Anytime Intention Recognition via Incremental Bayesian Network Reconstruction

Han The Anh and **Luís Moniz Pereira**
Centro de Inteligência Artificial (CENTRIA)
Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
$h.anh@fct.unl.pt, lmp@di.fct.unl.pt$

Abstract

This paper presents an anytime algorithm for incremental intention recognition in a changing world. The algorithm is performed by dynamically constructing the intention recognition model on top of a prior domain knowledge base. The model is occasionally reconfigured by situating itself in the changing world and removing newly found out irrelevant intentions. We also discuss some approaches to knowledge base representation for supporting situation-dependent model construction. Reconfigurable Bayesian Networks are employed to produce the intention recognition model.

Introduction

We propose a method for intention recognition (IR) in a dynamic, real-world environment. An important aspect of intentions is their pointing to the future, i.e. if we intend something now, we mean to execute a course of actions to achieve something in the future (Bratman 1987). 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. An IR method should take into account these changes, and may need to reevaluate the IR model depending on some time limit.

We use Bayesian Networks (BN) as the IR model. The flexibility of BNs for representing probabilistic dependencies and the efficiency of inference methods for BN has made them an extremely powerful tool for problem solving under uncertainty (Pearl 1988; 2000).

This paper presents a knowledge representation method to support incremental BN construction for IR during runtime, from a prior domain knowledge base. As more actions are observed, a new BN is constructed reinforcing some intentions while ruling out others. This method allows domain experts to specify knowledge in terms of BN fragments, linking new actions to ongoing intentions.

In order to proactively provide contextually appropriate help to users, assisting systems need the ability to recognize their intentions in a timely manner, given the observed actions. Moreover, the IR algorithm should be anytime, i.e. the IR decision can be made at any moment and can be refined if more time is allotted; e.g. in interface agent domain (Armentano and Amandi 2009). In this paper, we employ an anytime BN inference algorithm to design an anytime IR algorithm. There has been an extensive range of research regarding this kind of approximate BN inference algorithms (Ramos and Cozman 2005; Guo and Hsu 2002).

In the next section we present and justify a BN model for IR. Then, a method for incremental BN model construction during runtime is presented.

Bayesian Network for Intention Recognition

In (Pereira and Han 2009), a Causal BN structure for intention recognition is presented and justified based on Heinze's intentional model (Heinze 2003). In the sequel some background knowledge and the structure of the network is recalled. In this work we do not need the network causal property; hence, only background of the naive BNs is recalled.

Definition 1 *A Bayes Network 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 assumption of conditional independence, saying that variables are independent of their non-effects conditional on their direct causes (Pearl 2000).*

Definition 2 *Let G be a dag that represents causal relations between its nodes. For two nodes A and B of G, if there is an edge from A to B (i.e. A is a direct cause of B), A is called a parent of B, and B is a child of A. The set of parent nodes of a node A is denoted by* parents(A). *Ancestor nodes of A are parents of A or parents of some ancestor nodes of A. If node A has no parents ($parents(A) = \emptyset$), it is called a* top node. *If A has no child, it is called a* bottom node. *The nodes which are neither top nor bottom are said* intermediate. *If the value of a node is observed, the node is said to be an* evidence node.

In a BN, associated with each intermediate node of its dag is a specification of the distribution of its variable, say A, conditioned on its parents in the graph, i.e. $P(A|parents(A))$ is specified. For a top node, the unconditional distribution of the variable 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 $P(X_1, ..., X_N) = \prod_{i=1}^{N} P(X_i|parents(X_i))$, 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 in the dag, say $O = \{O_1, ..., O_m\} \subset V$. We can determine the conditional probability of a variable X given the observed value of evidence nodes by using the conditional probability formula

$$P(X|O) = \frac{P(X,O)}{P(O)} = \frac{P(X, O_1, ..., O_m)}{P(O_1, ..., O_m)} \quad (1)$$

where the numerator and denominator are computed by summing up the joint probabilities over all absent variables with respect to V (see (Pereira and Han 2009) for details).

In short, to define a BN, one needs to specify the structure of the network, its CPD and the prior probability distribution of the top nodes.

Network Structure for Intention Recognition

The first phase of the IR system is to find out how likely each conceivable intention is, based on current observations such as observed actions of the intending agent or the effects of its actions (those of actions actually observed, or those of actions whose direct observation was missed) had in the environment. A conceivable intention is the one having causal relations to all current observations. It is brought out by using a BN with nodes standing for binary random variables[1] (i.e. receiving values *true* or *false*) that represent causes, intentions, actions and effects.

Intentions are represented by intermediate nodes whose ancestor nodes stand for causes that may give rise to them. Intuitively, we extend Heinze's tri-level model (Heinze 2003) (Figure 1) with a so-called pre-intentional level that describes the causes of intentions, used to estimate prior probabilities of the intentions. However, if these prior probabilities can be specified without considering the causes, intentions are represented by top nodes. These reflect the problem context or the intending agent's mental state.

Observed actions are represented as children of the intentions that causally affect them. Observable effects are represented as bottom nodes. They can be children of observed action nodes or of some unobserved actions that might cause those effects—which are added as children of intention nodes.

[1]Note that here intentions are considered at the most abstract level w.r.t. the IR problem. For example, observing an action of taking a car out of a garage, at first one would want to recognize whether or not the agent has an intention of "traveling"—rather than more concrete instances immediately (e.g. go to beach, mountain, etc.). This approach complies with Bratman's opinion that one typically settles on plans that are partial and them fills them in as need be and as time goes by (Bratman 1987). Moreover, as we shall see, there is no restriction to consider intentions as sets of intention instances; the random variables representing such intentions would then receive values in the corresponding set of instances. In this paper, we keep at this level of abstract representation for simplicity of formalization.

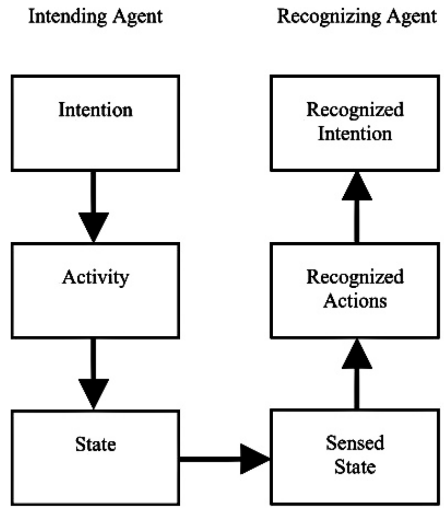

Figure 1: Heinze's tri-level decompositional model of intentional behavior of the intending agent: Intentional level; Activity level; and State level. Intention recognition is the reversal of this process.

The causal relations among nodes of the BNs (e.g. which causes give rise to an intention, which intentions trigger an action, which actions have an effect), as well as their CPD and the distribution of the top nodes, are specified by domain experts. However, they might be learnt mechanically. By using equation (1) the conditional probabilities of each intention on current observations can be determined, X being an intention and O being the set of current observations.

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. For example, in the Elder Care domain, different elders will have different 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 (Pereira and Han 2010). However, consulting the domain expert to manually change the BN w.r.t. each situation is also very costly or unfeasible.

In (Pereira and Han 2009) is provided a way to construct situation-sensitive BNs, i.e. ones that change according to the given situation. It uses Logic Programming (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 (Baral, Gelfond, and Rushton 2009; Han, Ramli, and Damásio 2008), which supports coherent updates. The LP techniques employed for this are deduction with top-down procedure (XSB-Prolog) (XSB 2009) (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.

In a BN for intention recognition, we assume that only prior probabilities of the top nodes (causes) 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. They may change in the long-term periods, but just occasionally. In short, for a given situation at hand, a BN for intention recognition can be situated in it by re-computing the prior probabilities of the cause nodes in accordance with the situation.

Based on this discussion, in the next sections we show an anytime IR method by incrementally constructing a BN as new actions are observed.

Incremental Bayesian Network Construction for Intention Recognition

For simplicity, the following formally defines a simpler version of a BN for intention recognition where we do not include the state level (the effects of actions).

Definition 3 (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 nodes, intention nodes and action nodes, respectively. They stand for binary random variables.*
- 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* $pa(A) = Is \;\; \forall A \in As$*. This means that* Cs *are the top nodes; cause nodes have connections only to intention nodes; and each intention node connects to all action nodes.*
- *CPD tables of the intention and action nodes and prior probabilities of the cause nodes 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, for all* $X \in V_W$ *where* $V_W = Cs \cup Is \cup As$*.*

Furthermore, it is required that $Cs = \bigcup_{I \in Is} pa(I)$*, i.e. there is no isolated cause node in W.*

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 of IRBNs designed for single actions. We refer to them as the unit BNs for intention recognition.

Definition 4 (Unit IRBN) *The Bayesian network for intention recognition for an action A, denoted by* `irBN(A)`*, is an IRBN where its set of intentions refers to a single action A. We denote by* P_A *the probability distribution in* `irBN(A)`*.*

Next we stipulate a reasonable set of assumptions for a domain knowledge base.

Definition 5 (Knowledge Base) *The domain knowledge base* KB *consists of a set of actions* AS *and a set of unit IRBNs for every action in* AS*, satisfying that*

- *An intention I has the same set of parents (causes) and CPD table in all the unit IRBNs that it belongs to. Let* $C(I)$ *denote the set of parents of I and* $P_{KB}(I|C(I))$ *define its CPD table.*
- *A cause C has the same prior probability distribution in all the unit IRBNs that it belongs to, denoted by* $P_{KB}(C)$*.*

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

Definition 6 (Project of CPD Table) *Let T be a CPD table defining* $P(X|V)$*, the probability of a random variable X conditional on a set of random variables V. The project of T on a strict subset* V' *of V* ($V' \subset V$) *according to an assignment* u *to the variables in* $U = V \setminus V'$ *is a CPD table defining* $P(X|V', U = u)$*, i.e. the part of T for V that corresponds to* $U = u$*.*

If the variables of V are binary, we denote by $\mathtt{proj}(\mathtt{T}, \mathtt{V}')$ *the project of T on* V' *according the assignment of all variables in U to false. Furthermore, from now on if we do not mention the assignment, we mean implicitly the assignment of all variables in U to false.*

Now let us define how to combine an IRBN with a set of unit IRBNs. This occurs when a set of new actions are observed, and we would like to combine the unit IRBNs for those actions with the current IRBN to get a new IRBN.

In the formalism of this paper we exploit an assumption that the intending agent has only one most likely intention at a given time, to be discerned by the IRBN. Our approach can be naturally extended to consider the case of having multiple intentions at the same time; however, we will not formalize that here. The extension will be further discussed in the future work section.

Intuitively, conceivable intentions are those that give rise to all current observable actions; thus, the set of conceivable intentions in the obtained IRBN is the intersection of the sets of such intentions of the IRBN and of the unit IRBNs of the newly observed actions. The CPD table of an action in the obtained IRBN is given by the project of the corresponding CPD table in its unit IRBN on the new set of intentions.

We start by defining how to combine a set of unit IRBNs.

Definition 7 (Combination of Unit IRBNs) *Let* $O = \{A_1, ..., A_n\}$ ($n \geq 0$) *be a set of actions such that* $O \subseteq AS$*. Let* $irBN(A_i) = \langle \{Cs_i, Is_i, \{A_i\}\}, pa_i, P_{A_i} \rangle$ *be the unit IRBN for action* A_i ($1 \leq i \leq n$)*. The BN for intention recognition for O, denoted by* `irBN(O)`*, is the triple* $\langle \{Cs_O, Is_O, O\}, pa_O, P_O\} \rangle$ *where*

- $Is_O = \bigcap_{i=1}^{n} Is_i$*;* $Cs_O = \bigcup_{I \in Is_O} C(I)$*;*
- $pa_O(C) = \emptyset \;\; \forall C \in Cs_O$*;* $pa_O(I) = C(I) \;\; \forall I \in Is_O$*; and* $pa_O(A_i) = Is_O \;\; \forall i : 1 \leq i \leq n$*;*
- $P_O(C) = P_{KB}(C) \;\; \forall C \in Cs_O$*;* $P_O(I|pa_O(I)) = P_{KB}(I|C(I)) \;\; \forall I \in Is_O$*; and for* $1 \leq i \leq n$*,* $P_O(A_i|pa_O(A_i))$ *is defined by* $proj(T, Is_O)$ *where T is the CPD table for* A_i *in* `irBN(A_i)`*, i.e. defined by* $P_{A_i}(A_i|pa_i(A_i))$*.*

It is easy to see that `irBN(O)` is an IRBN (Definition 3). Now let us define how to combine two IRBNs.

Definition 8 (Combination of IRBNs) *Let* $W_1 = \langle \{Cs_1, Is_1, As_1\}, pa_1, P_1 \rangle$ *and* $W_2 = \langle \{Cs_2, Is_2, As_2\}, pa_2, P_2 \rangle$ *be two IRBNs. The combination of these two IRBNs is an IRBN, denoted by* $\mathtt{comb}(\mathtt{W}_1, \mathtt{W}_2) = \langle \{Cs, Is, As\}, pa, P_W \rangle$*, defined as follows*

- $As = As_1 \cup As_2;\ Is = Is_1 \cap Is_2;\ Cs = \bigcup_{I \in Is} C(I);$
- $pa(C) = \emptyset\ \ \forall C \in Cs;\ \ pa(I) = C(I)\ \ \forall I \in Is;$ *and* $pa(A) = Is\ \ \forall A \in As;$
- $P_W(C) = P_{KB}(C)\ \ \forall C \in Cs;\ P_W(I|pa(I)) = P_{KB}(I|C(I))\ \ \forall I \in Is;$ *and for each* $A \in As$, $P_W(A|pa(A))$ *is defined by the CPD table proj*(T, Is) *where T is the CPD table for A in* `irBN(A)`, *i.e. defined by* $P_A(A|pa(A))$.

When some intentions are found irrelevant—e.g. because they are much unlikely[2]—those intentions should be taken out of the IRBN. This is enacted by considering them as completely false and employing a project operator.

Definition 9 (Remove Intentions from IRBN) *Let* $W = \langle\{Cs, Is, As\}, pa, P_W\rangle$ *be an IRBN and* $R \subset Is$ *a strict subset of Is. The result of removing the set of intentions R from W is an IRBN, denoted by* `remove(W,R)` $= \langle\{Cs_R, Is_R, As_R\}, pa_R, P_R\rangle$, *and defined as follows*

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

In a given situation, an IRBN is situated by recomputing the prior probabilities of the top nodes.

Definition 10 (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 (also in (Pereira and Han 2009)).*

Formally, the situate operator is defined by `situate(W,SIT)` $= \langle\{Cs, Is, As\}, pa, P_S\rangle$, *where*

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

Anytime Intention Recognition Algorithms

Criteria let decide if the BN should be reconfigured to take into account the situation at hand. We use here a simple criterion, saying that if there is a "salient" intention currently in the BN, it should be reconfigured—i.e. be situated according to the latest information about the situation at hand; otherwise, the BN remains the same. The property "salient" reads differently in different application domains, and it is up to the domain expert to design its specification. E.g., in security domain, "salient" may read *dangerous* (intrusion intention; committing suicide intention (of an assisted person), etc.).

[2] 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.

Algorithm 1. Let KB be the prior domain knowledge base and AS its set of actions. Repeat the following steps until there is only one intention remaining in the IRBN or some time limit is reached; in the latter case, the most likely intention in the previous cycle is the final result.

1. Let $O \subseteq AS$ be the set current observed actions. Combine the current IRBN W with *irBN(O)* we obtain $W' =$ `comb(W, irBN(O))`. If O is the set of initially observed actions, let $W' = irBN(O)$.
2. If there is an "salient" intention in W', *situate* it according to the situation at hand *curSIT*: `situate(W', curSIT)` $= W''$; otherwise, the IRBN remains the same: $W'' = W'$.
3. Compute the likelihood of each intention in W'', conditional on the set of current observed actions in W''. Remove the intentions which are much less likely than the others (following Definition 9).

If an observed action makes the set of conceivable intentions empty, the action is considered irrelevant to the sought for intention, and discarded. There may be several intentions being pursued by the agent, but that issue is not further examined here yet.

At any cycle, if the likelihood of all the intentions are very small (say, smaller than a given threshold), one could say that the sought for intention is abandoned. This is because the causes and actions do not support or force the intending agent to keep pursuing his initial intention anymore.

In Algorithm 1, a standard inference algorithm is used for computing conditional probabilities based on Formula (1). If the time limit is reached within the first cycle, i.e. when the initial set of actions are observed, the algorithm cannot provide any IR decision. Hence, in the sequel we use an anytime BN inference algorithm (Ramos and Cozman 2005) to provide a truly anytime IR algorithm. There are a number of such BN inference algorithms in the literature. However, the question which one is the most appropriate for the presented IRBN structure is beyond the scope of this paper.

Definition 11 (Anytime Algorithm) *An algorithm is* anytime *if it can produce a solution in a given time* T *and the quality of the solutions improves with time after* T.

Accordingly, Algorithm 1 is considered anytime only if the first cycle is guaranteed to pass. It mostly depends on whether the BN inference can be done within the time limit. Using an anytime inference algorithm can solve this problem. Although the precision might decrease, we accept it due to time limitation.

Algorithm 2 (Anytime). The only difference is in step 3. An anytime BN inference algorithm is used to compute the likelihood of intentions. Stop when the time limit is reached.

Revising Knowledge Base Representation

The knowledge base presented above contains a single BN for a particular action, and in the BN all conceivable intentions that may give rise to the action are included. The set of intentions in each BN is usually very big.

The set of conceivable intentions for an action should depend on the situation in which the action is observed. We propose two methods for revising the knowledge base representation to tackle this problem. The first uses a set of conditional unit IRBNs for each action, and provides a criterion to choose an appropriate one depending on the situation at hand. In contrast, the second method keeps the same representation as before, and uses common sense reasoning to compute the set of conceivable intentions of an action in the given situation; then, a `remove` operator is performed.

Conditional Unit IRBNs

We provide a set of unit IRBNs for each action, and a criterion for choosing what is the appropriate one depending on the situation at hand. Each unit IRBN in the set is accompanied by a precondition, and the criterion must satisfy that in each situation only one unit IRBN is for an action.

This method would perform efficiently by encoding a situation as a logic program, then w.r.t. which the preconditions are evaluated. However, designing a BN for each situation is very costly since the set of of situations needing to be considered may be very large. Moreover, it is difficult if not impossible to specify all situations at the beginning.

The following method handles this issue in a more constructive way.

Situation-sensitive Intentions

Undoubtedly, whether an intention may give rise to a particular action or not should depend on the situation in which the action is observed. Given some situation at hand, common-sense reasoning can be employed for that purpose. In the sequel LP techniques are used for common sense reasoning.

The prior domain knowledge base KB is accompanied by a logic program P_{KB} in order to help decide which intentions in a unit IRBN are conceivable in a given situation. Let $AS = \{A_1, ..., A_N\}$ $(N > 1)$ be the set of actions of KB and $BNs = \{W_1, ..., W_N\}$, where $W_i = \langle\{Cs_i, Is_i, As_i\}, pa_i, P_i\rangle$ $(1 \leq i \leq N)$, be the set of unit IRBNs of 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. Thus, for $1 \leq i \leq N$ and $I \in Is_i$, P_{KB} contains the following rule:

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

Furthermore, for each $I \in \bigcup_{i=1}^{N} Is_i$, P_{KB} contains two rules: $expect(I) \leftarrow Cond_1.$ $expect\_not(I) \leftarrow Cond_2.$

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_i $(1 \leq i \leq N)$ is observed. The current situation is encoded by a logic program SIT. 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 to find all true *conceivable/1* atoms of the program $P_{KB} \cup SIT \cup \{A_i \leftarrow\}$.

Suppose O is the set of obtained conceivable intentions. Then, the IRBN obtained by removing the other intentions from W_i, i.e. $\mathtt{remove}(\mathtt{W_i}, \mathtt{Is_i} \setminus \mathtt{O})$, is used for IR.

Example 1 (Elder Care) *An elderly person stays alone in his apartment. An IR 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:* $\mathtt{Is} = \{\mathtt{book}, \mathtt{water}, \mathtt{weapon}, \mathtt{lightSwitch}\}$. *The accompanying logic program P_{KB} contains the following rules, for each $I \in Is$:* $\mathtt{conceivable(I)} \leftarrow \mathtt{look},\ \mathtt{expect(I)},\ \mathtt{not\ expect\_not(I)}$. *Suppose in P_{KB} the expectation and counter-expectation rules for these intentions are*

1. $expect(book).$ $\quad expect\_not(book) \leftarrow light\_off.$
   $expect\_not(book) \leftarrow burglar\_alarm\_ring.$
2. $expect(water).$ $\quad expect\_not(water) \leftarrow light\_off.$
   $expect\_not(water) \leftarrow burglar\_alarm\_ring.$
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 part 1 say the intention of looking for a book is always expected except when the light is off or the burglar alarm is ringing.

If at the moment, light is off ($SIT = \{light\_off \leftarrow\}$), then $\mathtt{conceivable(light\_switch)}$ *is the only true* $\mathtt{conceivable/1}$ *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.*

Now suppose light is on, the tv is not on, and the burglar alarm is not ringing. There are three conceivable intentions: book, water *and* lightSwitch*. Then we need to remove the intention* weapon *from the unit IRBN.*

If light is on, tv is not on, and the burglar alarm is ringing, the are conceivable intentions: weapon *and* lightSwitch*.*

Complexity Assessments

Let *MI, MC* be the maximal number of intentions and maximal number of causes in a unit IRBN of KB, respectively, and *NA* be the number of actions in KB (i.e. $|AS|$). Let $M = MI + MC + NA$. The complexity will be evaluated in terms of M. First we evaluate the worst-case complexity of the operators: project, combine, remove, situate.

- It is easy see that the project $\mathtt{proj(T, V')}$ in Definition 6 can be done in $2^{|V'|}$. That is to create a new CPD table from the original table. However, in a real implementation, it can be done by simply setting a pointer to the original table. Thus, the complexity of this operator can be considered as constant.
- The operator `comb` combining two IRBNs in Definition 8 can be done in linear time, i.e. $O(M)$; the same complexity for the `remove` operator in Definition 9.
- The complexity of the `situate` operator in Definition 10 is mainly from that of inferring the prior probabilities of the top nodes according to the situation at hand. If encoding the situation as a logic program and using the top-down querying procedure of XSB-Prolog (as done in (Pereira and Han 2009)) these inferences can be done in polynomial time (XSB 2009).

In short, all these operators have polynomial time complexity. Thus, the complexity of the algorithms depend mostly on the BN inference algorithms (in step 3). In algorithm 1, the exact probabilistic inference method based on formula (1) is used. Like in the general case of BNs, the inference in IRBNs is exponential ($O(2^M)$).

As we see, the complexity of the algorithms strongly depends on the size of the IRBNs—which depends on the size of its set of intention nodes. Note the size of the set of causes proportionally depends on that of the set of intentions. Hence, the two discussed methods for revising the domain knowledge base representation, which reduce the set of intentions of IRBNs, can help to enable to considerably improve the performance.

Related Work

Bayesian networks have been one of the most successful models applied for intention or plan recognition problem. The most important works[3] were proposed by Charniak and Goldman, e.g. in (Charniak and Goldman 1993), and more recently, by Geib and Goldman, e.g. in (Geib and Goldman 2009). Depending on the structure of plan libraries, they employed some knowledge-based model construction to build BNs from the library, and then infer the posterior probability of explanations (for the set of observed actions).

Different from our IR model, in those works, the causes of intentions (or root goals as they dubbed) do not figure in the model; the prior probabilities of intentions are assumed to be given in the plan library. As discussed previously, we believe that this assumption is not reasonable because those prior probabilities should depend on the situation at hand, which are captured by the causes of the intentions. This way, our model can appropriately explain the abandonment of intentions—when the causes do not support or force the intending agent to hold those intentions anymore.

Furthermore, in their approaches, the intentions are not envisaged depending on the situation at hand. Thus, the constructed BN model is usually quite large.

Lastly, to our knowledge, we make here the first attempt to design an anytime algorithm for intention recognition.

Conclusion and Future Work

We have presented an anytime algorithm for incremental intention recognition in a changing world. The algorithm is performed by dynamically constructing a BN model for intention recognition from a prior domain knowledge base consisting of BN units. The model is occasionally updated w.r.t. the situation at hand when required.

We have also discussed some methods to knowledge base representation for supporting situation-dependent model construction, which enable to reduce the size of the model—and consequently, the complexity of the algorithm. Our next step is to improve these methods to allow representing the knowledge base in terms of easily maintained BN fragments; e.g, a BN fragment consists of a single intention and a single action. The fragments are chosen depending on the situation at hand, and combined by employing a wide range of BN combination methods, e.g. in (Laskey and Mahoney 1997).

Another future direction is to consider the case the intending agent may pursue multiple intentions simultaneously. This can be achieved simply by creating a new IR process whenever a newly observed action is not explained by the current intentions. Now, a set of IR processes is maintained and reconfigured as observation of new actions emerge. The result is a set of most likely intentions for each IR process.

[3] Here we only consider the works using naive BN models; other BN models such as dynamic BNs are not yet discussed.

References

Armentano, M. G., and Amandi, A. 2009. Recognition of user intentions for interface agents with variable order markov models. In *UMAP'09*, 173–184. LNCS 5535.

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

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

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

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

Guo, H., and Hsu, W. 2002. A survey of algorithms for real-time bayesian network inference. In *Joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems*.

Han, T. A.; Ramli, C. K.; and Damásio, C. V. 2008. An implementation of extended p-log using xasp. In *ICLP08*, 739–743. Springer LNCS 5366.

Heinze, C. 2003. *Modeling Intention Recognition for Intelligent Agent Systems*. Ph.D. Dissertation, The University of Melbourne, Australia.

Laskey, K., and Mahoney, S. 1997. Network fragments: Representing knowledge for constructing probabilistic models. In *UAI*. Morgan Kaufmann.

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

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge U.P.

Pereira, L. M., and Han, T. A. 2009. Intention recognition via causal bayes networks plus plan generation. In *Procs. 14th Portuguese Intl. Conf. on Artificial Intelligence (EPIA'09)*, 138–149. Springer LNAI 5816.

Pereira, L. M., and Han, T. A. 2010. Intention recognition with evolution prospection and causal bayesian networks. In *Computational Intelligence for Engineering Systems 3: Emergent Applications*. Springer. (forthcoming).

Ramos, F. T., and Cozman, F. G. 2005. Anytime anyspace probabilistic inference. *International Journal of Approximate Reasoning* 38(1):53 – 80.

XSB. 2009. The xsb system version 3.2 vol. 2: Libraries, interfaces and packages.