughout the agent's actions might be efficient and more realistic. It could also be effective in a case where the agent changes goal along the way or in a case where the agent repeats the same cycle of actions, cases that we still can not solve with our method.

References

[Bourgne et al., 2007] Bourgne, G., El Fallah Segrouchni, A., and Soldano, H. (2007). Smile: Sound multi-agent incremental learning. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 164–171. ACM.

[Carberry, 2001] Carberry, S. (2001). Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(5):31–48.

[Cardona-Rivera and Michael Young, 2017] Cardona-Rivera, R. E. and Michael Young, R. (2017). Toward combining domain theory and recipes in plan recognition. In *The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition*, Palo Alto, California, USA.

[Charniak and Goldman, 1993] Charniak, E. and Goldman, R. P. (1993). A bayesian model of plan recognition. *Artificial Intelligence*, 64(5):53–79.

[E-Martin et al., 2015] E-Martin, Y., R-Moreno, M. D., and Smith, D. E. (2015). Practical goal recognition for iss crew activities. In *IJCAI the International Workshop on Planning and Scheduling for Space 2015 (IWPSS)*, Buenos Aires, Argentina.

[E-Martin and Smith, 2017] E-Martin, Y. and Smith, D. E. (2017). Goal recognition with noisy observations. In *The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition*, Palo Alto, California, USA.

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

[Goldman et al., 2018] Goldman, R. P., Friedman, S. E., and Rye, J. M. (2018). Plan recognition for network analysis: Preliminary report. In *The AAAI 2018 Workshop on Plan, Activity, and Intent Recognition*, New Orleans, Louisiana, USA.

[Ha et al., 2011] Ha, E., Rowe, J. P., Mott, B. W., and Lester, J. C. (2011). Goal recognition with markov logic networks for player-adaptive games. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 2011.*

[Hong, 2001] Hong, J. (2001). Goal recognition through goal graph analysis. *J. Artif. Int. Res.*, 15(1):1–30.

[Inoue et al., 2014] Inoue, K., Ribeiro, T., and Sakama, C. (2014). Learning from interpretation transition. *Machine Learning*, 94(1):51–79.

[Lang, 2004] Lang, J. (2004). A preference-based interpretation of other agents actions. In *International Conference on Automated Planning and Scheduling 2004*, volume 4, pages 33–42, Whistler. AAAI.

[Lorthioir et al., 2018] Lorthioir, G., Bourgne, G., and Inoue, K. (2018). Identifying goals of agents by learning from observations. In *MIWAI.*

[Mirsky et al., 2017] Mirsky, R., Stern, R., Gal, Y., and Kalech, M. (2017). Plan recognition design. In *The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition*, Palo Alto, California, USA.

[Schmidt et al., 1978] Schmidt, C. F., Sridharan, N. S., and Goodson, J. L. (1978). The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(5):45–83.

[Singla and Mooney, 2011] Singla, P. and Mooney, R. J. (2011). Abductive markov logic for plan recognition. In *AAAI Proceedings of the twenty-fifth national conference on Artificial intelligence 2011*, pages 1069–1075.

[Vered et al., 2018] Vered, M., Pereira, R. F., Magnaguagno, M. C., Meneguzzi, F., and Kaminka, G. A. (2018). Online goal recognition as reasoning over landmarks. In *The AAAI 2018 Workshop on Plan, Activity, and Intent Recognition*, New Orleans, Louisiana, USA.