Multi-Agent Intention Recognition using Logical Hidden Semi-Markov Models

Shi-guang Yue, Ya-bing Zha, Quan-jun Yin and Long Qin
College of Informaiton System and Management, National University of Defense Technology, Changsha, HN, China

Keywords: Multi-Agent Intention Recognition, Probabilistic Graphical Models, First Order Logic, Duration Modeling, Logical Semi-Markov Models.

Abstract: Intention recognition (IR) is significant for creating humanlike and intellectual agents in simulation systems. Previous widely used probabilistic graphical methods such as hidden Markov models (HMMs) cannot handle unstructural data, so logical hidden Markov models (LHMMs) are proposed by combining HMMs and first order logic. Logical hidden semi-Markov models (LHSMMs) further extend LHMMs by modeling duration of hidden states explicitly and relax the Markov assumption. In this paper, LHSMMs are used in multi-agent intention recognition (MAIR), which identifies not only intentions of every agent but also working modes of the team considering cooperation. Logical predicates and connectives are used to present the working mode; conditional transition probabilities and changeable instances alphabet depending on available observations are introduced; and inference process based on the logical forward algorithm with duration is given. A simple game "Killing monsters" is also designed to evaluate the performance of LHSMMs with its graphical representation depicted to describe activities in the game. The simulation results show that, LHSMMs can get reliable results of recognizing working modes and smoother probability curves than LHMMs. Our models can even recognize destinations of the agent in advance by making use of the cooperation information.

1 INTRODUCTION

Intention recognition (IR) in simulation is to identify the specific goals that an agent or agents are attempting to achieve (Sadri, 2011). Since goals are always hidden in mind, they can only be inferred by analysing the observed agents' actions and/or the changes in the state (environment) resulting from their actions. IR is significant for creating human-like and intellectual agents in real time strategy games, artificial societies and other simulation systems. Agents who recognize intentions of opponents and/or friends can make counter and/or cooperative decisions efficiently.

As an intersection of psychology and artificial intelligence, the IR problem has attracted many attentions for decades (Schmidt et al., 1978). Hidden Markov models (HMMs), which are special cases of dynamic Bayesian networks (DBNs), have been widely used to recognize intentions in different scenarios. For example, Zouaoui-Elloumi etc. built an autonomous system to detect suspicious ship in a port based on HMMs (Zouaoui-Elloumi et al., 2010). Dereszynski etc. learnt probabilistic models of high-level strategic behaviour and recognized the adversarial strategies in a real-time strategy game (Dereszynski et al., 2011).

One problem of HMMs is that the Markov assumption cannot always been satisfied. Thus, some refined models are proposed by modelling duration and transition of hidden states more accurately, or introducing hierarchical structures. For example, hidden semi-Markov models (HSMMs) improve the recognition performance by modeling duration explicitly (Yu, 2010). They have been used to infer complex agent motions from partial trajectory observations in the IR domain (Southey et al., 2007). The typical refined hierarchical models include hierarchical HMMs (Fine et al., 1998) and abstract HMMs (Bui et al., 2002). Thi Duong et al. further proposed Coxian switching hidden semi-Markov model, which both built a hierarchical structure and introduced Coxian distribution modeling the duration of states to recognize human activities of daily living (Duong et al., 2009).

Yue S., Zha Y., Yin Q. and Qin L..
Multi-Agent Intention Recognition using Logical Hidden Semi-Markov Models.
DOI: 10.5220/0005036707010708
In *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2014), pages 701-708
ISBN: 978-989-758-038-3

Another problem of HMMs is that they are actually propositional, which means they handle only sequences of unstructured symbols. Therefore, Kersting et al combined HMMs and first order logic and proposed Logical hidden Markov models, which belong to statistical relational learning methods (Kersting et al., 2006). Comparing with HMMs, LHMMs can infer complex relations and have fewer parameters by adding instantiation process. However, they do not consider relaxing Markov assumption, which will lead to a similar performance decline when long-term dependences between hidden states exist as HMMs. Thus, we presented logical hidden semi-Markov models (LHSMMs) by using the idea of HSMMs (Zha et al., 2013). Even though our previous work has proved the achievement of applying LHSMMs in IR, we only consider intentions of one single agent. However, most complex tasks have to be done by one or more teams. Agents always play different roles and cooperate to achieve their common goals. In this case, multi-agent intention recognition (MAIR) problems have to be solved, which means that we need to recognize not only the intentions of every agent, but also the composition and cooperation mode of teams (Pfeffer et al., 2009).

Since LHSMMs inherit advantages of LHMMs and relax the Markov assumption, we will use LHSMMs to solve MAIR problem, as an extension to our previous work. Besides considering intentions of more than one agent, we will refine the previous models further in three aspects. First, logical predicates and connectives are used to present the working modes of the team. Second, Conditional transition probabilities are applied which make transition probabilities depend on previous observations. Third, the alphabet of instances are changeable during the inference, because the number of simulation entities may change because of dying, escaping and reinforcement. The former forward algorithm with duration variable (LFAD) which is the core of the inference is also adjusted according to the modification of models. A simple virtual game "Killing monsters" is designed to evaluate the performance of LHSMMs in MAIR. In this game, two warriors move around and kill monsters on a grid map, they can both act individually and cooperatively. In the simulation, we use lognormal distribution to model the duration of working modes (abstract hidden states), and compute the probabilities of working modes, monsters being chosen at each time. We will show that LHSMMs can correctly recognize working modes and intentions of warriors in the game. Additionally, LHSMMs can even recognize the destinations of the agent in advance by making use of the cooperation information.

The rest of the paper is organized as follows: the next section will gives the formal definition of LHSMMs, the inference algorithm, and a directed graphical representation of a game is presented. Section 3 presents the simulation and results. Subsequently, we have a conclusion and discuss the future work in Section 4. In this paper, we will apply LHSMMs to recognize intentions of two agents and their working modes in a simple virtual game.

2 LOGICAL HIDDEN SEMI-MARKOV MODELS

This section will introduce the LHMMs, which will be used to recognize intentions of agents and the working mode. We will give a formal description of models and the inference process in 3.1 and 3.2 respectively. A multi-agent game is also designed to evaluate the models in 3.3.

2.1 Model Definition

LHSMMs extend LHMMs by modelling the duration of the hidden abstract states just as HSMMs extend HMMs. In this paper, we further refine our former models by redefining the logical alphabet, the selection probability and the transition matrix.

A LHSMM is a five-tuple $Ms = (\mathbf{\Sigma}, \boldsymbol{\mu}, \mathbf{\Delta}, \Upsilon, \mathbf{D})$, the $\mathbf{\Sigma} = \{\Sigma_t\}$ records possible instances for the variables in every abstract state at every time. Since the number of simulation entities may change because of dying, escaping or reinforcement, the $\mathbf{\Sigma}$ depends on observations available up to time t, (Σ_t is the logical alphabet at time t given $O_{1:t} = \{O_1, O_2, \cdots, O_t\}$). $\boldsymbol{\mu} = \{\mu_t\}$ is a selection probabilityset over $\mathbf{\Sigma}$,thus it is also a function of $O_{1:t}$. $\mathbf{\Delta}$ is the transition matrix defining transition probabilities between abstract states. Abstract transition are expressions of the form $p : H \xleftarrow{O} B$ where $p \in [0,1]$, B , H and O are logic sentences which represents hidden states. A θ is a substitution, and $B\theta_B$ is one state of $G_\Sigma(B)$, where $G_\Sigma(B)$ represents the set of all ground or variable-free atoms of B over the alphabet Σ , so are H and O .

We also use the idea of logical transition in Natarajan et al.'s LHHMMs (2008) and let the value of p in $\mathbf{\Delta}$ depend on instances and observations. Figure.1 gives an example of conditional transition probabilities from the abstract states A(X).

$$A(X)\begin{cases} if\ condition\ 1,\ then \begin{cases} p^{1}_{A(X)\to B(Y)} : B(Y) \\ p^{1}_{A(X)\to C(Z)} : C(Z) \end{cases} \\ if\ condition\ 2,\ then \begin{cases} p^{2}_{A(X)\to B(Y)} : B(Y) \\ p^{2}_{A(X)\to C(Z)} : C(Z) \end{cases} \end{cases}$$

Figure 1: An example of conditional transitions from an abstract state.

A(X), B(Y) and C(Z) are abstract states, X, Y and Z are objects in these predicates, $p^{1}_{A(X)\to B(Y)}$, $p^{1}_{A(X)\to C(Z)}$ are the probabilities of states switching from A(X) to B(Y) and C(Z) respectively when condition 1 is satisfied by current observations. The meanings of $p^{2}_{A(X)\to B(Y)}$ and $p^{2}_{A(X)\to C(Z)}$ are similar. Let $B\theta_B \in G_\Sigma(B)$, $H\theta_B\theta_H \in G_\Sigma(H\theta_B)$ and $O\theta_B\theta_H\theta_O \in G_\Sigma(O\theta_B\theta_H)$. Then the model makes a transition from state $B\theta_B$ to $H\theta_B\theta_H$ and emits symbol $O\theta_B\theta_H\theta_O$ with probability

$$p \bullet \mu(H\theta_B\theta_H \mid H\theta_B) \bullet \mu(O\theta_B\theta_H\theta_O \mid O\theta_B\theta_H) \quad (1)$$

Υ is a set of abstract transitions encoding a prior distribution, which has a similar representation as $\mathbf{\Delta}$, except that any B in Υ is the start state with no instance. In addition, any self-transition probability in $\mathbf{\Delta}$ is forced to be 0, and the duration distribution of hidden states will be defined in $\mathbf{D}$. Let **B** be the set of all atoms that occur as body parts of transitions in $\mathbf{\Delta}$, $\mathbf{D}$ define the duration distribution and the corresponding emissions of every atom in **B**. Elements of $\mathbf{D}$ are representations $p_d : B \xleftarrow{\mathbf{O}_d} B$, where $\mathbf{O}^d$ is the set of $d-1$ observations emitted between B (B appears d times in this duration).

In this paper, we will define $p_d(j) = \int_{d-1}^{d} f(x)dx$, $d = 1, 2, \ldots$, where $f(x)$ is the probability density function of the lognormal distribution $\log norm(\mu, \sigma, \theta)$, μ and σ indicate the mean and standard variance of $\log(x-\theta)$ respectively. And they both depend on s_j, which is a instantiated hidden state. θ is the threshold and we make it 0. Since $p_d(j)$ is usually small when d is far from its expectation for s_j, lognormal distribution is more suitable to represent the lasting time of an intention than the geometric distribution.

LHSMMs inherit the graphical representation from LHMMs. Every node in the directed graph represents an abstract state which is a predicate with one or more terms. The transition will begin from the Start node according to Υ. After reaching an abstract state, the node has to be instantiated using $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$. There are three kinds of edges representing transitions: the solid, the dotted and dashed ones. The detailed and formal descriptions of graphical models can be found in reference (Kersting, 2006) Actually, LSHMMs are special cases of LHMMs, when $\mathbf{D}$ only contains geometric distributions.

2.2 Inference

Online intention recognition is a filter problem computing $\Pr(S_t = s_j \mid O_{1:t}, Ms)$, where s_j is the instantiated hidden abstract state at time t. The notion of s_j is similar with nodes in HMMs. It indicates the abstract state and corresponding instantiated results. The *Ms* is the parameter set. Since *Ms* is known in the process of inference, it will be neglected in this part for simplicity. And we can compute the posterior probability by

$$\Pr(S_t = s_j \mid O_{1:t}) = \frac{\Pr(S_t = s_j, O_{1:t})}{\Pr(O_{1:t})} = \frac{\Pr(S_t = s_j, O_{1:t})}{\sum_j \Pr(S_t = s_j, O_{1:t})} \quad (2)$$

For simplicity of notation in the following section, we also denote:

$S_{[t_1,t_2]} = s_j$, s_j starts at time t_1 and ends at t_2 with duration $d = t_1 - t_2 + 1$. This also implies that the state at time $t_1 - 1$ and $t_2 + 1$ can not be s_j.

$S_{t_1,t_2]} = s_j$, s_j starts before time t_1 and ends at t_2 with duration $d > t_1 - t_2 + 1$. This also implies that $S_{t_1-1} = s_j$ the state at time $t_2 + 1$ can not be s_j.

$S_{[t_1,t_2} = s_j$, s_j starts at time t_1 and will not end at t_2 with duration $d > t_1 - t_2 + 1$. This also implies that $S_{t_2+1} = s_j$ the state at time $t_1 - 1$ can not be s_j.

Since we do not know whether s_j will end at time t, we have to compute the $\Pr(S_t = s_j, O_{1:t})$ by

$$\Pr\left(S_{t}=s_j,O_{1:t}\right)=\Pr\left(S_{t]}=s_j,O_{1:t}\right) + \sum_{t'=1}^{t}\sum_{d=t-t'+2}^{\max d}\Pr\left(S_{[t',t'+d-1]}=s_j,O_{1:t}\right) \tag{3}$$

The first part is the probability that the hidden state .the probability of this part will be 0;

In LHSMMs, S_1 is always the start state, whose duration is 1, so we can make $t'>1$ and compute the $\Pr\left(S_{[t',t'+d-1]}=s_j,O_{1:t}\right)$ by

$$\begin{aligned}&\Pr\left(S_{[t',t'+d-1]}=s_j,O_{1:t}\right)\\ &=\sum_i \Pr\left(S_{t'-1]}=s_i,O_{1:t'-1}\right)\\ &\bullet\Pr\left(S_{[t',t}=s_j,O_{t':t}\mid S_{t'-1]}=s_i,O_{1:t'-1}\right)\end{aligned} \tag{4}$$

The meaning of $\Pr\left(S_{t'-1]}=s_i,O_{1:t'-1}\right)$ is similar with $\Pr\left(S_{t]}=s_j,O_{1:t}\right)$, and $\Pr\left(S_{[t',t}=s_j,O_{t':t}\mid S_{t'-1]}=s_i,O_{1:t'-1}\right)$ means that the hidden state switch s_i to s_j at time $t'-1$ and the s_j will last to time $t+1$ at least with observations $O_{t':t}$ .Since both $O_{t':t}$ and $S_{[t',t}=s_j$ are only determined by $S_{t'-1]}=s_i$, we only need to compute $\Pr\left(S_{[t',t}=s_j,O_{t':t}\mid S_{t'-1]}=s_i\right)$ as follows:

$$\begin{aligned}&\Pr\left(S_{[t',t}=s_j,O_{t':t}\mid S_{t'-1]}=s_i\right)\\ &=\sum_{p:H\xleftarrow{O}B} p\bullet\mu_{t'-1}\left(s_j\mid H\sigma_B\right)\bullet\mu_{t'-1}\left(O_{t'}\mid O\sigma_B\sigma_H\right)\\ &\bullet\left(1-\sum_{d'=1}^{d}p_{d'}\left(j\right)\right)\bullet\mu_{t'-1}\left(O_{t'+1:t}\mid s_j\right)\end{aligned} \tag{5}$$

σ_B and σ_H are results of $mgu\left(B,s_i\right)$ and $mgu\left(H,s_j\right)$ respectively, where *mgu* is the most general unifier (MGU) operation in first-order logic. $p\bullet\mu_{t'-1}\left(s_j\mid H\sigma_B\right)\bullet\mu_{t'-1}\left(O_{t'}\mid O\sigma_B\sigma_H\right)$ is the probability that s_i transfers to s_j and emits $O_{t'}$,which is similar with equation (1), $\left(1-\sum_{d'=1}^{d}p_{d'}\left(j\right)\right)\bullet\mu_{t'-1}\left(O_{t'+1:t}\mid s_j\right)$ is the probability that s_j lasts for more than d times and emits $O_{t'+1:t}$ .Then, we will sum all B which satisfies s_i in $\mathbf{\Delta}$. Thus, the key problem is to compute $\Pr\left(S_{t]}=s_j,O_{1:t}\right)$, which can be solved using a logical forward algorithm with duration (LFAD). In the LFAD, the forward variable $\Pr\left(S_{[t-d+1,t]}=s_j,O_{1:t}\right)$ is represented as $\alpha_t\left(i,d\right)$, and $p_{j,d}$ indicates the probability of duration d for s_j . The pseudo-code of LFAD is given in Figure 2. After using LFAD, $\Pr\left(S_{t]}=s_j,O_{1:t}\right)$ can be obtained by $\sum_d \alpha_t\left(d,j\right)$.

2.3 Graphical Representation

To evaluate the performance of LHSMMs using in MAIR, we design a simple game named “Killing the monsters”. There are 2 warriors and 4 monsters on a grid map. The warriors know the locations of monsters, and warriors’ mission is to find the shortest path to the chosen monster, get there and kill it. The map and location information can be

$S_0=\left\{Start\right\}$
    for $t=1,2,\ldots,T$ *do*
        for $d=1:t$ *do*
            $S_t=\phi$
            foreach $s_j\in S_{t-d}$ *do*
                foreach max. *spec.* $p:H\xleftarrow{O}B\in\Delta\cup\gamma$ *s.t.* $\sigma_B=mgu\left(s_j,B\right)$ *exists do*
                    foreach $s_i=H\sigma_B\sigma_H\in G_{\Sigma_{t-d}}\left(H\sigma_B\right)$ *s.t.* $O_{t-d:t-1}$ *unifies with* $O\sigma_B\sigma_H$
                    if $s_i\notin S_t$ *then*
                        $S_t=S_t\cup\left\{s_i\right\}$
                        $\alpha_t\left(i,d\right)=0$
$\alpha_t\left(i,d\right)=\alpha_t\left(i,d\right)+\sum_{d'=1:t-d}\alpha_{t-d}\left(j,d'\right)\bullet p\bullet p_{j,d}\bullet\mu_{t-d}\left(s_i\mid H\sigma_B\right)\bullet\mu_{t-d}\left(O_{t-d:t-1}\mid O\sigma_B\sigma_H\right)$

Figure 2: The pseudo-code of the LFAD.

found in Figure 3

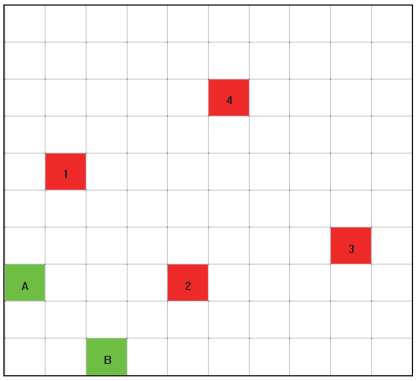

Figure 3: Initial positions in the grid map.

The red points are locations of monsters that will stay there and not move around. The green points are start points of the warriors, in each step, warriors may go into one of the four adjacent places or stay at current one. However, a warrior cannot get through the place where a live monster stands, except that the monster is chosen to be killed by him.

WAGo(MX) and WBGo(MX) are predicates which mean Warrior A and Warrior B go to the destination MX respectively. Re(MX) is a function which returns the resembling point adjacent to MX. The abstract state 3 represents warriors act individually. The dotted lines from state 3 have the same meaning as they are in LHMMs, these lines imply that the abstract state from 4 to 5 cannot stay for a moment, their functions are changing the instance of destinations when state 3 terminates. The abstract state 2 means that warriors will go towards assembling points together. However, it is notable that results of Re(MX) are uncertain and are sampled from a known distribution. The conditional transitions are depicted in Figure 4(a)and Figure 4(b).

$$3 \mapsto \begin{cases} \text{if } IsKilled(A,M_1) \wedge IsKilled(A,M_2) \text{ then } \begin{cases} p^1_{3,4} : 4 \\ p^1_{3,2} : 2 \end{cases} \\ \text{if } IsKilled(A_1,M_1) \text{ then } \begin{cases} p^1_{3,6} : 6 \\ p^2_{3,2} : 2 \end{cases} \\ \text{if } IsKilled(A_2,M_2) \text{ then } \begin{cases} p^1_{3,5} : 5 \\ p^2_{3,2} : 2 \end{cases} \\ \text{else then } \begin{cases} p^2_{3,4} : 4 \\ p^3_{3,2} : 2 \end{cases} \end{cases}$$

Figure 4(a): Conditional transitions from abstract state 3.

$$2 \mapsto \begin{cases} \text{if } \quad IsReached(A,B,\mathrm{Re}(MX)) \quad \text{then} \quad 1: \quad 7 \\ \qquad \text{else} \quad \text{then} \quad 1: \quad 3 \end{cases}$$

Figure 4(b). Conditional transitions from state 2.

$IsKilled(A,MX)$ is a predicate which means Monster MX is killed by Warrior A. And $IsReached(A,B,\mathrm{Re}(MX))$ means both A and B have reached the resembling point before going to kill their target MX. The observations are position series of warriors which can be used to judge whether the transition conditions are satisfied. The directed graphical representation of our game is depicted in figure 5.

3 SIMULATION

To evaluate the performance of LHSMMs using in MAIR, we set parameters manually and run the game. Since learning algorithm is not discussed in this paper, these parameters will be used directly. The conditional transition probabilities are given in Table 1.

Table 1: The conditional transition probabilities.

$p^1_{3,4}$	$p^1_{3,2}$	$p^1_{3,6}$	$p^2_{3,2}$
0.6	0.4	0.6	0.4
$p^1_{3,5}$	$p^2_{3,2}$	$p^2_{3,4}$	$p^3_{3,2}$
0.6	0.4	0.6	0.4

When the warrior is moving towards his destination, there may be more than one shortest path, and the warrior will choose one of them with an equal probability. Similarly, the assembling point is chosen by a uniform distribution. It is also assumed that the duration of abstract state 2 and state 3follow the same distribution.We collect the duration data and learn parameters of the $\log norm(\mu,\sigma,\theta)$ using distribution fitting tool. Then, the abstract self-transition probability p in LHMMs can be made equal to $1/\mu$, since the expectation of $geo(p)$ is $1/p$.

We select one typical set of traces of the warriors after several runs. The warriors both executed missions individually and cooperatively, and the working mode has even been terminated twice before the chosen monster was killed. The detailed information is shown in Table 2.

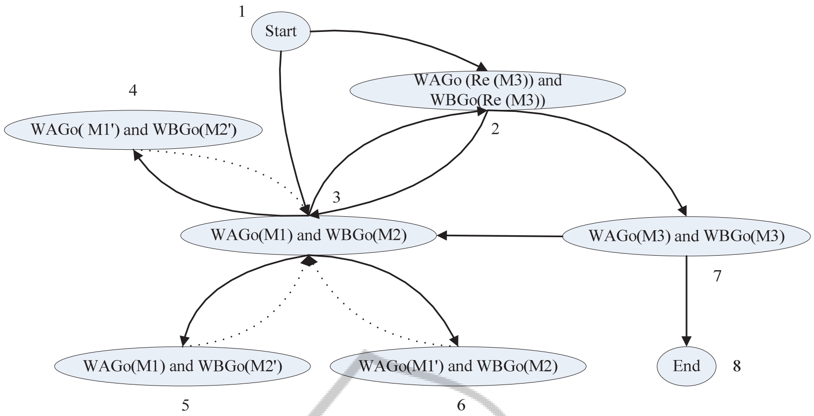

Figure 5: The directed graphical representation of the game.

Table 2: The set of traces.

No.	Durations	Abstract states	Instances	Interrupted
1	$t \in [1,1]$	Start	None	No
2	$t \in [2,8]$	State 2	M3: 4	Yes
3	$t \in [9,13]$	State 3	M1: 2, M2: 3	No
4	$t \in [14,17]$	State 3	M1: 1, M2: 4	Yes
5	$t \in [18,25]$	State 2	M3: 3	No
6	$t \in [26,26]$	State 7	M3: 3	No
7	$t \in [27,27]$	End	None	No

The first working mode chosen is working together and their target is the monster No. 4, but warriors changed their minds and decide to go to No. 2 and No. 3. After Warrior A completed his mission, their abstract state changed to state 6, which meant that both of them would have a new target. However, since state 6 could not stay and had to transfer to state 3, the records showed that warriors were staying in state 3. The fourth mission were interrupted again at t=17, and warriors decided to go to the monster No.3 together. In this time, the place which is in the north of the monster No.3 is regarded as the assembling point. According to the records, the Warrior B reached there earlier and he waited for Warrior B for 5 steps. Then, they went to kill the monster No.3 together and finished the game. We used LHSMMs and LHMMs to compute the probabilities of abstract state at each time respectively. The results are shown in Figure 6 and Figure 7.

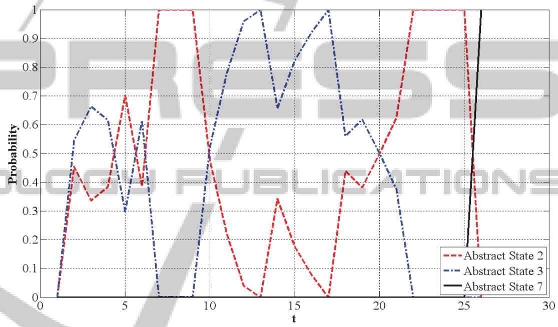

Figure 6: Probabilities of abstract states computed by LHSMMs.

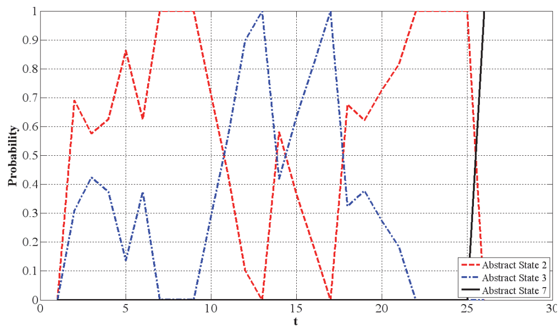

Figure 7: Probabilities of abstract states computed by LHMMs.

We only show the probabilities of abstract state 2, 3 and 7, because state 1 and 8 only exist at the first and the last step, state 4, 5 and 6 have transferred to state 3 during inference. The results prove that both of models can recognize the abstract at most times, but LHSMMs generally had a better performance: probabilities change more smoothly and LHMMs have a failed recognition at t=14. We also use LHSMMs to recognize monsters being chosen by two warriors. The results are shown in Figure8 and Figure 9.

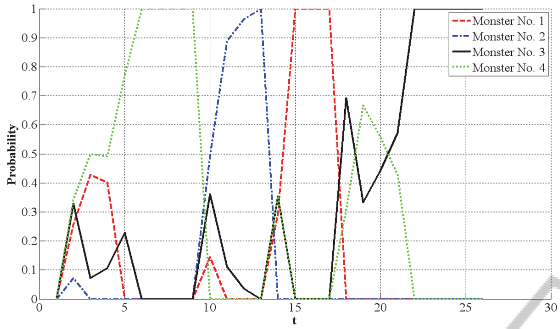

Figure 8.Probabilities of monsters chosen by Warriors A.

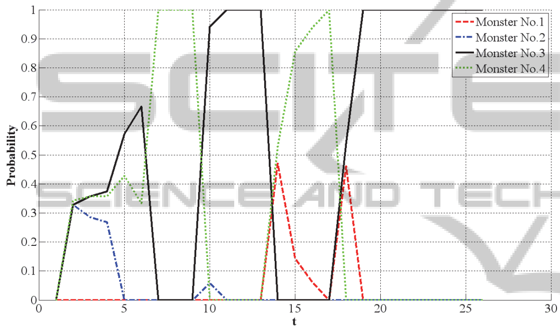

Figure 9: Probabilities of monsters chosen by warriors B.

Comparing recognition results shown in Figure 8 with the instance information in Table 2, we can find that LHSMMs have quite good performance to find the real monster chosen by Warrior A. Although there is a shake before t=20, the probability of Monster increases very fast and reach 1 at t=22, that is exactly the time when Warriors B is waiting him at the assemble point. Thus, even though Warrior A has not reach the assemble point we can still recognize his destination accurately. Figure 9 also shows the efficiency of recognizing intentions of Warrior B using LHSMMs.

4 CONCLUSIONS

In this paper, we analyze the history of intention recognition methods and advantages of LHSMMs using in MAIR. We further refine our former models by adding conditional transition probabilities and making the alphabet of instances changeable. According to these modifications, the inference process of MAIR based on the LFAD is depicted. We also design a simple game to evaluate the performance of LHSMMs. After using first order logic to describe the abstract states of the two warriors, we give the directed graphical representation of the game. The simulation results show that LHSMMs have a good performance on recognizing both working modes and missions of every warrior. In addition, we also find that the probability curves of abstract states computed by LHSMMs are smoother than LHMMs, and the result of working mode recognition is quite helpful to identify the goal of members in the team. In the future, we may do some research on modifying Viterbi and Baum-Welch algorithm in LHSMMs. Approximate inference algorithm which may need less computing time is also absorbing.

REFERENCES

Bui, H. H., Venkatesh, S. & West, G. (2002), Policy recognition in the abstract hidden Markov models, Journal of Artificial Intelligence Research, 17, 2002, 451–499.

Duong, T., Phung, D., Bui, H., Venkatesh, S. (2009), Efficient duration and hierarchical modeling for human activity recognition, *Artificial Intelligence,* 173(7-8), 830–856.

Dereszynski, E. Hostetler, J., Fern, A. & Dietterich, T., Hoang, T. & Udarbe, M. Learning Probabilistic Behavior Models in Real-time Strategy Games, In AIIDE'2011, *Proc. of the 7th In Artificial Intelligence and Interactive Digital Entertainment*, 20-25. AAAI Press.

Fine, S., Singer,Y., & Tishby, N. (1998), The hierarchical hidden Markov model: Analysis and applications, *Machine Learning*, 32(1): 41-62.

Kautz, H. A. & Allen, J.F., Generalized plan recognition, In AAAI'1986, *Proc. of the 5th National Conference on Artificial Intelligence*, 32-37.AAAI Press.

Kersting K., De Raedt L. & Raiko T. (2006).Logical hidden Markov models. *Journal of Artificial Intelligence Research*, 25, 425–456.

Natarajan, S., Bui, H. H. &Tadepalli, P., et al. 2008.Logical hierarchical hidden Markov models for modeling user activities. In ILP'2008, *LNCS* 2008, 192-209.Springer Heidelberg

Pfeffer, A., Das, S. & Lawless, D., et al. (2009) Factored reasoning for monitoring dynamic team and goal formation. *Information Fusion*, 10(1), 99-106.

Sadri, F. (2011). Logic-Based Approaches to Intention Recognition. In N. Chong, & F. Mastrogiovanni (Eds), *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*, Hershey, PA: Information Science Reference, 346-375.

Schmidt, C. F., Sridharan, N. S. & Goodson J. L. (1978).The plan recognition problem: an intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(1-2), 45-83.

Southey, F. Loh, W. & Wilkinson, D, Inferring Complex Agent Motions from Partial Trajectory Observations ,In IJCAI'2007, *Proc. of the 20th International Joint Conference on Artificial Intelligence*, 2631-2637. AAAI Press.

Yu, S. Z. (2010). Hidden semi-Markov models, *Artificial Intelligence*, 174, 215–243.

Zha, Y. B., Yue, S. G. &Yin, Q.J.，et al., 2013, Activity recognition using logical hidden semi-Markov models, In ICCWAMTIP'2013, *Proc. of 10th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 77-84, UESTC Press.

Zouaoui-Elloumi, S., Roy, V. &Marmorat J P, et al. Securing harbors by modeling and classifying ship behavior. In BRIMS'2011, *Proc. of 20th Annual Conference on Behavior Representation in Modeling and Simulation*, 114-121, Curran Associates.