Context-dependent Incremental Decision Making scrutinizing the Intentions of Others via Bayesian Network Model Construction

The Anh Han*† and Luís Moniz Pereira‡

Abstract

Decision making about which are the scrutinized intentions of others, usually called intention reading or intention recognition, is an elementary basic decision making process required as a basis for other higher-level decision making, such as the intention-based decision making which we have set forth in previous work. We present herein a recognition method possessing several features desirable of an elementary process: (i) The method is context-dependent and incremental, enabling progressive construction of a three-layer Bayesian network model as more actions are observed, and in a context-situated manner that relies on a logic programming knowledge base concerning the context; (ii) The Bayesian network is structured from a specific knowledge base of readily specified and readily maintained Bayesian network fragments with simple structures, thereby enabling the efficient acquisition of that knowledge base (engineered either by domain experts or else automatically from a plan corpus); and, (iii) The method addresses the issue of intention change and abandonment, and can appropriately resolve the issue of the recognition of multiple intentions. The several aspects of the method have been experimentally evaluated in applications and achieving definite success, using the Linux plan corpus and the so-called IPD plan corpora, which are playing sequences generated by game playing strategies needing to be recognized, in the iterated Prisoner's Dilemma. One other application concerns variations of Elder Care in the context of Ambient Intelligence.

Keywords: Intention Recognition, Bayesian Network Combination, Plan Corpora, Logic Programming, Evolutionary Game Theory, Elder Care.

1 Introduction

Decision about what another agent intends to do, or recognition of his/her intentions, is quite ubiquitous. Driving on the street, we often have to recognize intentions of other drivers for our safety. Do they want to turn right, turn left, or just go straight? Talking to friends, for a smooth conversation we usually need to recognize what they intend, as messages are not always very explicitly conveyed. To secure a successful collaboration with others, we recognize what they want and intend to do. One might risk one's life or the life of one's beloved if one cannot recognize the intentions of a hostile enemy.

We propose a method for making a decision about the intentions of others in a dynamic, real-world environment. An important aspect of intentions is *future-directedness*, i.e., if we intend something now, we mean to execute a course of actions to achieve something in the future [10, 73]. Most actions may be executed only at a far distance in time. During that period, the world is changing, and the initial intention may be changed to a more appropriate one or even abandoned [80, 11, 26]. To act appropriately, especially when beneficial to take into account intentions of others, as for our previously implemented intention-based decision making framework [32, 67, 43, 33, 41, 42], an intention recogni-

*Corresponding author; AI lab, Computer Science Department, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium; **Email: h.anh@ai.vub.ac.be; Phone: +32 483640581.**

†MLG, Département d'Informatique, Université Libre de Bruxelles, Boulevard du Triomphe CP212, 1050 Brussels, Belgium.

‡Centro de Inteligência Artificial (CENTRIA), Universidade Nova de Lisboa, 2829-516 Caparica, Portugal; **Email: lmp@fct.unl.pt**

tion method should take into account these changes, and, when necessary, be able to reevaluate the decision making model itself, depending on some time limit. In addition, as new actions are observed, the model should be reconfigurable to incorporate them. In other words, the model is incremental and, furthermore, the intention recognition prediction must be available at anytime [30].

Generally, *intention recognition* (also called *goal recognition*) is defined as the process of becoming aware of the intention of another agent and, more technically, as the problem of inferring an agent's intention through its actions and their effects on the environment [16, 83, 44, 2, 42]. *Plan recognition* is closely related to intention recognition, extending it to also recognize the plan the observed agent is following in order to achieve his intention [74, 2]. Mere intention recognition is performed in domains in which it is preferred to have a fast detection of just the user goal/intention rather than a more precise but time consuming detection of the user's complete plan, e.g., in the interface agents domain [2, 47, 55]. Like most other intention or plan recognition work, we assume that the actions to be used for the recognition task are given, and are observed with certainty. Usually, the actions are given by an *action* or *activity recognition* system (see, e.g. [86, 22]). Dealing with uncertainty in action recognition is beyond the scope of our work here [1]. Generally, the inputs to both intention and plan recognition systems are a set of conceivable intentions and a set of plans for achieving each intention, given in terms of a plan library [16, 26] or a plan corpus [7, 8, 3]). There are also generative approaches based on planning algorithms, which do not require plan library/corpus (e.g., see [70]).

The future-directedness of intentions also means that once an agent intends something, he has settled on a particular course of action [10, 17, 80, 73]. This makes the intentions relatively stable, pending new information. An agent who made the decision to act in a certain way commits to sticking to this decision till there are reasons to trigger further deliberations [10, 80]. In other words, intentions are relatively resistant to reconsideration unless there are pondered reasons to do so [11, 10, 80, 73, 42]. Following this, any attempt to tackle the issues of intention change or abandonment cannot be solely based on observable actions. The reasons why the intention is changed or abandoned must be taken into account. There can be changes in the environment—possibly enacted by other agents—that impel the observed agent to refrain from following his initial intentions. Agents may also imitate or adopt intentions of other more successful agents—a form of social learning usually implemented in the context of Evolutionary Game Theory (see discussion later). And here the context-dependent modeling appears to be unavoidable, in order to provide more accurate decision about others' intentions (Section 7).

In this work, we resort to Bayesian Networks (BNs) as the intention recognition model. The flexibility of BNs for representing probabilistic dependencies and the efficiency of inference methods for BNs have made them an extremely powerful and natural tool for problem solving under uncertainty [62, 63]. To perform intention recognition, we construct a three-layer BN [65, 67, 41]—justified based on Heinze's causal intentional model [44, 83]—and use it for evidential reasoning from observations to intention hypothesis.

We surmise a knowledge representation method to support incremental BN model construction for performing intention recognition during runtime, from an initially given domain knowledge base. As more actions are observed, a new BN is constructed from the previous one reinforcing some intentions whilst ruling out others. This incremental method allows domain experts to specify knowledge in terms of small and simple BN fragments, which can be easily maintained and changed, and which are used to compose the situated ongoing BN model. Alternatively, these fragments can be easily learned from data. We also propose a method to represent relationships among intentions, when considering the case of agents that may pursue multiple intentions simultaneously (Section 5). It is an indispensable aspect, but mostly omitted in prior work, which moreover allows us to sometimes significantly decrease the complexity of the BN inference [28].

[1] Though in our paper "Moral Reasoning with Uncertainty" [40], we address the case of judging under uncertainty of actual actions performed, and the dispensing of a verdict by a jury, illustrated with an example.

Our method is generally inspired in that knowledge experts often consider a related set of variables together, and organize domain knowledge in larger chunks. An ability to represent conceptually meaningful groupings of variables and their interrelationships facilitates both knowledge elicitation and knowledge base maintenance [52]. To this end, there have been several methods proposed for Bayesian Network construction from small and easily maintained network fragments [62, 68, 56, 52, 89, 59, 51]. In essence, a combination of BNs is a graph that includes all nodes and links of the networks, where nodes with the same name are combined into a common node. The main issue for a combination method is how the influence of different parents of the common node can be combined in the new network, given the partial influence of each parent in the corresponding fragment. The most extensively used and popular combination method is Noisy-Or, firstly proposed by [62] for BNs of Boolean variables, and generalized by [82, 21] for the general case of arbitrary domains. The Noisy-OR method is discussed in Section 4.

The rest of this article has the following structure. The next section describes a short review of prior work, pointing out those limitations of that work which we address in our method (Section 2). Section 3 recalls some background of BNs that is necessary for further discussion of the intention recognition method, which is described in detail in Section 4. Section 5 describes a method for expressing relationships amongst intention variables in a BN model for intention recognition. Section 6 discusses terminologies used for the evaluation of our method, including the evaluation metrics, and the first set of experimental results on the Linux plan corpus. Section 7 presents our own, so-called IPD plan corpora benchmarks based on the iterated Prisoner's Dilemma, and show our experimental results for it. We also describe how to incorporate contextual information in our model, and how it helps to improve the intention recognition performance. Section 8 proffers some extensions of our method to take into account contextual information. Further developments and concluding remarks, in Section 9, end the article.

2 Related Work

Work on intention and plan recognition has been paid much of attention for more than thirty years, and a large number of methods have been applied. They can be roughly categorized into two main groups: *Consistency* and *Probabilistic* approaches [2, 81, 26, 74].

Consistency approaches face the problem by determining which intention is consistent with the observed actions, i.e. whether the observed actions match with at least a plan achieving the intention. The earliest work on plan recognition belongs to this group [76, 88, 49, 46]. More recent work can be found in a rather comprehensive survey by [74]. The problem with the consistency approaches is that they cannot handle well the case where the current observed actions enable more than one intention—they cannot directly select between those intentions.

Probabilistic approaches, on the other hand, are mainly based on Bayesian network and (Hidden) Markov models [16, 69, 24, 1, 24, 18, 1, 14, 48, 83, 77, 26, 65, 67, 3]. A significant advantage of the probabilistic approaches is that they can directly address the above issue of the consistency approaches—by finding the most probable intentions given the current observations, on the basis of accumulated statistical evidence or simply subjective beliefs encoded in a Bayesian network or Markov model.

Bayesian approaches have exhibited the most successful models applied to intention/plan recognition [16, 69, 29, 27, 26]. The first model was built by [15, 16]. Depending on the structure of plan libraries, a knowledge-based model construction is employed to build BNs from the library—which is then used to infer the posterior probability of explanations (for the set of observed actions). This approach, mostly advanced by [29] and especially in the more recent work [26] [2], addresses a number of issues in intention/plan recognition, e.g., when the observed agent follows multiple intentions or interleaved plans simultaneously; fails to observe actions; addresses partially ordered plans. However, there are some important aspects not yet explored therein, partially for the

[2] Note that this work is based on Bayesian inference, though they do not build Bayesian networks as in [15, 16].

sake of computational efficiency. First, prior probabilities of intentions are assumed to be fixed. This assumption is not always reasonable because those prior probabilities should in general depend on the situation at hand [11, 10, 69, 12], and can justifiably be captured by causes/reasons of the intentions, as in our method [67, 34, 36, 83, 44]. Indeed, [26] also highlighted the need to account for contextual information or state of the world as a potential extension to their plan recognizer. In [69], a similar context-dependent Bayesian approach is used, though the model therein is not incremental. The authors demonstrated that taking into account contextual information is important to appropriately recognize drivers' intention in the traffic monitoring domain [69].

Second, intentions are assumed to be independent of each other. This is not generally the case since the intentions may support or exclude one another, leading to the need to reconfigure the model. Those works hence might not appropriately address multiple intentions recognition. [69] proposed to combine, in their BN model for plan recognition, the mutually exclusive plan nodes into a single variable. As a step further, we formally define how that can be done appropriately, so as to guarantee the consistency in the obtained BN (Section 5). This latter assumption must always, explicitly or implicitly, be made by the approaches based on (Hidden) Markov models, e.g. [3, 13], or statistical corpus-based machine learning [7, 8]. Generally, in those approaches, a separate model is built for each intention; thus no relations amongst the intentions are expressed or can be expressed. These works were restricted to the single intention case. The method in this article attempts to tackle the multiple case more appropriately.

Different from most above mentioned works, our model is *context-dependent*, which is achieved by including in it causes/reasons of intentions. This way, our model can appropriately deal with the abandonment/changes of intentions—when the causes/reasons do not support or force the intending agent to hold those intentions anymore—in an *integrated* manner. In contrast, in [25], the authors build a *separate* model to recognize when the observed agent abandons its current intention, which may then trigger revision of the intention recognition model. To the best of our knowledge, this is the only work addressing the abandonment issue. However, the system therein is only evaluated with a rather small benchmark (with three intentions), and only for the accuracy of the abandonment recognition itself. The benefit from having this additional intention abandonment recognition module for enhancing intention/plan recognition performance has not been studied, as the authors themselves mention in their recent study [26]. We address this issue in Section 7.

3 Bayesian Networks

Definition 1 *A Bayesian Network (BN) is a pair consisting of a directed acyclic graph (DAG) whose nodes represent variables and missing edges encode conditional independencies between the variables, and an associated probability distribution satisfying the Markov assumption of conditional independence, saying that variables are independent of non-descendants given their parents in the graph [62, 63].*

In a BN, associated with each node of its DAG is a specification of the distribution of its variable, say A, conditioned on its parents in the graph (denoted by $pa(A)$)—i.e., $P(A|pa(A))$ is specified. If $pa(A) = \emptyset$ (A is called *root* node), its unconditional probability distribution, $P(A)$, is specified. These distributions are called Conditional Probability Distribution (CPD) of the BN.

The joint distribution of all node values can be determined as the product of conditional probabilities of the value of each node on its parents

$$P(X_1, \ldots, X_N) = \prod_{i=1}^{N} P(X_i|pa(X_i)) \tag{1}$$

where $V = \{X_i|1 \leq i \leq N\}$ is the set of nodes of the DAG.

Suppose there is a set of evidence nodes (i.e. their values are observed) in the DAG, say $O = \{O_1, \ldots, O_m\} \subset V$. We can determine the conditional probability distribution of a variable X given the observed value of evidence nodes by using the

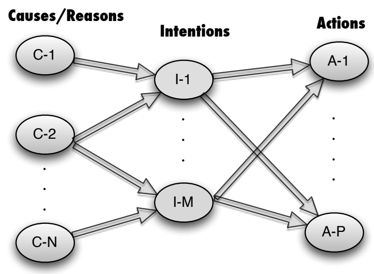


Figure 1: General structure of a Bayesian network for intention recognition. The Bayesian network consists of three layers. The pre-intentional layer consists of cause/reason nodes, connecting to intention nodes in the intentional layer, which in turn connect to action nodes in the activity layer.

conditional probability formula

$$P(X|O) = \frac{P(X,O)}{P(O)} = \frac{P(X,O_1,\ldots,O_m)}{P(O_1,\ldots,O_m)} \tag{2}$$

where the numerator and denominator are computed by summing the joint probabilities over all absent variables with respect to V as follows

$$\begin{aligned} P(X=x,O=o) &= \sum_{av \in ASG(AV_1)} P(X=x,O=o,AV_1=av) \\ P(O=o) &= \sum_{av \in ASG(AV_2)} P(O=o,AV_2=av) \end{aligned} \tag{3}$$

where $o = \{o_1, ..., o_m\}$ with $o_1, ..., o_m$ being the observed values of $O_1, ..., O_m$, respectively; $ASG(Vt)$ denotes the set of all assignments of vector Vt (with components are variables in V); AV_1, AV_2 are vectors components of which are corresponding absent variables, i.e. variables in $V \setminus (O \cup \{X\})$ and $V \setminus O$, respectively.

4 Incremental Intention Recognition

In [65, 67], a general BN model for intention recognition is presented and justified based on Heinze's causal intentional model [44, 83]. Basically, the BN consists of three layers: cause/reason nodes in the first layer (called *pre-intentional*), connecting to intention nodes in the second one (called *intentional*), in turn connecting to action nodes in the third (called *activity*) (Figure 1).

In general, it is possible to build a single BN network containing all the relevant factors. But in a real application domain, it can be envisaged that such a network is very large, which clearly leads to high complexity for the BN inference [63, 62]. To address this problem, in this work we present a method for incrementally constructing a BN model with the purpose of performing *incremental* intention recognition. The idea is that, given the current observations, only their relevant factors are incorporated for the moment into the network.

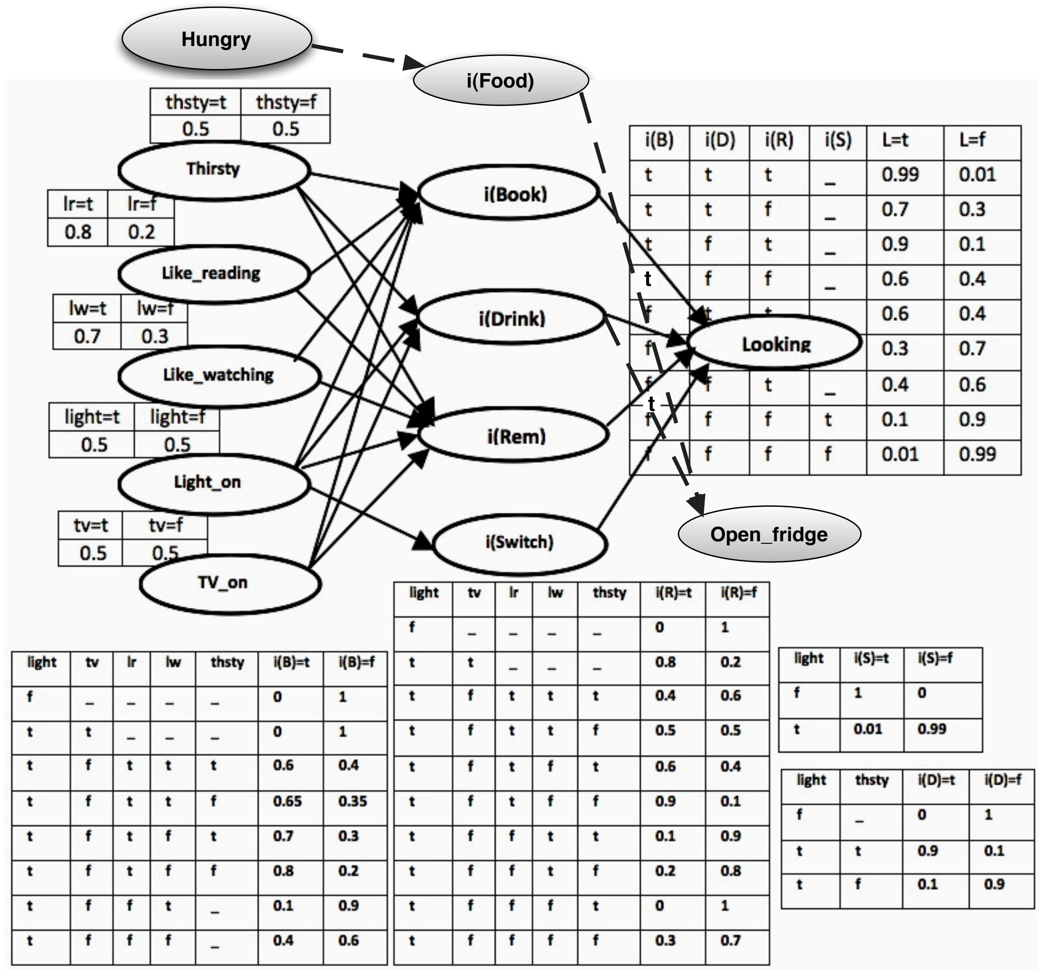
Definition 2 (Intention Recognition BN – IRBN) *A BN for intention recognition (IRBN) W is a triple* $\langle \{Cs, Is, As\}, pa, P_W \rangle$ *where*

- Cs, Is *and* As *are the sets of cause/reason nodes, intention nodes and action nodes, respectively. They stand for binary random variables (i.e. their value is either* true (T) *or* false (F)*).*
- pa *is a mapping which maps a node to the set of its parent nodes such that:* $pa(C) = \emptyset \ \ \forall C \in Cs$*;* $pa(I) \subseteq Cs \ \ \forall I \in Is$*; and* $\emptyset \neq pa(A) \subseteq Is \ \forall A \in As$*.*
- *CPD tables are given by the probability distribution* P_W*, i.e.* $P_W(X|pa(X))$ *defines the probability of* X *conditional on* $pa(X)$ *in* W*,* $\forall X \in Cs \cup Is \cup As$*.*

Note that the set of cause/reason nodes of an IRBN, Cs, can be empty, as in the case of the Linux plan corpus we shall see later on (Section 6).

For illustration of the concepts and the intention recognition method presented in this section, we consider an extended example from the Elder Care domain [66, 67].

Example 1 (Elder's Care) *An elder stays alone in his apartment. The assistant system (with the capability of intention recognition) observes that he is looking for something in the living room. In order to assist him, the system needs to figure out what he*



thsty=t	thsty=f
0.5	0.5

lr=t	lr=f
0.8	0.2

lw=t	lw=f
0.7	0.3

light=t	light=f
0.5	0.5

tv=t	tv=f
0.5	0.5

i(B)	i(D)	i(R)	i(S)	L=t	L=f
t	t	t	_	0.99	0.01
t	t	f	_	0.7	0.3
t	f	t	_	0.9	0.1
t	f	f	_	0.6	0.4
f	t	t	[illegible]	0.6	0.4
f	[illegible]	[illegible]	[illegible]	0.3	0.7
f	f	t	_	0.4	0.6
f	f	f	t	0.1	0.9
f	f	f	f	0.01	0.99

light	tv	lr	lw	thsty	i(B)=t	i(B)=f
f	_	_	_	_	0	1
t	t	_	_	_	0	1
t	f	t	t	t	0.6	0.4
t	f	t	t	f	0.65	0.35
t	f	t	f	t	0.7	0.3
t	f	t	f	f	0.8	0.2
t	f	f	t	_	0.1	0.9
t	f	f	f	_	0.4	0.6

light	tv	lr	lw	thsty	i(R)=t	i(R)=f
f	_	_	_	_	0	1
t	t	_	_	_	0.8	0.2
t	f	t	t	t	0.4	0.6
t	f	t	t	f	0.5	0.5
t	f	t	f	t	0.6	0.4
t	f	t	f	f	0.9	0.1
t	f	f	t	t	0.1	0.9
t	f	f	t	f	0.2	0.8
t	f	f	f	t	0	1
t	f	f	f	f	0.3	0.7

light	i(S)=t	i(S)=f
f	1	0
t	0.01	0.99

light	thsty	i(D)=t	i(D)=f
f	_	0	1
t	t	0.9	0.1
t	f	0.1	0.9

Figure 2: Elder Intentions Recognition IRBN

intends to find. The possible things are: something to read (Book – i(B)); something to drink (Drink – i(D)); the TV remote control (Rem – i(R)); and the light switch (Switch – i(S)). The IRBN representing this scenario is that of Figure 2, but without the grey-filled nodes for now.

There are five nodes in the first layer: Thirsty (thsty), Like_reading (lr), Like_watching (lw), Light_on (light) and TV_on (tv). We have, $Cs = \{thsty, lr, lw, light, tv\}$. *Intention nodes in the middle layer are,* $Is = \{i(B), i(D), i(R), i(S)\}$. *Action nodes are,* $As = \{Looking\}$.

The mapping pa *is defined by the arrows in the IRBN, e.g.,* $pa(i(B)) = \{thsty, lr, lw, light, tv\}$ *and* $pa(Looking) = \{i(B), i(D), i(R), i(S)\}$.

The intention recognition method will be performed by incrementally constructing an IRBN as more actions are observed. The construction is based on a prior knowledge base consisting of unit fragments of BN (Figure 3).

Definition 3 (Unit Fragments) *There are just two types of unit fragments issued for IRBN model construction (Figure 3):*

1. *A unit fragment* for an action A *consists of an intention* I *connecting to (i.e. causally affecting)* A*, and is denoted by* $UF_{\mathfrak{A}}(I, A)$.

2. *A unit fragment* for an intention I *consists of a context-independent and fixed over time set of causes/reasons* Cs *connecting to (i.e. causally affecting)* I*, and is denoted by* $UF_{\mathfrak{I}}(Cs, I)$.

Note that a unit fragment for an action contains a single action, which thus enables to easily handle the set of conceivable intentions in the IRBN, in a context-dependent manner, as implemented with operators described in the sequel. In contradistinction, a unit fragment for an intention includes a fixed set of causes/reasons rather than a single one. This simplified modification can be made because the prior probability of each cause/reason node can be updated in a context-dependent manner (as we shall see in Section 8), allowing us to easily switch on and off the effect of a cause/reason node.

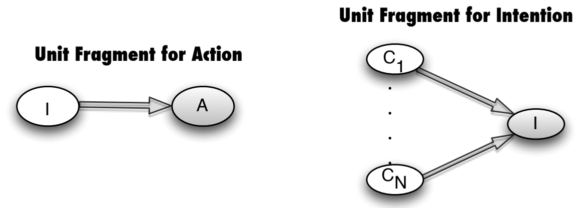


Figure 3: Two Types Unit Fragments.

Example 2 (Unit Fragments) *Here are some unit fragments for actions and intentions that will be used later for constructing the IRBN in Figure 2.*

Some unit fragments for the action Looking: $UF_{\mathfrak{A}}(i(B), L)$, $UF_{\mathfrak{A}}(i(D), L)$, $UF_{\mathfrak{A}}(i(R), L)$ *and* $UF_{\mathfrak{A}}(i(S), L)$.

Some unit fragments for intentions:

- $UF_{\mathfrak{I}}(\{thsty, lr, lw, light, tv\}, i(B))$,
- $UF_{\mathfrak{I}}(\{thsty, light, tv\}, i(D))$,
- $UF_{\mathfrak{I}}(\{thsty, lr, lw, light, tv\}, i(R))$ *and*
- $UF_{\mathfrak{I}}(\{light\}, i(S))$.

We next stipulate some conditions to guarantee the consistency of the knowledge base.

Definition 4 (Knowledge Base) *The domain knowledge base* KB *consists of a set of actions* Δ*, a set of intentions* Υ*, a set of unit fragments for each action in* Δ *and a single unit fragment for each intention in* Υ*, satisfying that*

- *An intention* I *has a unique unit fragment in KB. The set of its parents (causes/reasons) and the CPD table associated with it are fixed. Let* $\mathfrak{C}(I)$ *denote the set of the parents of* I *and* $P_{KB}(I|\mathfrak{C}(I))$ *define its CPD table.*

- *A cause* C *has the same prior probability distribution in all the unit fragments (for intentions) that it belongs to, denoted by* $P_{KB}(C)$.

The simple structures of unit fragments enable domain experts to easily construct and maintain the knowledge base. The fragments can also be learnt from appropriate datasets, as we shall see later for the Linux and IPD plan corpora.

Before presenting the intention recognition algorithm, let us define some (original) operators for handling CPD tables and IRBNs.

4.1 Operators for constructing IRBNs

As a new action A is observed, we need to incorporate it into the current IRBN. Firstly, the appropriate unit fragments for A are selected from the prior domain knowledge base. In Section 8.2, we will discuss methods for selecting the appropriate fragments in a situation-sensitive manner. They are based on the intuition that whether an intention may give rise to an action depends on the situation in which the action is observed. That enables to reduce the size of the BN model, which otherwise could be very large.

For now, let us assume that the operator *select(A,SIT)* provides the (context-dependent) set of unit fragments for action A given the situation at hand, *SIT*. If *SIT* is empty, *select(A,SIT)* is the set of all unit fragments for action A from the knowledge base.

Then, after having obtained the appropriate fragments, we combine them using the Noisy-OR method [62, 82, 19], thereby obtaining a BN with a single action (Figure 4). We then add into it appropriate causes/reasons for each intention.

Definition 5 (Unit IRBN via Noisy-OR) *The unit IRBN W for action* A *in a given situation SIT is an IRBN with a single action, denoted by $irBN(A) = \langle\{Cs, Is, \{A\}\}, pa, P_W\rangle$. It is obtained via the Noisy-OR method as follows.*

Let $select(A, SIT) = \{UF_{\mathfrak{A}}(I_1, A), \ldots, UF_{\mathfrak{A}}(I_N, A)\}$ and for $1 \le i \le N$, $P(A = T|I_i = T) = q_i$ (defined in fragment $UF_{\mathfrak{A}}(I_i, A)$). Then,

- $Is = \{I_1, \ldots, I_N\};\ \ Cs = \bigcup_{I \in Is} \mathfrak{C}(I);$
- $pa(I) = \mathfrak{C}(I)\ \ \forall I \in Is;\quad pa(A) = Is;$
- $P_W(C) = P_{KB}(C)\ \ \forall C \in Cs;\ \ P_W(I|pa(I)) = P_{KB}(I|\mathfrak{C}(I))\ \ \forall I \in Is;$ *and, according to the* Noisy-OR *method*

$$P_W(A = T|pa(A)) = 1 - \prod_{1 \le i \le N : I_i = T} (1 - q_i). \quad (4)$$

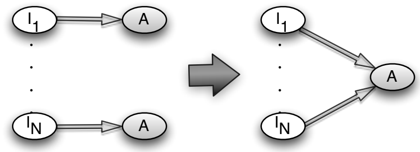


Figure 4: Noisy-OR Combination Method [62, 82, 21]: $P(A = T|I_1, \ldots, I_N) = 1 - \prod_{i:I_i=T}(1 - q_i)$ where $P(A = T|I_i = T) = q_i$.

The rationale and appropriateness of the application of the Noisy-OR method here for combining unit fragments for an action is based on the intuition that each intention can be interpreted as "*cause*" of the observed action [10, 69]; and the action occurs when one or more of such intentions are active. More detailed arguments for this can be found in [19, 62].

Example 3 (Noisy-OR) *Consider the action node Looking (i.e., L), with four parent intention nodes i(B), i(D), i(R), and i(S). In the unit fragments for* L, *$UF_{\mathfrak{A}}(i(B), L)$, $UF_{\mathfrak{A}}(i(D), L)$, $UF_{\mathfrak{A}}(i(R), L)$, and $UF_{\mathfrak{A}}(i(S), L)$, we define $P(L = T|i(B) = T) = 0.9$, $P(L = T|i(D) = T) = 0.7$, $P(L = T|i(R) = T) = 0.8$, $P(L = T|i(S) = T) = 0.2$, respectively.*

The combination of these unit fragments using the Noisy-OR method, in Figure 5, and the CPD table for the node L *in the obtained BN is defined following Eq. (4). The CPD for the node Looking (L) in Figure 2, now instead of being fully constructed beforehand, can be defined by this Noisy-OR combination from the simple unit fragments.*

Obviously, from the design point of view, it is easier and usually much cheaper to construct the small fragments (and then combine them) than to construct the whole BN [62, 51] (see also the introduction section, Section 1).

Now we need to combine the obtained unit IRBN, `irBN(A)`, with the current IRBN. For that, in the sequel we define how to combine two IRBNs. Intuitively, we simply add up all the new nodes and links of the new IRBN to the current IRBN, keeping the CPD tables from the two original IRBNs.

Definition 6 (Combination of IRBNs) *Let $W_1 = \langle\{Cs_1, Is_1, As_1\}, pa_1, P_1\rangle$ and*

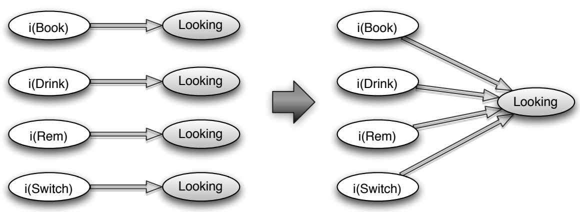


Figure 5: Noisy-OR Combination for the node *Looking*.

$W_2 = \langle\{Cs_2, Is_2, As_2\}, pa_2, P_2\rangle$ *be two IRBNs, such that* $As_1 \cap As_2 = \emptyset$ *(any actions in* As_2 *which are already so named in* As_1 *are renamed). The combination of these two IRBNs is an IRBN, denoted by* $\mathtt{comb}(\mathtt{W}_1, \mathtt{W}_2) = \langle\{Cs, Is, As\}, pa, P_W\rangle$*, where*

- $As = As_1 \cup As_2$; $Is = Is_1 \cup Is_2$; $Cs = Cs_1 \cup Cs_2$;
- $pa(I) = \mathfrak{C}(I) \quad \forall I \in Is$; $\quad pa(A) = pa_1(A) \cup pa_2(A) \quad \forall A \in As$;
- $P_W(C) = P_{KB}(C)$ $\forall C \in Cs$; $P_W(I|pa(I)) = P_{KB}(I|\mathfrak{C}(I)) \quad \forall I \in Is$; $P_W(A|pa(A)) = P_k(A|pa_k(A))$ *if* $A \in As_k$ *(with* $k = 1, 2$*).*

Note that here it is allowed the possibility that the observed agent follows multiple intentions simultaneously.

When some intentions are found irrelevant, e.g., because they are much unlikely [3], those intentions should be removed from the IRBN. These intentions might resurface in the IRBN later if there appear newly observed actions triggering them. This could happen frequently when one considers the case where agents might abandon/change their initial intentions—especially for the multiple intention recognition case.

The removal is enacted by considering the intentions (to be removed) as completely false and employing a *project* operator, described below.

Definition 7 (Project of CPD Table) *Let Tb be a CPD table defining* $P(X|V)$*, the probability of a random variable X conditional on a set of random* binary *variables V. Considering a strict subset* $V' \subsetneq V$*, the project of Tb on* V'*, denoted by* $\mathtt{proj}(\mathtt{Tb}, \mathtt{V}')$*, is the part of Tb corresponding to all variables in* $V \setminus V'$ *being* false.

Note that this approach guarantees the consistency of the probability distribution over the set of the remaining intentions. It requires neither normalization over the set nor creating a new CPD: any computation can be done directly with the original CPD.

Definition 8 (Remove Intentions from IRBN) *Let* $W = \langle\{Cs, Is, As\}, pa, P_W\rangle$ *be an IRBN and* $R \subset Is$ *be a strict subset of Is. The result of removing the set of intentions R from W is an IRBN, denoted by* $\mathtt{remove}(\mathtt{W}, \mathtt{R}) = \langle\{Cs_R, Is_R, As_R\}, pa_R, P_R\rangle$*, where*

- $As_R = As$; $Is_R = Is \setminus R$; $Cs_R = \bigcup_{I \in Is_R} \mathfrak{C}(I)$;
- $pa_R(I) = \mathfrak{C}(I) \quad \forall I \in Is_R$; $\quad pa_R(A) = pa(A) \setminus R \quad \forall A \in As_R$;
- $P_R(C) = P_{KB}(C) \quad \forall C \in Cs_R$; $P_R(I|pa_R(I)) = P_{KB}(I|\mathfrak{C}(I)) \quad \forall I \in Is_R$; *and for each* $A \in As_R$, $P_R(A|pa_R(A))$ *is defined by the CPD table* $proj(Tb, pa_R(A))$ *where Tb is the CPD table for A in W, i.e. defined by* $P_W(A|pa(A))$.

Based on these operators, we now describe an algorithm for incremental intention recognition in a real-time manner.

Algorithm 1 (Incremental Intention Recognition) *Repeat the following steps until some given time limit is reached. The most likely intention in the previous cycle of the repeat loop is the final result* [4].

1. *Let A be a new observed action. Combine the current IRBN W with* $\mathtt{irBN}(\mathtt{A})$ *to obtain* $\mathtt{W}' = \mathtt{comb}(\mathtt{W}, \mathtt{irBN}(\mathtt{A}))$*. If A is the initially observed action, let* $\mathtt{W}' = \mathtt{irBN}(\mathtt{A})$.

[3] One intention is much less likely than the other if the fraction of its likelihood and that of the most likely intention is less than some small threshold. It is up to the KB designer to provide it.

[4] Obviously, one can easily modify the algorithm to find the set of N most likely intentions. In the next section we shall see that in more detail.

2. *Compute the probability of each intention in W', conditional on the set of current observations in W'. Remove the intentions which are much less likely than the others (following Definition 8).*

Example 4 (Elder's Incremental IR) *When observing the action* Looking*, the IRBN in Figure 2 (without action node* Open_Fridge*) is created. We compute the probabilities* [5] V_1, V_2, V_3, V_4 *of each intention i(B), i(D), i(R), and i(S), respectively, conditional on the observations, including action node* Looking *and the two cause/reason nodes,* tv_on *and* light_on*.*

Let us consider the possible cases.

- If light is off, then $V_1 = V_2 = V_3 = 0$ and $V_4 = 1$, regardless of the observed state of the TV.
- If light is on and tv is off, then $V_1 = 0.71$, $V_2 = 0.49$, $V_3 = 0.50$, and $V_4 = 0.011$.
- If light is on and tv is on, then $V_1 = 0$, $V_2 = 0.50$, $V_3 = 0.80$, and $V_4 = 0.01$.

Hence, if one observes that the light is off, the elder is definitely looking for the light switch, given that he is looking around. Otherwise, if one observes the light is on, and that the TV is on too, the intention of looking for the remote control, *i(R)*, is most probable; but, if the TV is off, then looking for something to read, *i(B)*, is most probable. They are the output of the algorithm whenever the decision needs to be made immediately after observing the first action.

Now suppose we are in the second case (light is on and tv is off), and the decision has not had to be made. Another action, open the fridge – *Open_fridge (OpenF)*, is observed. Suppose there are two unit fragments for this action, $UF_{\mathfrak{A}}(i(D), \mathit{OpenF})$, and $UF_{\mathfrak{A}}(i(F), \mathit{OpenF})$, and one unit fragment for the intention *i(F)*, $UF_{\mathfrak{I}}(\{\mathit{Hungry}\}, i(F))$, in the knowledge base (where *i(F)* stands for the intention of looking for some food). Respectively, CPD tables in these unit fragments are given by, $P(\mathit{OpenF} = T | i(D) = T) = 0.3$, $P(\mathit{OpenF} = T | i(F) = T) = 0.8$, and $P(i(F) = T | \mathit{Hungry} = T) = 0.9$. A new IRBN with the new action node is created (Figure 2 in full, with all grey-filled nodes). Note that now the intention node *i(Switch)*, and with all its in- and out-connections, is being removed, since it is very unlikely compared with others (see above, $V_4 = 0.011$).

Now the conditional probabilities of intentions in the IRBN are: $V_1 = 0.57$, $V_2 = 0.55$, $V_3 = 0.47$, and $V_5 = 0.63$ (V_5 is that of the intention *i(Food)*). Looking for food becomes the most probable intention.

[5] In this work, for Bayesian network reasoning and inference, we utilize the Bayesian reasoning engine SMILE running in a MAC-OS-X computer, publicly available at: http://genie.sis.pitt.edu/.

5 Relation Among Intentions

When considering the case in which the observed agent may pursue multiple intentions simultaneously, it is undoubtedly indispensable to take into account and express the relations amongst the intentions in the model. Pursuing one intention may exclude some other intention to be pursued [69, 10, 73]. It may be so because of some resource limitation, e.g., allowance time is not enough for accomplishing both intentions at the same time [50]. It also may be because of the nature or restriction of the observed agent's task: the agent is restricted to pursuing a single intention (e.g., in constructing Linux and Unix plan corpora, a user is given one task at a time to complete) [7, 8, 53].

We introduce a so-called *exclusive relation* e—a binary relation on the set of intention nodes—representing that if one intention is pursued, then the other intention cannot be pursued. It is usually, although perhaps not always, the case that intentions exclusiveness is symmetric. It holds for the resource limitation case: one intention excludes the other intention because there is not enough resource for accomplishing both, which in turn implies that the latter intention excludes the former one too. It also clearly holds for the case where the agent is restricted to pursuing a single intention. Here we assume that e is symmetric; it can be renamed *mutually exclusive relation.*

Intentions I_1 and I_2 are mutually exclusive iff they cannot be pursued simultaneously, i.e. $P(I_1 = T, I_2 = T) = 0$. Thus, for any action A, if $I_1, I_2 \in pa(A)$ then the CPD table for A is undefined. Hence, the BN needs to be restructured. The mutually exclusive intentions must be combined into a single intention node since they cannot co-exist as parents of a node. Each intention represents a possible value of the new combined node. Namely, let $I_1, \ldots, I_t$ be such that $e(I_i, I_j),\ \forall i, j : 1 \leq i < j \leq t$. The new combined node, I, stands for a random variable whose possible outcomes are either $I_i,\ 1 \leq i \leq t$, or $\tilde{I}$—the outcome corresponding to the state that none of the $I_i = T$. Note that if the intentions are exhaustive, $\tilde{I}$ can be omitted. Next, I is linked to all the action nodes that have a link from one of $I_i,\ 1 \leq i \leq t$.

There remains to re-define CPD tables in the new BN. They are kept the same for action A where $I \notin pa(A)$. For A such that $I \in pa(A)$, the new CPD table at $I = I_i$ corresponds to the CPD table in the original BN at $I_i = T$ and $I_j = F\ \forall j \neq i$, i.e. $P(A|I = I_i, \ldots) = P(A|I_0 = F, \ldots, I_{i-1} = F, \mathbf{I_i} = \mathbf{T}, I_{i+1} = F, \ldots, I_t = F, \ldots)$. Note that the left hand side is defined in the new BN, and the right hand side is defined in the original BN. Similarly, the new CPD table at $I = \tilde{I}$ corresponds to $I_i = F$ for all $1 \leq i \leq t$. In addition, prior probability $P(I = I_i) = P(I_i = T)$ and $P(I = \tilde{I}) = \prod_{i=1}^{t} P(I_i = F)$ (and then being normalized).

We now specify the CPD table of I. In the new BN, the causes/reasons of each intention are connected to the combined node, i.e. $pa(I) = \bigcup_{i=1}^{t} \mathfrak{C}(I_i)$. Applying the Markov assumption (see Def. 1) we have $P(I = I_i|pa(I)) = P_i(I_i = T|\mathfrak{C}(I_i))$ and $P(I = \tilde{I}|pa(I)) = \prod_{i=1}^{t} P_i(I_i = F|\mathfrak{C}(I_i))$, where P_i is the probability distribution of the unit fragment for I_i.

In the next section we focus on the single intention recognition case, showing how the approach to representing relationships amongst several intentions can significantly decrease the complexity of the probability inference therein. We then present experimental results on the Linux plan corpus. After that, in Section 7, we provide further experimentation on our novel so-called IPD plan corpora.

6 Single Intention Recognition

6.1 The Model

Suppose the observed agent pursues a single intention at a time. In this case, all intentions are mutually exclusive, and they can be combined into a single node. The IRBN then has a single intention node, linking to all action nodes. All cause/reason nodes are connected to the intention node.

Let $I_1, \ldots, I_n$ be the intentions in the original IRBN. As usual, they are assumed to be exhaustive, i.e. the observed agent is assigned an intention from them. The combined node I thus has n possible outcomes $I_i,\ 1 \leq i \leq n$. Let $As = \{A_1, \ldots, A_m\}$ be the set of current observed actions, all linked to the single intention I. The set of all cause/reason nodes are $Cs = \cup_{i=1}^{n} \mathfrak{C}(I_i)$. Suppose $C_e \subseteq Cs$ is the set of cause/reason nodes which are observed (evidence nodes). For instance, in the Elder Care examples presented above, the state of TV and the state of the light are observed cause/reason nodes.

Let $C_{ne} = Cs \setminus C_e$. Applying Eq. (2), we obtain the probability of each intention conditional on the current observations

$$P(I = I_j|C_e, As) = \frac{P(I_j, C_e, As)}{\sum_{i=1}^{n} P(I_i, C_e, As)}\ \forall 1 \leq j \leq n \tag{5}$$

where, by applying the joint probability formulas (1) and (3), we obtain for all j that

$$P(I_j, C_e, As) = \prod_{i=1}^{m} P(A_i|I_j) \left(\sum_{C_{ne}} P(I_j|Cs) \prod_{C \in Cs} P(C) \right). \tag{6}$$

This implies that, when not including causes/reasons for intentions (i.e., $Cs = \emptyset$) as in the case of Linux plan corpus below, our intention recognizer has a linear complexity on the number of intentions $O(|n|)$.

If no cause/reason nodes are observed, i.e. $C_{ne} = Cs$ (as in the case of the Linux plan we examine in the next subsection), we obtain

$$P(I_j, C_e, As) = P(I_j) \prod_{i=1}^{m} P(A_i|I_j). \tag{7}$$

If all of them are observed, i.e. $C_{ne} = \emptyset$ (as we shall see in the IPD Plan corpora), the term $\prod_{C \in Cs} P(C)$ is simplified in the fraction of Eq. (5), since it appears and is the same in both numerator and denominator. Thus, in these two cases, we do not need to define prior probabilities distribution of the root nodes in Cs, as to be applied to obtain experimental results for the Linux and IPD plan corpora in the following sections. Note that in the latter case we still need to compute the conditional probabilities $P(I_j|Cs)$.

6.2 Experiments with Linux Plan Corpus

6.2.1 The Linux Plan Corpus

Plan corpus is the term used to describe a set of plan sessions and consists of a list of goals/intentions and the actions a user executed to achieve them [3]. Although there are many corpora available for testing machine learning algorithms in other domains, just a few are available for training and testing plan/intention recognizers; furthermore, each of the plan/intention recognizers using plan corpora usually has its own datasets—which leads to a difficult comparison amongst each other. For that important reason, we chose Linux plan corpus [8]—one of the rare regularly used plan corpora—which was kindly made publicly available by Nate Blaylock—in order to test our system. It also enables a better comparison with other systems using this corpus [9, 8, 3, 81].

The Linux plan corpus is modeled after Lesh's Unix plan corpus [53]. It was gathered from 56 human users (graduate and undergraduate students, faculty, and staff) from the University of Rochester Department of Computer Science. The users have different levels of expertise in the use of Linux, and they were allowed to perform as many times as they wished, in order to contribute more plan sessions. The sessions, consisting in sequences of commands performed by the users to achieve a given goal/intention, were automatically recorded. For example, a goal is to find a file with a given name or copy some files to a given folder, and the users can use the Linux commands such as "find", "cp", "cd", "ls", etc. At the end of each session, the users were asked to indicate whether they succeeded in achieving their goal/intention. In total, there are 547 sessions, 457 of which were indicated as successfully completing the goal, 19 goals and 43 actions (commands).

The Linux plan corpus is an important and hard benchmark for intention/goal recognition. First, data is collected from real humans and thus noisy. Second, involved humans expertise is varied, and they sometimes used wrong commands due to limited knowledge about the domain [8, 54]. Furthermore, we observe that plan sessions' lengths in the corpus are quite varied. The minimum, maximum, and mean number of actions in a plan session are 1, 60, and 6.124, respectively.

6.2.2 Learning Unit Fragments from Data

For unit fragment $UF_{\mathfrak{A}}(I, A)$, the conditional probability of A given I is defined by the frequency of A in a plan session for achieving the goal/intention I divided by the frequency of any action for achieving I:

$$P(A = T|I = T) = \frac{\mathit{freq}(A_I)}{\mathit{freq}(I)}. \quad (8)$$

For better understanding, in the plan corpus each action is marked with the intention which the action is aiming at. Then, $\mathit{freq}(A_I)$ is the frequency of A being marked by I, and $\mathit{freq}(I)$ is the frequency of seeing the mark I.

Note that prior probabilities of all the intentions in the corpus are given initially, and used for generating tasks for users [54, 8].

6.2.3 Making Predictions

Similar to [8, 3], instead of letting the recognizer make a prediction after each observed action, we set a *confidence* threshold τ $(0 \leq \tau \leq 1)$, which allows the recognizer to decide whether or not it is confident enough to make a prediction; the recognizer only makes a prediction if the likelihood of the most likely intention in the model is greater than τ. Otherwise, it predicts "don't know".

In addition, instead of only predicting the most likely intention, the recognizer provides a set of N most likely ones (*N-best prediction*).

6.2.4 Evaluation Metrics

For evaluating our system and comparing with the previous ones [8, 3], we use three different metrics. *Precision* and *recall* report the number of correct predictions divided by total predictions (predicts when confident enough) and total prediction opportunities (always predicts, whatever the confidence is), respectively. More formally [3], let $Seq = a_1, \ldots, a_n$ be a sequence of actions for achieving intention I (a session for testing). Considering the N-best prediction case, let

$$correct(A) = \begin{cases} 1 & \text{if } I \text{ is one of the } N \text{ most likely} \\ & \text{intentions when observing } A \\ 0 & \text{otherwise} \end{cases}$$

Then, precision and recall for Seq are defined as follows, respectively,

$$precision(Seq) = \frac{\sum_{i=1}^{n} correct(a_i)}{z},$$
$$recall(Seq) = \frac{\sum_{i=1}^{n} correct(a_i)}{Z}$$

where z and Z are the number of predictions made (when the recognizer is confident enough) and the total number of prediction opportunities, respectively.

On the other hand, *convergence* is a metric that indicates how much time the recognizer took to converge on what the current user goal/intention was. Let t be such that $correct_i = 0$ for $0 \leq i \leq t-1$ and 1 for $t \leq i \leq n$ (i.e. t is the first time point which from there on the system always correctly predicts), convergence for sequence Seq is

$$convergence(Seq) = \frac{z - t + 1}{z}.$$

Finally, the overall precision, recall and convergence are obtained by taking averages over all testing sessions.

6.2.5 Experiments and Results

Because of the small size of the Linux corpus, and similar to previous work, we ran experiments using the one-out cross validation method [3]. Just one at a time, one plan session in the whole corpus is left out. The rest of the corpus is used for training the model, which is then evaluated against the left out plan session. We study the effect of confidence level τ with respect to the precision and convergence (for recall, it clearly is a decreasing function of τ) (Figure 6). As expected, the greater N the better the precision and convergence scores. The difference in the precision and convergence between two different values of N is large when τ is small, and gets smaller for greater τ. Most interestingly, we observe that the precision and convergence are not monotonic increasing on τ. There are *critical values* of τ at which the measures have the maximal value, and those values are smaller for greater N. This observation suggests that in a plan/intention recognition task, the more precise (i.e. the smaller N) the decision needed to make is, the greater confidence level the recognizer should gain to make a good (enough) decision. On the other hand, the recognizer should not be too cautious, possibly leading to refuse to make a prediction when it would have been able to make a correct one. In short, these experiments suggest an important need to (experimentally) study the confidence threshold τ carefully, for particular application domains and particular values of N. Using the same τ for all values of N could decrease the recognizer's performance.

Table 1 shows some of the results for different values of N (and the corresponding value of τ). Similar to the previous works on the same Linux corpus [8, 3], we keep the best results of each case with respect to τ for the comparison. For example, we obtained a precision of 78.6% for 1-best that is increased to 87.0% for 3-best prediction and 88.3% for 4-best one. Convergence is increased from 72.2% for 1-best to 82.2% for 3-best and 82.4% for 4-best prediction.

The best performance on the Linux plan corpus so far, in terms of precision and convergence (recall is not referred), has been reported in [3], where the authors use a variable Markov model with exponential moving average. In this respect we obtained an increment of 14% better precision and 13.3% better convergence for 1-best prediction, 8.2% better precision and 9.3% better convergence for 2-best prediction, and 7.5% better precision and 7.7% better convergence for 3-best prediction. We also obtained better

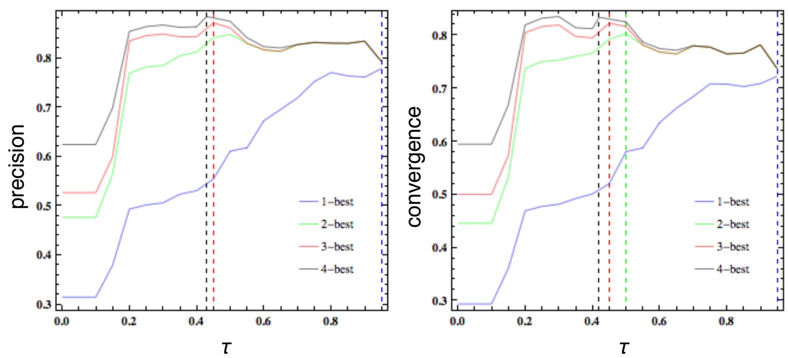


Figure 6: Plot of our method's precision and convergence for $\tau \in [0, 1]$ and for different values of N ($N = 1, 2, 3, 4$) in Linux plan Corpus. The greater N, the better precision and convergence. The difference in precision and convergence between two different values of N is large when τ is small, and gets smaller for greater τ. Most interestingly, we observe that precision and convergence are not monotonic increasing on τ. There are *critical values* of τ at which the measures have maximal value, and those values are smaller for greater N. This observation suggests that in plan/intention recognition task, the more precise (i.e. the smaller N) the decision needed to make is, the greater confidence level the recognizer should gain to make a good (enough) decision. On the other hand, the recognizer should not be too cautious to make a prediction, leading to refuse to make a prediction when it would have been able to make a correct one. In short, it suggests the important need to study (experimentally) the confidence threshold τ carefully for particular application domains.

recalls in all cases and for all metrics of 4-best prediction compared with [8], the only work reporting these scores. For the Linux plan corpus, [81] reports only the precision score, which is worse than that of [3].

Note that in [3] the authors use a more fine-grained preprocessing method for their work, but we suspect it will have increased their performance. To fairly compare with all previous work we use the original corpus.

The Linux plan corpus allows an appropriate comparison with prior work. However, it does not include contextual information (reasons/causes of intentions), and there is no intention change/abandonment occurrences (users follow a single intention throughout entire plan sessions). To evaluate the context-dependent aspect as well as the capability of dealing with intention change/abandonment, we next present new plan corpora.

7 IPD Plan Corpora

We present new plan corpora in the context of iterated Prisoner's Dilemma (IPD) [6] [79] and provide experimental results for them. The intentions/goals to be recognized are the (known) strategies in IPD (see below). Strategies in the (evolutionary) game theory context are usually considered representative types of agents' behavior in a social or biological setting. An agent that adopts a given strategy can be interpreted as if it intends to follow the corresponding type of behavior, in the given social or biological setting. Plan sessions are sequences of moves played by such strategies. This way of interpreting game strategies was explored in [35, 36] for studying the evolutionary roles of intention recognition.

[6] The approach also applies readily to other social dilemmas such as Snow Drift and Stag Hunt [79]. A social dilemma represents a particular type of situation in a social or biological context.

Table 1: Intention Recognition Results on the Linux Plan Corpus

N-best	**1-best**	**2-best**	**3-best**	**4-best**
τ	0.95	0.5	0.45	0.42
Precision	0.786	0.847	0.870	0.883
Recall	0.308	0.469	0.518	0.612
Converg.	0.722	0.799	0.822	0.824

7.1 Iterated Prisoner's Dilemma

The Prisoner's Dilemma (PD) is a symmetric two-player non-zero game defined by the payoff matrix

$$\begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} R, R & S, T \\ T, S & P, P \end{pmatrix}. \end{array}$$

Each player has two options in each round, cooperates (C) or defects (D). A player who chooses to cooperate with a player who defects receives the sucker's payoff S, whereas the defecting player gains the temptation to defect, T. Mutual cooperation (resp., defection) yields the reward R (resp., punishment P) for both players. PD is characterized by the payoff ranking $T > R > P > S$ (and, in addition, $2R > S + T$ for IPD). Thus, in a single round, it is always best to defect, but cooperation may be rewarded if the game is iterated. Let r denote the average number of rounds the game is iterated.

IPD is usually known as a story of tit-for-tat (TFT), which won both Axelrod's tournaments [4]. *TFT* starts by cooperating, and does whatever the opponent did in the previous round. It will cooperate if the opponent cooperated, and will defect if the opponent defected. But if there are erroneous moves due to noise (i.e. an intended move is wrongly performed with a given execution error), the performance of *TFT* declines: it cannot correct errors or mistakes. Tit-for-tat is then replaced by generous tit-for-tat (GTFT), a strategy that cooperates if the opponent cooperated in the previous round, but sometimes cooperates even if the opponent defected (with a fixed "forgiveness" probability $p > 0$) [60, 79]. *GTFT* can correct mistakes. Subsequently, *TFT* and *GTFT* were replaced by win-stay-lose-shift (WSLS) as the winning strategy chosen by evolution [61, 79]. *WSLS* repeats the previous move whenever it did well, but changes otherwise.

Some other less famous strategies which we are going to use later are *GRIM* – a grim version of *TFT*, prescribing to defect except after a round of mutual cooperation, and Firm-But-Fair (FBF) – known as a tolerant brother of *TFT*, prescribing to defect only if getting a sucker's payoff S in the previous round. Details of all strategies summarily depicted above can be found in [79].

7.2 Corpus Description

In the following, we prescribe how to create plan corpora for training and testing the described intention recognition method, for a given set of strategies. We start by making an assumption that all strategies to be recognized have the memory size bounded by $M \geq 0$, i.e. their decision at the current round is independent of the past rounds that are at a time distance greater than M. Note that the above mentioned strategies have memory bounded by $M = 1$.

For clarity of representation, abusing notations, R, S, T and P are henceforth also referred to as elementary game states, in a single round of interaction. Additionally, E (standing for *empty*) is used to refer to a game state having had no interaction. The most basic element in a plan corpus is the corpus action, having the following representation.

Definition 9 (Corpus Action) *An action in a plan corpus is of the form $s_1 \ldots s_M \xi$, where $s_i \in \{E, R, T, S, P\}$, $1 \leq i \leq M$, are the states of the M last interactions, and $\xi \in \{C, D\}$ is the current move* [7].

[7] From now on, this notion of an action is used, which is different from the notion of a move (either C or D).

Definition 10 (Plan Session) *A plan session of a strategy is a sequence of corpus actions played by that strategy (more precisely, of a player using that strategy) against an arbitrary player.*

We denote by Σ_M the set of all possible types of action for memory size M. Clearly, $|\Sigma_M| = 2 \times 5^M$. For example,

$$\Sigma_1 = \{EC, RC, TC, SC, PC, ED, RD, TD, SD, PD\}.$$

Now, for an example of a plan session, let us consider the strategy *TFT* and the following sequence of its interactions with some other player (denoted by X), in the presence of noise

$$\begin{array}{rcccccc}
\textbf{round}: & 0 & 1 & 2 & 3 & 4 & 5 \\
\textbf{TFT}: & - & C & C & D & D & D \\
\textbf{X}: & - & C & D & D & C & D \\
\textit{TFT-states}: & E & R & S & P & T & P
\end{array}$$

The corresponding plan session for *TFT* is $[EC, RC, SD, PD, TD]$. At the 0-th round, there is no interaction, thus the game state is E. *TFT* starts by cooperating (1-st round), hence the first action of the plan session is EC. Since player X also cooperates in the 1-st round, the game state at this round is R. *TFT* reciprocates in the 2-nd round by cooperating, hence the second action of the plan session is RC. Similarly for the third and the fourth actions. Now, at the 5-th round, *TFT* should cooperate since X cooperated in 4-th round, but because of noise, it makes an error to defect. Therefore, the 5-th action is TD.

This way of encoding actions and the assumption about the players' memory size lead to the equivalent assumption that the action in the current round is *independent* of the ones in previous rounds, regardless of the memory size. The independence of actions will allow to derive a convenient and efficient intention recognition model, discussed in the next subsection. Furthermore, it enables to save the game states without having to save the co-player's moves, thus simplifying the representation of plan corpora.

Definition 11 (Plan Corpus) *Let $\mathcal{S}$ be a set of strategies to be recognized. A plan corpus for $\mathcal{S}$ is a set of plan sessions generated for each strategy in the set.*

For a given set of strategies, different plan corpora can be generated for different purposes. In the sequel, for example, we generate different plan corpora for training and for testing the intention recognition method.

7.3 Plan Corpora Generation

Let us start by generating a plan corpus for seven most popular strategies within the IPD framework: *AllC* (always cooperate), *AllD* (always defect), *TFT*, *GTFT* (probability of forgiving a defect is $p = 0.5$), *WSLS*, *GRIM* and *FBF* (described above).

We collect plan sessions of each strategy by playing a random move (C or D) with it in each round. To be more thorough, we can also play all possible combinations for each given number of rounds r. E.g, if $r = 10$, there are $2^{10} = 1024$ combinations: C or D in each round. When noise is present, each combination is played repeatedly a number of times, since each time one might obtain different reaction moves from the simulated co-player.

The training corpus to be used here is generated by playing with each strategy all the possible combinations 10 times, for each number of rounds r from 5 to 10. The testing dataset is generated by playing a random move with each strategy in each round, also for r from 5 to 10 [8]. We continue until obtaining the same number of plan sessions as for the training dataset (corpus). Both datasets are generated in the presence of noise. Namely, an intended move is wrongly performed, with respect to the definition of the strategy, with probability 0.05.

In this testing dataset, changes/abandonment of players' initial intention (strategy) are not allowed. The players use the same strategy in all the rounds. We refer to this testing dataset as `Testset-IRFIX`.

[8] It might seem unnatural to interact with a strategy randomly, but it enables us to collect a highly diverse set of move sequences of the strategy, for both training and testing. It allows to thoroughly evaluate our intention recognizer, even if our additional experiments confirm that its relative performance (shown in Subsection 7.5) would not change much when the strategy interacts with a big enough set of strategies.

For testing the context-dependent aspect of our intention recognition method, as well as taking into account intention changes/abandonment, we next introduce the concept of *social learning* within the framework of evolutionary game theory [78, 45, 87].

7.4 Social Learning

We consider it necessary for an agent to acquire knowledge from other agents, i.e. learn "by being told" instead of learning only by experience. Indeed, this social learning is a fairly practical and economical way of increasing abilities, widely used by human beings, as widely studied in evolutionary biology and economics [72, 71, 78]. Let us see how social learning can be modeled in Evolutionary Game Theory (EGT) [45, 79] given a fixed set of strategies. Agent strategies can change, say through mutation or learning, but we shall not consider that issue here.

In social learning, agents in a population can observe the behavior of others and the outcomes of that behavior. They may copy the behavior of others whenever these appear to be more successful [79, 78, 71]. The accumulated payoff from all interactions emulates the agents' *fitness* or social *success* and the most successful agents will tend to be imitated by others. There are many ways to model social learning [45, 78]. The most popular one is implemented using the so-called pairwise comparison rule [84, 79]: an agent **A** with fitness f_A will adopt the strategy of a randomly chosen agent **B** with fitness f_B with a probability given by

$$p(f_A, f_B) = \left(1 + e^{-\beta(f_B - f_A)}\right)^{-1}, \qquad (9)$$

where the quantity β controls the "imitation strength", i.e. how strongly the players are basing the decision to imitate on payoff comparisons. Henceforth, **A** and **B** are referred to as imitating and imitated agents, respectively. For simplicity, we use $\beta = 1$ in the remainder of this article: the imitation depends on comparing the exact payoffs.

It is now allowed the possibility that a player can change his/her strategy (intention) by imitating the randomly met player's strategy (intention), depending on how much the latter player is more successful. The two players' ongoing success difference (SD) causally affects the imitating player's current intention. In addition, this intention is causally affected by the so-called imitation event (IE), stating whether the player is meeting some other player for learning/imitating.

In an everyday life situation, an imitation event, *IE*, can be more generally thought of as a contact of the intending agent with some source of information that might influence its current intention to change. For example, an intention of going out might be influenced to change/abandon by information about weather, traffic, etc., whether the source of that information is a person, a newspaper, TV or Radio. In turn, the success difference, *SD*, measures how strong is the influence on the change, based on the credibility or strength of the information. The above mentioned intention of going out is undoubtedly influenced more by the forecast information provided by a trustworthy TV channel than an untrustworthy one or unreliable person. Another example is that a suggestion from a doctor, after some tests, apparently has more effect on changing/abandoning the intention of drinking alcohol than that from a friend, or that from a daily newspaper.

Now we can specify a Bayesian network for intention recognition IRBN with two cause/reason nodes, a single intention node, and observed action nodes (Figure 7). In the following we define the CPD tables of the IRBN.

Bayesian Network definition. Recall that the intention node I has n possible outcomes I_i, $1 \leq i \leq n$ (Subsection 6.1). We define the conditional probability distribution $P(I_i|IE, SD)$. If the player does not meet any other player for imitation (i.e., $IE = F$), I_i is independent of the success difference *SD*. Hence, $P(I_i|IE = F, SD) = P(I_i|IE = F)$. Now, let us consider the case $IE = T$. If the successes are also observable (thus, *SD* is observed, say, equal χ)[9],

[9] There may be noise in the evaluation of the successes. The observed value χ of SD is randomly taken in the range $((1-\epsilon)\chi_1, (1+\epsilon)\chi_1)$, where ϵ is a small positive number (here we use $\epsilon = 0.01$) and χ_1 is the exact value of the difference.

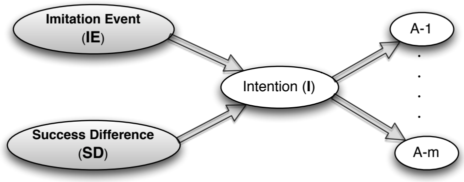


Figure 7: Bayesian Network for intention recognition (IRBN) in the context of iterated Prisoner's Dilemma, where players can change their initial intention (strategy) by imitating successful others, based on a social learning model [45, 78, 71]. The IRBN have two cause/reason nodes in the first layer (IE and SD), connecting to a single intention node in the second layer (I), in turn connecting to action nodes in the third layer.

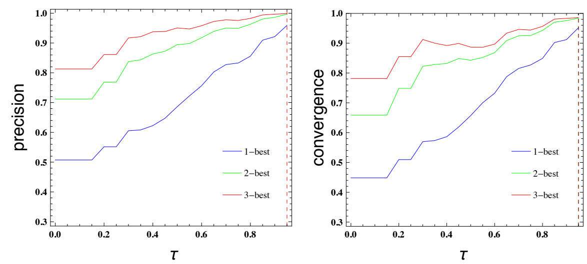


Figure 8: Precision and convergence for $\tau \in [0, 1]$ and for different values of N ($N = 1, 2, 3$) with respect to `Testset-IRFIX` dataset, where N is the number of most likely considered intentions. The greater N, the better the precision and convergence. For high enough confidence τ, the intention recognizer obtained quite good scores.

but the strategy of the imitated player is not, we have

$$P(I_i|IE = T, SD = \chi) = (1 - u)p_i + \frac{u}{S-1}\sum_{j \neq i} p_j \tag{10}$$

where $u = (1 + e^{-\chi})^{-1}$, p_i is the probability that I_i was the player's intention in the last prediction, and S is the number of strategies in the corpus. The equation is explained as follows. With probability $(1 - u)p_i$ the imitating player's strategy remains I_i. Moreover, when not being observed, the probability that I_i was the imitated player's strategy is (assumed) equal $1/(S - 1)$. The second term expresses the probability that the player adopts the new strategy I_i by imitation.

In the case the imitated player's strategy is also observed, denoted by I_{i^*}, similarly we have

$$\begin{aligned} P(I_{i^*}|IE = T, SD = \chi) &= (1 - u)p_i + u\sum_{j \neq i^*} p_j, \\ P(I_i|IE = T, SD = \chi) &= (1 - u)p_i \quad \forall\ i \neq i^*. \end{aligned} \tag{11}$$

Testing Dataset Generation (IRCHANGE). The testing dataset in this setting is generated by using a simplified evolutionary simulation involving the seven strategies described in Subsection 7.3. Similarly to the previous dataset, we play a random choice sequence with each of the seven above mentioned strategies for 10 rounds. The payoff of each strategy is accumulated over these rounds. Then, for each strategy, another strategy is randomly chosen from the other six, for possible imitation. The imitation decision is guided by the pairwise comparison rule (see Eq. (9)), which is based on a comparison of the performance of the two strategies, in terms of the difference of their accumulated payoffs in these 10 rounds.

After all the seven strategies are given the chance to change their strategy by imitating another, the interactions are repeated for 10 more rounds. At the 10-th round, we save the accumulated payoff values of the imitating and imitated strategies. We experiment until obtaining the same number of plan sessions as in the training dataset. The PD payoff matrix being used is: $T = 20$, $R = 15$, $P = 10$, $S = 5$; and $noise = 0.05$. This testing dataset is referred to as `Testset-IRCHANGE`.

Clearly, each plan session in the dataset consists of 20 actions. The first 10 actions are played by the initial strategy. The last 10 ones are played by a strategy which might be either the initial one or a different one, as the outcome of imitation (or social learning). The intention recognition system must identify both the first and the second strategies. While identifying the first strategy can be done efficiently with-

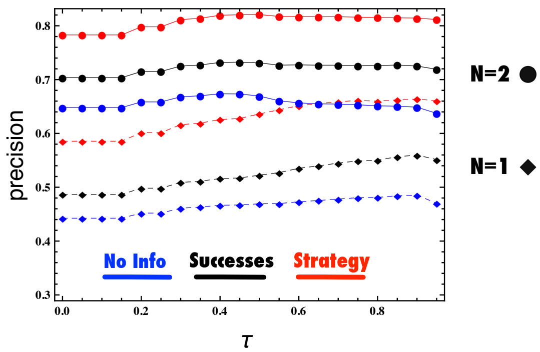


Figure 9: Precision for three different levels of contextual information, for $\tau \in [0, 1]$, with respect to `Testset-IRCHANGE` dataset. We plot for $N = 1$ (dashed diamond) and $N = 2$ (circle).

out taking into account contextual information, as we have seen in the previous datasets, identifying the second strategy is more difficult due to the stochastic change of the strategies (resulting from the stochasticity from the comparison rule). In the course of observing the first 10 actions generated by the initial strategy, the system (on average) converges to a correct prediction of it. If the strategy is changed to a different one, and the system has no clue about that change, it would poorly predict the new strategy, since it would still consider the old actions generated by the first strategy as belonging to the new strategy. Having (contextual) clues about the potential effects leading to strategy change could significantly enhance the prediction of the second strategy, as we shall see from our experimental results in the sequel.

7.5 Results on IPD Plan Corpora

The intention recognition model is acquired using the training corpus. Figure 8 shows the precision and convergence of the model with respect to the `Testset-IRFIX`, where again N is the number of most likely considered intentions. Given that the training as well as the testing datasets are generated in the presence of noise, the achieved performance is quite good. Namely, for a big enough τ, both precision and convergence scores are greater than 0.9, even for the 1-best case. It is noteworthy that in [35] a similar good performance is obtained for a smaller set of four strategies (intentions). Therein, based on the intention recognition model above presented, we implement a novel strategy capable of intention recognition that outperforms other strategies in an evolutionary setting.

In Figure 9 we show the effects of having different levels of contextual information on the intention recognition performance, using `Testset-IRCHANGE` dataset, where we consider $N = 1$ (dashed diamond) and $N = 2$ (circle). Namely, in the first case (blue curves), there is no information about the imitation event (IE) – it is not known if the recognized player may imitate and adopt another strategy. In the second case (black curves), *IE* and the successes are observable. In the third case (red curves), the strategy of the imitated player is also observable (see a discussion below for how these three levels of contextual information can be mapped into real world situations). It is clearly shown that the performance is considerably increased as more contextual information is available. Namely, comparing with the first case where no contextual information is taken into account, an increase of about 5% and 15% precision is achieved in the second and third cases, respectively.

These results show that, besides observable actions from the recognizing agent, it is important to take into account contextual information for intention recognition. That is what most previous work on intention/plan recognition has omitted, with the exception of [69], though the authors did not provide experimental results to show its importance (see Related Work in Section 2).

Note that our three-layer Bayesian network model described in Figure 7 is not restricted to context-dependent intention recognition in the (evolutionary) game theory setting. The two cause/reason nodes in the BN, imitation event *(IE)* and success difference *(SD)*, account for key relevant contextual information for intention recognition. The first one stands for relevant factors that might influence the stability of current intentions in the model, while the second measures how strong are such influences. Conceivably, those are all the relevant contextual information needed for the context-dependent intention recognition task in a given application domain.

We next further discuss the rationale of using (evolutionary) game-theoretic methods and concepts for modeling real world situations.

7.6 Scope Enlargement Discussion

The task of intention/plan recognition is mostly concerned with actions or behavior. In previous work, for a given application domain, these actions are supposed to be supplied by an *activity* recognition system. In contrast, here we aim at providing a general framework (and benchmarks) for studying and evaluating intention/plan recognition models. For that, it is crucial to have a general model of agent behavior and its changes.

We adopted here one of the most popular frameworks for modelling agents' behavior (including those amongst humans), the social dilemmas (not only the Prisoner's dilemma) and social learning within the framework of evolutionary game theory [45, 79, 71]. They have been widely used in Economics, Biology, Psychology, Artificial Intelligence, etc. to describe human (and other species) behavior [4, 78, 85, 79, 71, 87]. Although they do not exhibit all aspects of human behavior in the real world, they have been, experimentally as well as analytically, shown to reflect its core nature (most famously, in the context of the evolution of cooperation study [4, 85, 58, 75, 79]). For example, the notorious strategies described above (including *TFT, WSLS, GTFT*) frequently show up in many real-world experimental settings, ranging from Global Warming issues to Psychology and Economics behavioral researches [57, 23, 58]. They are also widely observed in a broad range of real human social behavior. For a review of social learning and evolutionary games in the real world the readers are referred to the seminal experimental studies in [85, 58].

Using game-theoretic concepts allows a higher level of abstraction of domain knowledge. That is, it enables us to account for and generalize similar situations in different application domains. As a result, our IPD benchmarks above presented are deemed domain-independent. Let us consider a simple scenario in the Elder Care domain to illustrate how it is mapped to game theoretic concepts, namely the three levels of contextual information, "No Info", "Successes", and "Strategy", alluded above (Figure 9). Suppose an elder intends to go fishing the next morning. But 'being told' that evening that it is going to rain then, it was suggested (he intend) to do something else. Whether he changes (or even abandons) his intention strongly depends on by whom he was 'being told'. Was it his 3-year old granddaughter or the TV/Radio forecast? That is, whether he adopts another person's suggested intention (of not going fishing the next morning) depends on that person's expertise or past success in the matter, or the credibility of the information provided. It seems almost sure that he would drop the intention of going fishing if it was said by the TV/Radio forecast. In order to appropriately recognize the elder's intention, it is crucial to account for such contextual information, and that it was encountered. Further, if he was also 'being told' by the forecast that, although it is going to rain in the morning, the weather in the afternoon should be perfect, one can predict that he might likely to adopt the intention of going fishing in the afternoon instead. Briefly, it corresponds to the first case, "No Info", if the system cannot observe (despite its actual occurrence) that the elder was 'being told' by another person. The second case, "Successes", corresponds to when the system can observe who passed him the information. The third case, "Strategy", corresponds to when the system also observes what he was advised by another.

In short, using the models of evolutionary game theory and social learning, one can generate general-purpose or domain-independent benchmarks for intention/plan recognition systems. The applicability of a system to such benchmarks may serve to show its applicability to a wider range of application domains.

8 More on Situation Sensitiveness

In the following we discuss some extensions to the above intention recognition model, which enable to take into account contextual information in several manners. We mainly focus on exploiting the powerful knowledge representation tool of Logic Programming (LP) [5] [10].

8.1 Situation-sensitive Bayesian Networks

Undoubtedly, BNs should be situation-sensitive since using a general BN for all specific situations of a problem domain is unrealistic and most likely imprecise [89, 51]. For example, in the Elder Care domain, different elders might have different living conditions and habits that need to be taken into account to recognize their intentions. Also, place, time of day, temperature, etc. need to be considered [67, 31]. However, consulting the domain expert to manually change the BN with respect to each situation is costly or unfeasible.

In [65, 67, 41], we described a method to construct situation-sensitive BNs, i.e. ones that change according to the given situation. It uses LP techniques to compute situation specific probabilistic information that is then updated into the BN. The BNs themselves are also encoded with P-log, a probabilistic logic implemented system [6, 38, 39, 41], which supports coherent updates. The LP techniques employed for this are deduction with top-down procedure (XSB-Prolog) [90] (to deduce situation-specific probabilistic information) plus integrity constraints and abduction (to abduce probabilistic information needed to support and explain observations in the given situation). However, we can employ too various other types of LP based reasoning, e.g., constraint satisfaction, contradiction removal, preferences, or inductive learning, whose results can be compiled (in part) into an evolving BN.

Consider three-layer BN for intention recognition IRBN, we assume that only prior probabilities of the top nodes (causes/reasons) are subject to changes along with the changing situation. The CPD tables for intention and action nodes reflect the internal state of the recognizing agent about world [44, 83]. They may change in the long-term periods, but just occasionally. In short, for a given situation at hand, an IRBN can be situated in it by recomputing the prior probabilities of the cause nodes in accordance with the situation.

Based on this idea, we now revise the Algorithm 1, making the IRBN model situation-sensitive. For that, we first describe a new operator. In a given situation, an IRBN can be situated by recomputing the prior probabilities of the top nodes.

Definition 12 (Situate IRBN) *Let* $W = \langle\{Cs, Is, As\}, pa, P_W\rangle$ *be an IRBN. We say that W is situated into a situation* SIT *if the prior probabilities of the top nodes of W, i.e.* $P_W(C)$ $(C \in Cs)$*, are recomputed according to* SIT*. In this work, a situation is encoded by a logic program and the prior probabilities of the top nodes are computed using LP techniques, as described above [65, 67]. Formally, the situate operator is defined by* `situate(W,SIT)` $= \langle\{Cs, Is, As\}, pa, P_S\rangle$*, where*

- $P_S(C)$*, for all* $C \in Cs$*, are the new prior probabilities of top nodes, resulted from the recomputation according to SIT.*
- $P_S(X|pa(X)) = P_W(X|pa(X))\ \forall X \in Is \cup As.$

It might be too costly to always change the IRBN as the situation relevant factors could constantly evolve. Here, we propose a simple criterion allowing to decide when the IRBN should be reconfigured to account for the situation at hand. It says, if there is a conceivable "salient" intention currently figuring in the IRBN, the IRBN should be reconfigured (after observing a new action), that is, be situated according to the latest information about the situation at hand. Otherwise, the IRBN remains the same. The property "salient" reads differently in different application domains, and it is up to the domain experts to design its specification. For example, in the Elder Care do-

[10] Here we consider solely normal logic programs, which consist of a set of rules of the form

$$A \leftarrow B_1, \ldots, B_m,\ not\ C_1, \ldots,\ not\ C_n \qquad (m, n \geq 0)$$

where A, $B_1, \ldots, B_m$, $C_1, \ldots, C_m$ are domain atoms, and $not\ C_i$ denotes the default negation of atom C_i, i.e. being false by default. The rule reads, if all the atoms and default negations in the body are true, then the head A is true. If the body is empty, that is, $m = n = 0$, the head is true unconditionally. In this case, it has the form, $A \leftarrow$ or simply A, and is also called a *fact*.

main, "salient" may read *dangerous*, e.g., when suspecting an intrusion intention to elders' house, or the elders' committal suicide intention. It may also read *urgent*, e.g., when detecting the elders' intention to find something to eat or something drink.

Accordingly, a simple change to the Algorithm 1 needs to be made.

Algorithm 2 *After the first step (item 1) in Algorithm 1, insert the following item:*

- *If there is a "salient" intention in the current IRBN W', situate it according to the situation at hand SIT, i.e.* situate(W′,SIT). *Otherwise, the IRBN remains the same.*

For illustration, let us extend the previous example in the Elder Care domain.

Example 5 (Elder Care (cont'd)) *In the scenario provided in Example 4, the IRBN model (after observing and including both actions) may vary depending on some factors such as the time of day, of the elders' last drink or last meal, his interest in football, etc. For illustration, we design a simple logical component for the IRBN to take into account those factors.*

```
pa_rule(pa(hg(T),d_(0,1)),[])
         :- time(X), last_eating(X1), X-X1 < 1.
pa_rule(pa(hg(T),d_(9,10)),[])
         :- time(X), last_eating(X1), X-X1 > 3.

pa_rule(pa(thty(T),d_(1,10)),[])
        :- time(X), last_drink(X1), X1-X < 1.
pa_rule(pa(thty(T),d_(9,10)),[])
        :- time(X), last_drink(X1), X1-X > 3.

pa_rule(pa(lr(T),d_(1,100)),[])
       :- time(X), X > 0, X < 5.

pa_rule(pa(lw(T),d_(9,10)),[])
 :- time(X), schedule(X1,football), X1 - X < 0.25, !.
pa_rule(pa(lw(T),d_(1,100)),[])
 :- time(X), (X > 23; X < 5).
```

Basically, in P-log, probabilistic information is given by $pa/2$ rules [38, 6]. For example, the rule $(pa(hg(T), d\_(9, 10)) \leftarrow Body)$ means that the probability of being hungry (i.e. $hg(T)$) is 9/10 if the precondition in the body of the rule, *Body*, holds. We provide a reserved $pa\_rule/2$ predicate which takes the head and body of some $pa/2$ rule as its first and second arguments, respectively, and includes preconditions for its activation in its own body. Thus, e.g., the second *pa_rule* above means, if the elder's last eating is more than 3 hours ago, it is quite probably that he is being hungry (with probability 9/10) (note that here the *pa* rule body, representing the precondition, is empty—represented by an empty list []) Now, a situation is given by asserted facts representing it and, in order to find the probabilistic information specific to the given situation, we simply use the XSB Prolog built-in *findall/3* predicate [90] to find all true $pa/2$ literals expressed by the $pa\_rule/2$ rules with true bodies in the situation.

There are several predicates specific to the domain knowledge base. For example, $time/1$ provides the current time of the day (in hours). The predicates $last\_drink/1$ and $last\_eating/1$, with an argument in time, hold if the elder's last drink and last meal were at the given time, respectively.

Now let us exhibit how some different situations can be encoded in our framework, and how they appropriately affect the intention recognition results, in a context-dependent manner.

- If the current time is 18 (i.e. $time(18)$) and the last time the elders ate was half an hour before (i.e. $last\_eating(17.5)$). But they did not have any drink for 3 hours (e.g., $last\_drink(14)$). Those three facts are asserted. Hence, the following two $pa\_rule/2$ literals are true, and are updated into the general IRBN

  ```
  pa_rule(pa(hg(T),d_(0,1)),[]).
  pa_rule(pa(thty(T),d_(9,10)),[]).
  ```

  Now the result is, $V_1 = 0.55$; $V_2 = 0.77$; $V_3 = 0.41$; $V_5 = 0.21$. It is now the case that looking for something to drink, *i(Drink)*, is the most likely intention, instead of looking for food as previously.

- If the elder also had the last drink just half an hour before (i.e. $last\_drink(17.5)$), then we have the following two rules instead,

  ```
  pa_rule(pa(hg(T),d_(0,1)),[]).
  ```

```
pa_rule(pa(thty(T),d_(1,10)),[]).
```

We obtain, $V_1 = 0.60$; $V_2 = 0.14$; $V_3 = 0.54$; $V_5 = 0.12$. In other words, the intention of looking for something to read, *i(Book)*, becomes the most likely intention.

- If we remain the situation, but shifting the time to *1 a.m.*, that is, the facts $time(1)$, $last\_eating(0.5)$, and $last\_drink(0.5)$ hold. And if there is no football scheduled around that time, we additionally have two rules

  ```
  pa_rule(pa(lr(T),d_(1,100)),[]).
  pa_rule(pa(lw(T),d_(1,100)),[]).
  ```

  We thus obtain, $V_1 = 0.40$; $V_2 = 0.14$; $V_3 = 0.29$; $V_5 = 0.121$, i.e., the intention of looking for something to read, *i(Book)*, remains the most likely intention. But, if there is football scheduled around that time, say, $schedule(1.15, football)$, the we have additionally the following two rules instead,

  ```
  pa_rule(pa(lr(T),d_(1,100)),[]).
  pa_rule(pa(lw(T),d_(9,10)),[]).
  ```

  Now the result is, $V_1 = 0.13$; $V_2 = 0.14$; $V_3 = 0.21$; $V_5 = 0.12$. In other words, he intention of looking for the remote control now becomes the most likely one.

In short, we have demonstrated the flexibility of our technique by means of representing situations as logic programs, and used them to update the current IRBN. We show how the context information can appropriately influence the interpretation of the observed agent's intention.

8.2 Situation-sensitive selection of unit fragments

In all the plan corpora benchmarks we have been using in previous sections, no context-dependent selection of unit fragments for actions is needed. In operator *select(A,SIT)* (recall from Subsection 4.1), the parameter *SIT* is always empty. However, the selected set of intentions for an action should in general be context dependent, that is, which intentions conceivably give rise to the action should depend on the situation in which the action is observed [11, 10, 73]. Commonsense reasoning can be employed for this purpose, which will be enacted by LP reasoning techniques [5] in the following.

The prior domain knowledge base KB is accompanied by a logic program P_{KB} in order to help decide which unit fragments for an observed action are selected in a given situation, i.e., which intentions are conceivable. Let $\Delta = \{A_1, \ldots, A_N\}$ be the set of actions of KB and $\Upsilon_i = \{I |\ UF_{\mathfrak{A}}(I, A_i) \in KB\}$ the set of intentions that belong to a unit fragment for action A_i in KB.

We say that an intention I is conceivable when observing action A if it is expected in the given situation and there is no expectation to the contrary (a similar technique was used in [64, 67, 32]). Thus, for $1 \leq i \leq N$ and $I \in \Upsilon_i$, P_{KB} contains the following rule:

$$conceivable(I) \leftarrow A_i,\ expect(I),\ not\ expect\_not(I).$$

Furthermore, for each $I \in \bigcup_{i=1}^{N} \Upsilon_i$, P_{KB} contains two rules:

$$\begin{aligned} expect(I) &\leftarrow Cond_1, \\ expect\_not(I) &\leftarrow Cond_2. \end{aligned} \tag{12}$$

The rules about expectations are domain-specific knowledge used to constrain the conceivable intentions in a situation. Counter-expectation rules supplement expectation rules for representing defeasible conditions or exceptions.

Now suppose that an action A, $A \in \Delta$, is observed. The current situation is encoded by a logic program SIT, consisting of LP facts describing the situation. In order to compute the set of conceivable intentions that may give rise to A_i, we simply use the XSB Prolog built-in *findall/3* predicate [90] to find all true *conceivable/1* atoms of the program $P_{KB} \cup SIT \cup \{A \leftarrow\}$. This provides the operational definition of the $select(A, SIT)$ operator discussed in Subsection 4.1.

For illustration, we modified the previous example in Elder Care domain, as follows.

Example 6 (Elder Care variant) *An elderly person stays alone in his apartment. An intention recognition system is set up to support his activities in the living room. At the moment the system observes that the elder is looking around for something (*look*). The knowledge base KB of the system has a unit IRBN for this action. For illustration, consider a small set of conceivable intentions,* $Is = \{book, water, weapon, lightSwitch\}$.

The accompanying logic program P_{KB} contains the following rules, for each $I \in Is$:

$conceivable(I) \leftarrow look,\ expect(I),\ not\ expect\_not(I).$

Suppose in P_{KB} the expectation and counter-expectation rules for these intentions are

1. $expect(book).$
   $expect\_not(book) \leftarrow light\_off.$
   $expect\_not(book) \leftarrow burglar\_alarm\_ring.$
2. $expect(water).$
   $expect\_not(water) \leftarrow light\_off.$
   $expect\_not(water) \leftarrow burglar\_alarm\_ring.$
3. $expect(weapon) \leftarrow burglar\_alarm\_ring.$
   $expect\_not(weapon) \leftarrow light\_off.$
   $expect\_not(weapon) \leftarrow no\_weapon\_availabe.$
4. $expect(lightSwitch).$
   $expect\_not(lightSwitch) \leftarrow light\_on, tv\_on.$

For example, the rules in line 1 say the intention of looking for a book is always expected except when the light is off or the burglar alarm is ringing. Let us consider some cases. If at the moment the light is off, i.e. $SIT = \{light\_off \leftarrow\}$, then $conceivable(light\_switch)$ is the only true $conceivable/1$ atom of the program $P_{KB} \cup SIT \cup \{look \leftarrow\}$. In this case, since there is only one conceivable intention, we can conclude immediately that looking for the light switch is the elder's intention, without having to construct an IRBN.

Now suppose the light is on, the tv is not on, and the burglar alarm is not ringing. That is, $SIT = \{light\_on \leftarrow\}$. Recall that what we consider here are normal logic programs, with default negation. It means that if something is not stated true as a fact, its negation is true by default. There are three conceivable intentions, *book, water,* and *lightSwitch*, since they are expected in this situation, and there are no expectations to the contrary (lines 1, 2, and 4). Hence, they are selected for constructing the unit IRBN for action *look*, given the current situation. The intention *weapon* is irrelevant, or not conceivable, and can be ruled out immediately at this situation-dependent selection stage.

If light is on, tv is not on, but now the burglar alarm is ringing, i.e, $SIT = \{light\_on \leftarrow,\ burglar\_alarm\_ring \leftarrow\}$. *Weapon* is accordingly expected, and there is no expectation to the contrary (line 3). Furthermore, *water* and *book* are not expected (line 2). Hence, there are two conceivable intentions: *weapon* and *lightSwitch*.

If, additionally, the tv is on, that is, $SIT = \{light\_on \leftarrow,\quad tv\_on \leftarrow,$ $burglar\_alarm\_ring \leftarrow\}$, then the intention of looking for the light switch is not expected (line 4). Hence, looking for *weapon* is the only conceivable intention recognizable in this situation.

Despite its simplicity, in considering only a small set of possible conceivable intentions, this example has demonstrated how the LP techniques can help to (significantly) reduce the size of the result IRBN, and sometimes reach the solution immediately. We envisage that the approach can provide even greater advantage, in terms of time complexity, when the set of conceivable intentions is large.

In this section we have presented two LP techniques which aim at enabling and enhancing context-dependent intention recognition. Taking into account contextual information can help significantly reduce the size of the IRBN in which the BN inference is performed. Although much work remains to be done—including gathering real data or generating appropriate plan corpora to evaluate them—the proposed techniques are quite promising. We envisage that all these might be done using BNs solely, it however appears that their combination can provide a more efficient solution. The BN inference is much costlier than LP inference. For example, consider a (definite) logic program that encodes all the causal relations of a BN. In the general case, while the inference, e.g. of a conditional probability in the BN, is exponential, the corresponding deductive inference in the logic program can be performed in polynomial time.

9 Conclusions and Future Work

We have presented a novel method for context-dependent and incremental intention recognition. The method is performed by dynamically constructing a three-layer BN model for intention recognition (IRBN), from a prior knowledge base consisting of readily maintained and constructed fragments of BN. Their simple structures allow easy maintenance by domain experts or automatically building from available plan corpora. The three-layer IRBN follows a *causal* intentional structure [44, 83], from causes/reasons (of intentions) to intentions (as causes of actions), then to (actually observed) actions. This causal structure enabled us to appropriately capture different aspects of the context-dependent intention recognition, from the context-dependent selection of intentions for the model construction to sensitizing prior probabilities of the intentions already in the model.

The presented above examples in the Elder Care domain illustrate several aspects of our intention recognition method, highlighting the importance of taking into account contextual information – an aspect usually omitted in the previous work [69, 26]. In addition, for the first time, we have shown experimentally that accounting for contextual information is crucial for the making of more accurate decisions about others' intentions, whenever the observed agents might change or abandon their initial goals/intentions. It is apparently an unavoidable aspect of real agents, clearly pointed out in [25, 26]. Taking into account such information enables us to reckon with the reasons why the agents change or abandon their goals/intentions.

Our method performs particularly well on the Linux plan corpus, showing its applicability to the important interface-agents domain [47, 2, 55]. It outperforms the existent methods that make use of the corpus. For further experimentation, we have presented the so-called IPD plan corpora for the famous strategies in the context of the iterated Prisoner's Dilemma. We employed the popular model of human behavior enacted by means of social learning and evolutionary game theory to simulate intention changes/abandonment—thus enabling us to evaluate the context-dependent aspect of our intention recognizer as well as its capability for dealing with intention changes/abandonment. Given that this modelling approach has been widely adopted and employed in several fields, as diverse as Economics, Psychology and Biology [45, 79], the good performance of the method for the IPD corpora makes it highly applicable for a wide range of application domains therein. In [35, 36, 34], we employ this intention recognition method to implement strategies that outperform the most successful known strategies in the iterated Prisoner's Dilemma [37].

Herein we have also attempted to tackle the problem where an observed agent may follow multiple intentions simultaneously in a more appropriate manner. We formally described how to represent relationships amongst intentions in the Bayesian network for intention recognition, particularly in order to maintain its consistency when one needs to combine mutually exclusive intentions [69]. This aspect is indispensable in multiple intentions recognition, but mostly omitted in previous work. However, the scalability of our method remains to be seen. For its evaluation we need to gather an appropriate plan corpus allowing for the possibility that users might pursue multiple intentions simultaneously. An idea is that instead of giving users one task at a time (in the case of the Linux or Unix domain [53, 8]), they are given several, so that they can complete them separately or in parallel. Another method is to use a planner to generate interleaved plans for multiple intentions.

A limitation of the current formalization in the multiple intentions recognition case is that we need to assume that the intentions to be combined are perfectly mutually exclusive. This assumption can be relaxed by utilizing a latent variable for any subset of perfectly mutually exclusive intention nodes. The latent variable figures in the BN either as a child or parent of the nodes, whichever works better for inference. We are exploring this direction to provide a more general method for representing relationships amongst intention nodes.

Another limitation of our current method is that it did not explicitly take into account temporal evolution of domain variables in the BN. It is usually done using Dynamic Bayesian Networks (DBNs) [69, 24, 1], where the state of each variable is represented by a series of nodes. In our method, the time evolution, to some degree, is implemented by means of updating the IRBN from time to time. For example, the states of the cause/reason nodes are updated using an external logic program which represents the evolving world. In this way not only can one significantly reduce the size of the BN, thereby its inference complexity (as the LP inference is significantly less expensive), but the declarative representation and reasoning of the LP techniques [5] could be important when the states of a node cannot be easily represented in an explicit manner. However, in domains where the states of the nodes might constantly change, the explicit time series representation of DBNs is apparently necessary. We envisage to bring in, to some degree, the time series representation of DBNs [20, 24, 1], to improve our method.

Acknowledgments

TAH acknowledges the support from the Belgian FWO Vlaanderen (Postdoctoral fellowship). Part of this work was previously done with support from FCT-Portugal (grant SFRH/BD/62373/2009).

References

[1] D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8:5–47, January 1998.

[2] M. G. Armentano and A. Amandi. Plan recognition for interface agents. *Artif. Intell. Rev.*, 28:131–162, August 2007.

[3] M. G. Armentano and A. Amandi. Goal recognition with variable-order markov models. In *Proceedings of the 21st international joint conference on Artificial intelligence*, pages 1635–1640, 2009.

[4] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, ISBN 0-465-02122-2, 1984.

[5] C. Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving.* Cambridge University Press, New York, NY, USA, 2003.

[6] C. Baral, M. Gelfond, and N. Rushton. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144, 2009.

[7] N. Blaylock and J. Allen. Corpus-based, statistical goal recognition. In *Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03)*, pages 1303–1308, 2003.

[8] N. Blaylock and J. Allen. Statistical goal parameter recognition. In Shlomo Zilberstein, Jana Koehler, and Sven Koenig, editors, *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS'04)*, pages 297–304. AAAI, 2004.

[9] N. Blaylock and J. Allen. Recognizing instantiated goals using statistical methods. In Gal Kaminka, editor, *IJCAI Workshop on Modeling Others from Observations (MOO-2005)*, pages 79–86, 2005.

[10] M. E. Bratman. *Intention, Plans, and Practical Reason.* The David Hume Series, CSLI, 1987.

[11] M. E. Bratman. Planning and the stability of intention. *Minds and Machines*, 2(1):1–16, 1992.

[12] S. M. Brown. *A decision theoretic approach for interface agent development.* PhD thesis, Faculty of the Graduate School of Engineering of the Air Force Institute of Technology Air University, USA, 1998.

[13] H. Bui. A general model for online probabilistic plan recognition. In G. Gottlob and

T. Walsh, editors, *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1309–1318, 2003.

[14] H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

[15] E. Charniak and R. P. Goldman. A probabilistic model of plan recognition. In *Proceedings of the ninth National conference on Artificial intelligence - Volume 1 (AAAI'1991)*, AAAI'91, pages 160–165. AAAI Press, 1991.

[16] E. Charniak and R. P. Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993.

[17] P. R. Cohen and H. J. Levesque. Intention is Choice with Commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[18] C. Conati, A. S. Gertner, K. Vanlehn, and M. J. Druzdzel. On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks. In *Proceedings of the Sixth International Conference on User Modeling*, pages 231–242, 1997.

[19] Fabio G. Cozman. Axiomatizing noisy-or. In Ramon Lopez de Mantaras and Lorenza Saitta, editors, *In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04)*, pages 979–980. IOS Press, 2004.

[20] T. Dean and K. Kanazawa. Probabilistic Temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 524–528. AAAI, 1992.

[21] Francisco Javier Diez. Parameter adjustment in Bayes networks. the generalized noisy or-gate. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 99–105. Morgan Kaufmann, 1993.

[22] Thi Duong, Dinh Phung, Hung Bui, and Svetha Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. *Artif. Intell.*, 173:830–856, May 2009.

[23] Armin Falk, Ernst Fehr, and Urs Fischbacher. Testing theories of fairness—intentions matter. *Games and Economic Behavior*, 62(1):287 – 303, 2008.

[24] Jeff Forbes, Tim Huang, Keiji Kanazawa, and Stuart Russell. The batmobile: towards a Bayesian automated taxi. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, pages 1878–1885, 1995.

[25] C. W. Geib and R. P. Goldman. Recognizing plan/goal abandonment. In *Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03)*, pages 1515–1517, 2003.

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

[27] Christopher W. Geib. Assessing the complexity of plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI'2004)*, pages 507–512, 2004.

[28] V. Gogate and R. Dechter. SampleSearch: Importance sampling in presence of determinism. *Artificial Intelligence*, 175(2):694–729, 2011.

[29] R. P. Goldman, C. W. Geib, and C. Miller. A new model of plan recognition. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 245–254, 1999.

[30] T. A. Han and L. M. Pereira. Anytime intention recognition via incremental Bayesian network reconstruction. In *AAAI 2010 Fall Symposium on Proactive Assistant Agents (PAA 2010)*, pages 20–25. AAAI, 2010.

[31] T. A. Han and L. M. Pereira. Collective intention recognition and elder care. In *AAAI 2010 Fall Symposium on Proactive Assistant Agents (PAA 2010)*, pages 26–31. AAAI, 2010.

[32] T. A. Han and L. M. Pereira. Intention-based decision making with evolution prospection. In L. Antunes and H.S. Pinto, editors, *Proceedings of 15th Portuguese Conference on Artificial Intelligence (EPIA'2011)*, pages 254–267. Springer LNAI 7026, 2011.

[33] T. A. Han and L. M. Pereira. Intention-based decision making for strategic scenarios dynamics via computational logic. In *Proceedings of 16th Portuguese Conference on Artificial Intelligence (EPIA'2013)*, Portugal, 2013.

[34] T. A. Han, L. M. Pereira, and F. C. Santos. Intention recognition promotes the emergence of cooperation. *Adaptive Behavior*, 19(3):264–279, 2011.

[35] T. A. Han, L. M. Pereira, and F. C. Santos. The role of intention recognition in the evolution of cooperative behavior. In T. Walsh, editor, *Proceedings of the 22nd international joint conference on Artificial intelligence (IJCAI'2011)*, pages 1684–1689. AAAI, 2011.

[36] T. A. Han, L. M. Pereira, and F. C. Santos. Corpus-based intention recognition in cooperation dilemmas. *Artificial Life journal*, 18(4):365–383, 2012.

[37] T. A. Han, L. M. Pereira, and F. C. Santos. Intention Recognition, Commitment, and The Evolution of Cooperation. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE Press, June 2012.

[38] T. A. Han, C. K. Ramli, and C. V. Damásio. An implementation of extended P-log using XASP. In *Proceedings of International Conference on Logic Programming (ICLP'08)*, pages 739–743. Springer LNCS 5366, 2008.

[39] T. A. Han, C. K. Ramli, and C. V. Damásio. Tabling for P-log probabilistic query evaluation. In *New Trends in Artificial Intelligence, Proceedings of 14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*, pages 115–126, 2009.

[40] T. A. Han, A. Saptawijaya, and L. M. Pereira. Moral reasoning under uncertainty. In *Proceedings of the 18th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18)*, pages 212–227. Springer LNAI 7180, 2012.

[41] The Anh Han. Evolution prospection with intention recognition via computational logic. Master's thesis, Technical University of Dresden, Germany and Universidade Nova de Lisbon, Portugal, June 2009.

[42] The Anh Han. *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation: From Artificial Intelligence Techniques to Evolutionary Game Theory Models*, volume 9. Springer SAPERE series, 2013.

[43] The Anh Han and Luís Moniz Pereira. Intention-based decision making and its applications. In H. Guesgen and S. Marsland, editors, *Human Behavior Recognition Technologies: Intelligent Applications for Monitoring and Security*, pages 174–211. IGI Global, 2013.

[44] C. Heinze. *Modeling Intention Recognition for Intelligent Agent Systems*. PhD thesis, The University of Melbourne, Australia, 2003.

[45] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[46] Jun Hong. Goal recognition through goal graph analysis. *Journal of Artificial Intelligence Research*, 15:1–30, 2001.

[47] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In G. F. Cooper, , and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 256–265. Morgan Kaufmann, 1998.

[48] M. Huber and R. Simpson. Recognizing the plans of screen reader users. In *In Workshop*

on Modeling Other Agents from Observations (MOO2004), 2004.

[49] H. Kautz and J.F. Allen. Generalized plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI'1986)*, pages 32–38. AAAI, 1986.

[50] N. Keng and D. Y. Y. Yun. A planning / scheduling methodology for the constrained resources problem. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 999–1003, 1989.

[51] K.B. Laskey. MEBN: A language for first-order Bayesian knowledge bases. *Artificial Intelligence*, 172(2-3):140 – 178, 2008.

[52] K.B. Laskey and S.M. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'97)*. Morgan Kaufmann, 1997.

[53] N. Lesh. *Scalable and Adaptive Goal Recognition*. PhD thesis, University of Washington, 1998.

[54] Linux-Plan-Corpus. http://www.cs.rochester.edu/research/cisd/resources/linux-plan/. Last access: November 21 2010.

[55] O. Madani, H. Bui, and E. Yeh. Efficient online learning and prediction of users' desktop actions. In *Proceedings of the 21nd international joint conference on Artificial intelligence (IJCAI'2009)*, pages 1457–1462, 2009.

[56] Suzanne M. Mahoney and Kathryn B. Laskey. Network engineering for complex belief networks. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 389–396. Morgan Kaufmann, 1996.

[57] Manfred Milinski, Dirk Semmann, Hans J. Krambeck, and Jochem Marotzke. Stabilizing the Earth's climate is not a losing game: Supporting evidence from public goods experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 103:3994–3998, 2006.

[58] T. J. H. Morgan, L. E. Rendell, M. Ehn, W. Hoppitt, and K. N. Laland. The evolutionary basis of human social learning. *Proceedings of the Royal Society B: Biological Sciences*, July 2011.

[59] S. Natarajan, P. Tadepalli, T. G. Dietterich, and A. Fern. Learning first-order probabilistic models with combining rules. *Annals of Mathematics and Artificial Intelligence*, 54:223–256, 2008.

[60] M. A. Nowak and K. Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355:250–253, 1992.

[61] Martin A. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in prisoner's dilemma. *Nature*, 364:56–58, 1993.

[62] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[63] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[64] L. M. Pereira and T. A. Han. Evolution prospection in decision making. *Intelligent Decision Technologies*, 3(3):157–171, 2009.

[65] L. M. Pereira and T. A. Han. Intention recognition via causal Bayes networks plus plan generation. In *Proceedings of 14th Portuguese International Conference on Artificial Intelligence (EPIA'09)*, pages 138–149. Springer LNAI 5816, October 2009.

[66] L. M. Pereira and T. A. Han. Elder care via intention recognition and evolution prospection. In *select extended papers from the 18th International Conference on Applications of Declarative Programming and Knowledge Management (INAP'09)*, pages 170–187. Springer, LNAI 6547, 2011.

[67] L. M. Pereira and T. A. Han. Intention recognition with evolution prospection and causal Bayesian networks. In *Computational Intelligence for Engineering Systems 3: Emergent Applications*, pages 1–33. Springer, 2011.

[68] A. Pfeffer, D. Koller, B. Milch, and K. T. Takusagawa. Pook: A system for probabilistic object-oriented knowledge representation. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 541–550, 1999.

[69] David V. Pynadath and Michael P. Wellman. Accounting for context in plan recognition, with application to traffic monitoring. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 472–481, 1995.

[70] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI'2010)*, pages 1121–1126, 2010.

[71] L. Rendell, R. Boyd, D. Cownden, M. Enquist, K. Eriksson, M. W. Feldman, L. Fogarty, S. Ghirlanda, T. Lillicrap, and K. N. Laland. Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213, 2010.

[72] Peter J. Richerdson and Robert Boyd. *Not By Genes Alone: How Culture Transforms Human Evolution*. The University of Chicago Press, 2006.

[73] Olivier Roy. *Thinking before Acting: Intentions, Logic, Rational Choice*. PhD thesis, ILLC Dissertation Series DS-2008-03, Amsterdam., 2009.

[74] F. Sadri. Logic-based approaches to intention recognition. In *Handbook of Research on Ambient Intelligence: Trends and Perspectives*, pages 346–375. 2011.

[75] Francisco C. Santos and Jorge M. Pacheco. Risk of collective failure provides an escape from the tragedy of the commons. *Proceedings of the National Academy of Sciences of the United States of America*, 108(26):10421–10425, 2011.

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

[77] O. C. Schrempf, D. Albrecht, and U. D. Hanebeck. Tractable probabilistic models for intention recognition based on expert knowledge. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 1429–1434. IEEE, 2007.

[78] K. Sigmund, H. De Silva, A. Traulsen, and C. Hauert. Social learning promotes institutions for governing the commons. *Nature*, 466:7308, 2010.

[79] Karl Sigmund. *The Calculus of Selfishness*. Princeton University Press, 2010.

[80] Munindar P. Singh. Intentions, commitments and rationality. In *In 13th Annual Conference of the Cognitive Science Society*, pages 493–498, 1991.

[81] Parag Singla and Raymond J. Mooney. Abductive markov logic for plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI'2011)*, pages 1069–1075, 2011.

[82] Sampath Srinivas. A generalization of the noisy-or model. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 208–218. Morgan Kaufmann, 1993.

[83] Karim A. Tahboub. Intelligent human-machine interaction based on dynamic Bayesian networks probabilistic intention recognition. *J. Intell. Robotics Syst.*, 45:31–52, January 2006.

[84] A. Traulsen, M. A. Nowak, and J. M. Pacheco. Stochastic dynamics of invasion and fixation. *Phys. Rev. E*, 74:11909, 2006.

[85] A. Traulsen, D. Semmann, R. D. Sommerfeld, H-J. Krambeck, and M. Milinski. Human strategy updating in evolutionary games. *Proceedings of the National Academy of Sciences of the United States of America*, 107:2962–2966, 2010.

[86] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine Recognition of Human Activities: A Survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[87] Karl Tuyls and Simon Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171:406–416, May 2007.

[88] R. Wilensky. *Planning and Understanding.* Addison-Wesley, 1983.

[89] Y. P. Xiang and K. L. Poh. Constructing Bayesian network in a changing world. In *Proceedings of AAAI Spring Symposium on Challenges to Decision Support in a Changing World*, pages 108–113. AAAI, 2005.

[90] XSB. The XSB system version 3.2 vol. 2: Libraries, interfaces and packages, March 2009.