

Research Article

A Logical Hierarchical Hidden Semi-Markov Model for Team Intention Recognition

Shi-guang Yue, Peng Jiao, Ya-bing Zha, and Quan-jun Yin

College of Information System and Management, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Shi-guang Yue; yshg052@hotmail.com

Received 22 April 2015; Accepted 15 September 2015

Academic Editor: Peng Shi

Copyright © 2015 Shi-guang Yue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intention recognition is significant in many applications. In this paper, we focus on team intention recognition, which identifies the intention of each team member and the team working mode. To model the team intention as well as the world state and observation, we propose a Logical Hierarchical Hidden Semi-Markov Model (LHHSMM), which has advantages of conducting statistical relational learning and can present a complex mission hierarchically. Additionally, the LHHSMM explicitly models the duration of team working mode, the intention termination, and relations between the world state and observation. A Logical Particle Filter (LPF) algorithm is also designed to infer team intentions modeled by the LHHSMM. In experiments, we simulate agents' movements in a combat field and employ agents' traces to evaluate performances of the LHHSMM and LPF. The results indicate that the team working mode and the target of each agent can be effectively recognized by our methods. When intentions are interrupted within a high probability, the LHHSMM outperforms a modified logical hierarchical hidden Markov model in terms of precision, recall, and F -measure. By comparing performances of LHHSMMs with different duration distributions, we prove that the explicit duration modeling of the working mode is effective in team intention recognition.

1. Introduction

Intention Recognition (IR) is to identify the specific goals that an agent/agents is/are attempting to achieve [1]. Since the goals are always hidden in mind, they can only be inferred by analyzing the agents' observed actions and/or the changes in the environment resulting from their actions.

The meaning of recognizing intention is significant in both real and virtual worlds. For example, in real-time strategy games, AI players can choose more efficient policies if their enemies' actions are known [2]; in the field of public security, we want to distinguish the persons who may behave abnormally and pay more attention to them through monitor devices [3]. Since the IR problem is common, it has attracted many attentions in recent decades.

As a famous branch of probabilistic graphical models (PGMs), the hidden Markov model (HMM) is very popular for analyzing sequential data in many applications [4]. In the IR domain, actions are discrete and observations are got sequentially. Thus, the HMM can also be used to solve IR

problems, whose main idea is as follows: the hidden states and observation sequence correspond to actions and the change of world states, respectively, and the intention is represented by the transitions between actions. Then, the inference of intention is solved by a most likelihood estimation: to find the intention which can reproduce the given observation sequence with the largest probability.

Even though the HMM looks suitable to model intentioned behaviors and observations, there are still some problems when the HMM is applied in the IR domain:

- (a) The HMM has a strong Markov assumption. In the HMM, the duration of a state is implicitly a geometric distribution, whose parameter is the state self-transition probability. However, in many applications, the duration of hidden states does not follow the geometric distribution.
- (b) The HMM does not have a hierarchical structure. However, when a mission is complex, we always need to decompose it into subtasks repeatedly until the

mission only consists of primitive actions, and the hierarchical structure is necessary to present the task decomposition and allocation.

- (c) The HMM is actually propositional. There is no concept of the class or object in the HMM, which makes the HMM a poor representative. Additionally, the propositional model is not suitable to tackle relations, which are important in the IR.
- (d) The most likelihood inference assumes that the intention is static. However, the team intention may be interrupted and changed because of new situations or other reasons.

To solve these problems, researchers modified the HMM in different ways. For example, the Hidden Semi-Markov Model (HSMM) uses an arbitrary distribution such as Poisson and Gamma to model the state duration explicitly and sets the self-transition probability zero [5]. In this way, the HSMM models the state duration more precisely, and it always outperforms the HMM when the type of state duration distribution is known. The Hierarchical Hidden Markov Model (HHMM) and Abstract Hidden Markov Model (AHMM) are two extensions of the HMM [6, 7]. They present the task hierarchies based on the theories of Hierarchical Finite State Machine and the Abstract Markov Decision Process, respectively. The Coxian Hidden Semi-Markov Model (CxHSMM) further combines ideas of the HSMM and the HHMM [8]. Two important contributions of the CxHSMM are as follows: (a) proposing a novel Dynamic Bayesian Network (DBN) structure to model human daily activities; (b) introducing the discrete Coxian distribution in the behavior modeling domain.

Another type of extensions is related to relation data. These models fall into the theory of the Statistical Relational Learning (SRL), which integrates the relational or logical representations and probabilistic reasoning mechanisms with machine learning [9]. For example, Kersting et al. proposed a logical hidden Markov model which combined the first-order logic with the HMM [10]. Comparing with the HMM, the LHMM can infer complex relations and have fewer parameters. However, it does not relax Markov assumption, which leads to a performance decline when long-term dependences between hidden states exist. To solve this problem, Zha et al. proposed a Logical Hidden Semi-Markov Model (LHSMM), which modified the LHMM in the same way the HSMM modified the HMM [11]. Natarajan et al. proposed a Logical Hierarchical Hidden Markov Model (LHHMM) and applied it upon recognizing intentions of an agent in a virtual game world [12]. The LHHMM has a hierarchical architecture and divides the state space into two types: (a) the unobserved user state and (b) the completely observed world state. Besides, the abstract state transitions are conditioned by the world state, which reflects how the world state affects the state of user.

Extensions of the HMM above focus on recognizing intentions of one agent. However, teamwork is quite common in many scenarios. In this paper, we focus on the problem of team intention recognition. Obviously, recognizing the intention of a team is more complex than that of a single agent. Because (a) we need to recognize the goal of each agent

as well as the team working mode, (b) task decomposition and allocation, which depend on the team working mode, need to be presented hierarchically, (c) the team intention may be interrupted with some unknown reasons, and (d) observations are noisy and partially missing.

The research on the LHSMM has shown that (a) the logic predicates and instantiation process can represent the working mode and the composition of the team well and (b) modeling duration of abstract state can get a higher precision and smoother recognition curve [13]. However, the LHSMM does not provide a hierarchical architecture. The LHHMM is also promising, but Natarajan et al. only consider single agent scenarios, and the LHHMM may suffer some problems when it is used in team intention recognition: (a) tasks modeled by the LHHMM are executed from bottom to up, which means the top goal cannot be interrupted when the subtasks are not completed; (b) the duration of the working mode is not modeled; (c) observation cannot be noisy or partially missing; (d) the chain structure is not proper to represent the primitive actions [12]. Actually, unlike higher level task, primitive actions usually depend on the world state and the goal, as that defined in a Markov decision process (MDP).

To solve problems of applying existing models in team intention recognition, we propose a framework named Logical Hierarchical Hidden Semi-Markov Model (LHHSMM). The LHHSMM borrows the ideas of the LHHMM and combines it with the LHSMM as well as the MDP. Our LHHSMM has advantages as below:

- (a) Comparing to PGMs such as the HMM and its extensions, the LHHSMM has the advantages of SRL methods. By introducing the first order logic, LHHSMM can infer complex relations and use logical inference to replace some probabilities computing. Additionally, the predicates and instantiation process are very suitable to represent team working mode and changes of team member.
- (b) A novel structure named logical hierarchical Markov chain (LHMC) is proposed to present logical transitions and decomposition of the intention and its subtasks (we call the intention and its subtasks policies). With this structure and transition conditions, our model inherits the compactness of presenting policies hierarchically in the LHHMM, plus a mechanism which makes the executing subtasks terminated forcibly in case that intentions are changed.
- (c) Considering that the team intention may be interrupted by some unknown reasons, we use a lognormal distribution to model the duration that the team working mode is not interrupted, or the time remaining before an interruption event happens. With this explicit duration modeling, the current team intention depends on not only the team intention and world state at previous step, which is another difference between the LHHMM and the LHHSMM and the reason that our model is called semi-Markov.
- (d) Primitive actions are selected based on the current policies and previous world states. This MDP-like

process makes the agents choose any primitive actions without limits of the executed action in previous time.

- (e) Observation functions are used to present the probabilistic relations between the world state and observation. It makes our model able to tackle noisy and partially missing observations.

To infer the team intentions modeled by the LHHSMM approximately, we provide a Logical Particle Filtering (LPF) based on logical definitions and dependency of the LHHSMM. In the LPF, we use the simplest importance distribution and a forward sampling method to sample the particles, and logical transitions and instantiation functions in the LHHSMM are introduced in this process.

We design a combat scenario to validate the LHHSMM and LPF: two agents move around and attack targets on a grid map. Our methods are used to infer the team working mode and targets of agents online according to the observed agents' traces. Based on this scenario, we design a decision model for the agents and generate a dataset consisting of 100 traces. We use three traces (one changes intentions, others do not) in the dataset to evaluate the LHHSMM and the LPF. Then, three metrics including precision, recall, and F -measure of recognizing the team working mode and targets of the agent are computed by the LHHSMM and a modified LHHMM, respectively. Last, we compare the performances of LHHSMMs with three different duration distributions to evaluate the effects of explicit duration modeling.

The rest of the paper is organized as follows: Section 2 introduces some related works including extensions of the HMM applied in IR and other researches on team intention recognition using PGMs. Section 3 gives the formal definition of the LHHSMM, the dependency among variables, and discussions on how to use a LHMC to present policies. Section 4 introduces the standard PF briefly and gives the process of inferring intentions approximately by the LPF. Section 5 presents the background, settings, and results of our experiments. Subsequently, we have conclusions and discuss future works in Section 6.

2. Related Works

By making an intersection of psychology and artificial intelligence [14], a great number of IR methods have been proposed; these methods can be divided into four categories broadly: consistency matching, probabilistic methods, hybrid methods, and statistical relational learning methods [15].

The logical reasoning based on event hierarchy [16] and fast and complete symbolic methods [17] are two representative consistency matching approaches. They solved the IR problem by determining which intention was consistent with observed actions, that is, whether the observed actions matched at least a plan achieving the intention. Their main drawback is that they may fail when there are two or more hypotheses to give contradict explaining upon the same observations. Relatively, probabilistic methods can provide us a probability for every possible intention, and they can make full use of prior knowledge by Bayesian inference [18]. Because of these advantages, some kinds of probabilistic

method are applied in IR domain, such as probabilistic grammar [19], PGMs including the HMM [20], and the Conditional Random Field (CRF) [21]. To improve the computational efficiency, some scholars proposed hybrid models, such as hybrid symbolic-probabilistic plan recognizer [22] and probabilistic hostile agent task tracker [23]. Their primary idea is to compute the likelihood of possible intentions using Bayesian reasoning after scaling down the hypothesis space by logical algorithms. However, these hybrid methods inherit the drawbacks of the related consistency matching and probabilistic methods. One key problem of PGMs and their corresponding hybrid methods is that they are actually propositional, which means they handle only sequences of unstructured symbols. Thus, the SRL was proposed and developed rapidly in the recent years. The SRL presents and infers the relations, which are quite important to recognize intentions in real world. Therefore, many researchers made great efforts to apply the SRL methods in the IR problem, such as the LHMM [10] and the Markov Logic Network (MLN) [24, 25].

In this section, we first review some research about applying PGMs in team intention recognition. Then, some extensions of the HMM related with our model will be analyzed.

2.1. Team Intention Recognition Using PGMs. A probabilistic graphical model is used to encode a complex distribution over a high-dimensional space, by using a graph-based representation. The graph usually consists of the nodes and edges, which correspond to the variables in the domain and direct probabilistic interactions between variables, respectively [26]. Two types of PGMs are widely applied in team intention recognition: the directed ones such as the DBN and HMM and the undirected ones such as the relational Markov network and the CRF.

Masato et al. [27] introduced the CRF to automatically recognize the composition of teams and team activities in relation to a plan. Mao et al. [28] viewed group plan recognition as inferring the decision making strategy of observed agents. They assumed that agents were rational and made the probabilistic reasoning based on the maximum expected utility principle. Their plan representation and inference were actually done on a Bayesian network. Pfeffer et al. [29] studied the problem of monitoring goals, team structure, and state of agents, in dynamic systems where teams and goals changed over time. By using DBN, they modeled coordination and communication of attackers explicitly in an asymmetric urban warfare environment. Saria and Mahadevan [30] presented a theoretical framework for online probabilistic IR in cooperative multiagent systems. Their model extended the abstract hidden Markov Model (AHMM) and consisted of a hierarchical dynamic Bayesian network that allowed reasoning about the interaction among multiple cooperating agents. Gaitanis [31] modified Saria's model by releasing the assumption that there was only one team grouping all the agents at only one level of coordination. Although these models solve multiagent IR problems successfully to some extent, they suffer the drawback of traditional PGMs: they are propositional and cannot make use of relations.

Some SRL methods such as the MLN and LHMM can also be regarded as special cases of PGMs. Sadilek and Kautz [32] modeled “capture the flag” domain using MLN and learned a theory that jointly denoised the data and inferred occurrences of high-level activities. Their research showed that the MLN was quite potential to solve multiagent IR problems. To compare the performance of the CRF, HMM, and MLN, Auslander et al. [33] used these methods to help commanders to detect maritime threats. Their evaluation corpus was from the 2010 Trident Warrior exercise, and they proved that the MLN was better to represent domain knowledge and learn weight settings from a few training instances. As far as we know, there are few attempts to apply extensions of the LHMM in team intention recognition, except for our previous research in [13].

2.2. Extensions of the HMM Applied in the IR Domain. Even though the HMM has some advantages in behavior modeling, the strong Markov assumption limits its application in many areas. Thus, some research has been done to extend HMM in the IR domain.

A Hidden semi-Markov model (HSMM) is the same as the HMM, except for modeling the duration of hidden states explicitly. Because of the advantage of modeling state duration precisely, people use HSMMs to solve IR problems in the digital games. For example, Hladky and Bulitko applied the HSMM to predict the position of a player in the first person shooting game [34]. The database consisted of 190 game logs collected in Counter Strike competitions. With these data, they trained the HSMM and evaluated predictor performance by both prediction accuracy error and human similarity error. Southey et al. also used the HSMM to recognize the destination and start point of an agent on the grid map of War-Craft 3 [35]. Their main contribution is that the observed trajectory can be partially missing.

van Kasteren et al. compared the performances of activity recognition using the HMM, HSMM, CRF, and Semi-Markov CRF, respectively; their activity data was recorded by real sensors in smart home [36]. The experiments showed that the modeling of duration explicitly always improved the recognition performances.

AHMM is a stochastic model for representing the execution of a hierarchy of contingent plans [7]. The core conceptions of the AHMM are the abstract policy and the termination variable. The abstract policy is defined as the selection of lower-level abstract policy or primitive action, given the current states and higher-level policy (the top-level policy only depends on the state). The termination variable indicates whether the corresponding abstract policy will terminate in each time slice. When a policy does not terminate, its higher level polices cannot terminate either. When the AHMM is used to recognize policies, there is also an observation layer, which depends on the states. Bui et al. applied the AHMM to track an object and predict the object future trajectory in a wide-area environment [37].

Another famous extension of the HMM is the Coxian hidden semi-Markov model (CxHSMM) [8]. This model modifies the HMM in two aspects: on one hand, it is a special DBN representation of a two-layer HMM, and it also has

termination variables; on the other hand, it used Coxian distribution to model the duration of primitive actions explicitly. This model was applied in recognizing human activities of daily living (ADLs), and Duong et al. showed that Coxian duration model had advantages over existing duration parameterization using multinomial or exponential family distributions.

3. Logical Hierarchical Hidden Semi-Markov Model

The LHSMM is a fusion of logical hierarchical hidden Markov model, logical hidden semi-Markov model, and Markov decision process. It is used to model the team to be recognized as well as the world state and observation. In this section, we will give a formal definition of the LHSMM and describe the dependency by a DBN representation. Then, we will explain how to use a logical hierarchical Markov chain to present the logical transition and decomposition of policies.

3.1. Definition. LHSMM in one time slice is a tuple $\mathbf{M} = (\mathbf{N}_t, \Delta_t, \gamma_t)$, where \mathbf{N}_t is a belief network with a time label, Δ_t is the set of transitions from \mathbf{N}_t to \mathbf{N}_{t+1} , and γ_t defines the initial distribution in \mathbf{N}_t .

\mathbf{N}_t represents the set of the policies to be recognized from level-1 to level- K (K is the top level), the primitive action, the world state, and observation at time t . The level- K policy is called intention and the primitive action is in level-0. Each \mathbf{N}_t is associated with a logical alphabet $\Sigma = (\Sigma^0, \Sigma^1, \dots, \Sigma^K)$ in first-order logic, where Σ^k belonging to level- k ($k = 0, 1, \dots, K - 1, K$), is a set of relation symbols r with arity $m \geq 0$ and a set of function symbols f with arity $n \geq 0$. When $m = 0$, r becomes a proposition, and f is a constant when $n = 0$. An atom $r(t_1, t_1, \dots, t_m)$ is a relation symbol r followed by a bracketed n -tuple of terms t_i . A term is a variable V or a function symbol $r(t_1, t_1, \dots, t_k)$ immediately followed by a bracketed k -tuple of term t_i . A term is called ground when it contains no variables. The *Herbrand* base of Σ^k , denoted as hb_{Σ^k} , is the set of all ground atoms constructed with the predicate and function symbols in Σ^k . The set $G_{\Sigma^k}(A)$ of an atom A consists of all ground atoms that belongs to hb_{Σ^k} [10].

Definition 1 (policy and primitive action). A level- k abstract policy A^k is an atom with items in Σ^k ($k > 0$), and the instantiated A^k is called ground policy π^k , which belongs to $G_{\Sigma^k}(A^k)$. In our paper, the top level ground policy π^K is also regarded as an intention. An abstract primitive action A^0 is an atom in Σ^0 , and π^0 is the corresponding ground primitive action. Each Σ^k is determined by a function $\Sigma^k(A^{k+1})$, except for Σ^K which is predefined. Thus, the abstract policy A^k ($k > 0$) determines the logical alphabet in its lower level.

Example 2. $\Sigma^1 = \langle \text{Relations: Attack}(T_1), \text{Avoid}(T_1); \text{Variables: } T_1, \text{Constants: } t_1, t_2 \rangle$ ($K = 1$), which means two kinds of intentions: attacking the target or avoiding the target, but there will be four intentions after the instantiation

process, because the target can either be t_1 or t_2 . Then, Attack (T_1) is an abstract policy A^K and Attack (t_1) is policy π^K . Attack (t_1) determines the level-0 logical alphabet $\Sigma^0 = \langle \text{Relations: Move (Direction)}; \text{Constants: north, south, east, west, null} \rangle$, which cannot be decomposed further.

Definition 3 (the world state and observation). The world state S and observation O are both sets of variables. S depicts the status of the agents and the environment; the variable symbols in S are not predefined but are generated by instantiation functions and logical transitions, which will be introduced later. The observation O is a function of S ; it provides us some inaccurate and partial information about S .

Example 4. $S = \langle \text{Variables: Position A}; \text{Constants: } l_1, l_2, l_3, l_4 \rangle$, which means the world state consists of a variable, the position of agent A, and there are four possible positions totally. $O = \langle \text{Functions: Obs (Position A)}; \text{Constants: } l_1, l_2, l_3, l_4, \text{null} \rangle$ is the observation of S , the observed value of Position A may reflect the real position of agent A or not, and the relations of S and O are presented by conditional probability $\text{Pr}(O | S)$.

Definition 5 (instantiation function). An instantiation function μ^k in level- k is defined by policy π^{k+1} in the higher level, except that μ^k is predefined. μ^k is a mapping $S \times \Sigma^k \times hb_{\Sigma^k} \rightarrow [0, 1]$, which can also be presented as a conditional probability $\mu^k(\pi^k | A^k, S)$, where A^k is an atom relation symbol in Σ^k , S is the world state in previous time, and $\pi^k \in G_{\Sigma^k}(A^k)$ is an instantiated object of A^k . It should be noted that when a variable in A^k has been instantiated in the higher level policy, the variable should be substituted by the former instance directly and cannot be selected by μ^k again.

Example 6. For the Σ^1 in Example 2, we can set $\mu^1(\text{Attack}(t_1)) = 0.3$, $\mu^1(\text{Attack}(t_2)) = 0.7$, $\mu^1(\text{Avoid}(t_1)) = 0.8$, and $\mu^1(\text{Avoid}(t_2)) = 0.2$. If T_1 is instantiated as t_2 in level-1. Then, we can set $\mu^0(\text{Move}(\text{right})) = 0.8$, $\mu^0(\text{Move}(\text{left})) = 0.2$ (there are only two directions), but $\mu^0(\text{Fire}(t_2))$ can only be 1.

Definition 7 (policy termination [7]). Policy termination variable $e^k \in \{0, 1\}$ indicates whether the level- k policy π^k will be terminated ($e^k = 1$) or not ($e^k = 0$) at the current time. When $e^k = 0$, π^k will continue at the next time; in the other case, it will be changed according to Δ_t or γ_t .

Definition 8 (intention duration [8]). Intention duration variable d is a counter to model the time remained before an interruption event happens. Since the reason of the interruption is always unknown, we initialize the value of d using a lognormal distribution when an intention starts. The value of d will reduce 1 in each step, and the intention will be interrupted when $d < 1$. Then, d will be initialized again.

Δ_t is the set of horizontal transitions, which determines how N_t transits into N_{t+1} . There are two types of transitions in Δ_t : ones are logical transitions which can be further classified into conditional logical probabilistic transitions,

specific logical transitions, and unified logical transitions and the other ones are standard probabilistic transitions (we just call them probabilistic transitions) as those in PGMs. The logical transitions only exist between abstract policies in the same level, and transitions in each level are generated by abstract policies in the higher level, except that level- K logical transitions are predefined. The probabilistic transitions will be analyzed when we introduce the DBN representation of our model.

Definition 9 (conditional probabilistic logical transition [12]). A conditional probabilistic logical transition has a form $p : B \rightarrow H$, which means that the abstract policy will transit to H from B with probability p , where B is the current abstract policy and H is the next abstract policy, they are both atoms in the alphabet of the current level, p is a conditional probability which can be presented as $\Delta_t(H | B, C(B))$, where $C(B)$ represents conditions associated with B , which is a set of logical sentences whose variables are included in the world state. Thus, we get the true value of $C(B)$ if we know the current world state. $C(B)$ can also be an empty set, and the value of p only depends on the higher level and policy and B in this case. We need to emphasize that if the variables of H have been instantiated in B , they will be substituted by the constants directly.

Definition 10 (specific logical transition). A specific logical transition has a form $B \rightarrow B'$, which means that we replace the current policy B with B' , where B' is the more specific case of B . As a kind of default reasoning, this kind of transition is often used to model exceptions, and it will be followed when the instance in B is consistent with that in B' .

Definition 11 (unified logical transition). A unified logical transition has a form $B \rightarrow B''$, which means that we replace the current policy B with B'' , where relation symbols of B and B'' are the same, but the orders of their variable symbols are different. When we use unified logical transition, the grounds B'' and B will be unified. This kind of transition is forced to follow if the current policy is B .

These three kinds of logical transitions can be represented by solid edge, dashed edge, and dotted edge, respectively, in a FSM, as in the LHMM [10]. We need to note that since dotted and dashed transitions do not take real time, they can only be presented in a FSM but cannot be reflected in the DBN representation. Figure 1 shows three examples to explain them.

A_1, A_2 , and A_3 are relation symbols which represent abstract policies, X_1, X_2 , and X_3 are variables, and x_2 is a constant. Suppose that we have computed the values of all conditional probabilistic logical transition according to the world state, and they are denoted on the edges. In Figure 1(a), the abstract policy can transit to $A_2(X_1, X_3)$ or $A_3(X_2, X_3)$ from $A_1(X_1, X_2)$ with probabilities 0.4 and 0.6, respectively. If the abstract policy reaches A_1 , the instantiation results of X_1 in A_1 must be the same as X_1 in A_2 . In Figure 1(b), if the instantiated result of X_2 in A_1 is not x_2 or we have no idea about it, the abstract policy transits to $A_2(X_1, X_3)$ or $A_3(X_2, X_3)$ with probabilities 0.4 and 0.6, respectively.

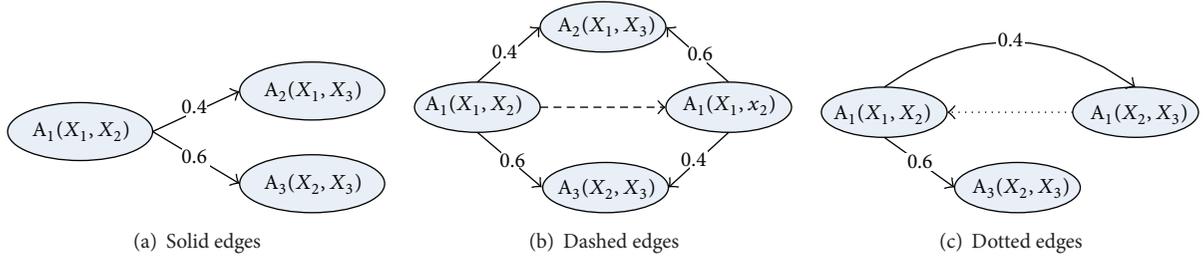


FIGURE 1: Three examples of logical transitions.

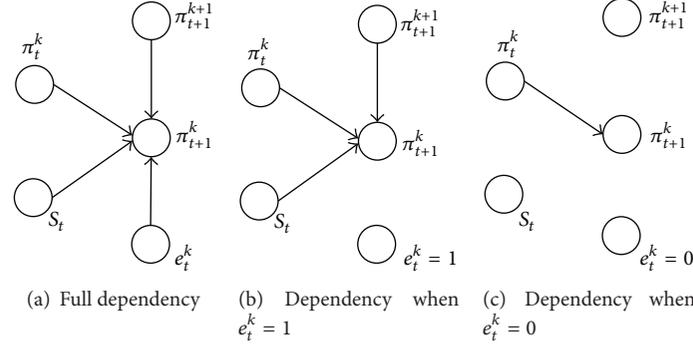


FIGURE 2: Subnetwork for a policy.

However, if we know the instance of X_2 is x_2 , the probabilities above will change to 0.6 and 0.4. In Figure 1(c), if the current abstract policy is $A_1(X_2, X_3)$, we will follow the dotted edge automatically and replace the abstract policy with $A_1(X_1, X_2)$, and the instances of X_1 and X_2 will be the same as instances of X_2 and X_3 in $A_1(X_2, X_3)$. In this way, $A_1(X_1, X_2)$ changes its instances consuming one time.

γ_t defines the initial distributions of abstract policies in \mathbf{N}_t , $\gamma_t(A^k \mid A^{k+1})$ returns the initial distribution of abstract policy in level- k ($k = 1, 2, \dots, K - 1$), which is generated by the current abstract policy A^{k+1} , and $\gamma_t(A^K)$ is predefined. $\gamma_t(A^k \mid A^{k+1})$ is only used to sample A^k when the level- k policy in the previous time has been terminated. For the Σ^1 in Example 2, γ_t can be represented as $\gamma_t: \{\text{Attack } (T_1): 0.7; \text{Avoid } (T_1): 0.3\}$.

3.2. Dependency in the LHSM. In this section, we will use a DBN presentation to describe the dependency among variables in the LHSM. However, the standard DBN is not available to present the logical transitions and instantiation process in our model, since it is actually propositional. Thus, we can only show the full DBN after substituting all variables. To explain logical dependency under standard probabilistic transitions, we will analyze the factors which each policy, primitive action, termination, duration, and state depend on and discuss the details about the logical transitions and instantiation process after that. Figure 2 shows the subnetwork for a level- k policy.

When $1 \leq k < K$, π_{t+1}^k depends on two ground relation symbols π_{t+1}^{k+1} as well as π_t^k and two variables e_t^k

and S_t , as is shown in the full dependency. Figure 2(b) means that when π_t^k has been terminated at time t , the transition and instantiation of A_{t+1}^k will depend on π_t^k , π_{t+1}^{k+1} and S_t . The transition process should be considered in two cases: when π_{t+1}^{k+1} is not terminated at time t , A_{t+1}^{k+1} defines the set of possible A_{t+1}^k and determines the probability of transiting from A_t^k to A_{t+1}^k together with $C(A_t^k)$ whose true value depends on S_t ; when π_{t+1}^{k+1} is terminated at time t , A_{t+1}^{k+1} determines the initializing probability of A_{t+1}^k . For the instantiation process, the instances of the same variables in π_{t+1}^{k+1} and π_{t+1}^k must be consistent, and the instances of π_{t+1}^k should also satisfy the logical transitions between π_t^k and π_{t+1}^k . Figure 2(c) means that π_{t+1}^k inherits π_t^k totally when π_t^k is not terminated at time t . When $k = K$, π_{t+1}^K only depends on π_t^K , S_t , and e_t^K , and the selection and instantiation of the intention are similar as other level policies, except that there is no influence from higher level policies. Figure 3 shows subnetwork for duration of the intention.

Duration d_{t+1} depends on a relation symbol A_{t+1}^K and two variables e_t^K and d_t , as is shown in Figures 3(a), 3(b), and 3(c), explain two cases, respectively: when A_t^K is terminated at time t , d_{t+1} will be initialized and have a new value according to A_{t+1}^K ; when A_t^K is not terminated at time t , we make $d_{t+1} = d_t - 1$. Figure 4 shows subnetwork for level- k ($k < K$) policy termination.

Figure 4(a) shows that e_t^k depends on a ground relation symbol π_t^k and two variables e_t^k and S_t . In Figure 4(b), when $e_t^{k+1} = 1$, e_t^k will be forced to 1, which means a policy will be terminated forcibly when its higher level policy ends. In

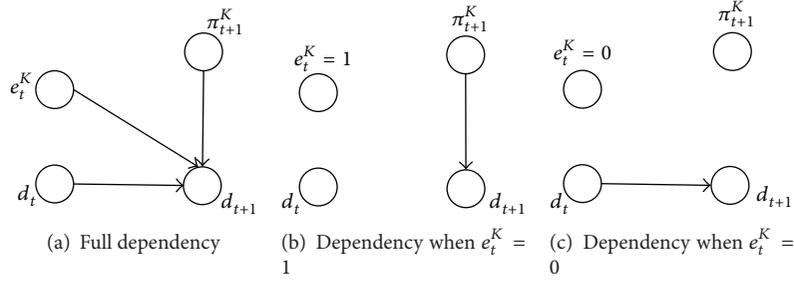


FIGURE 3: Subnetwork for duration.

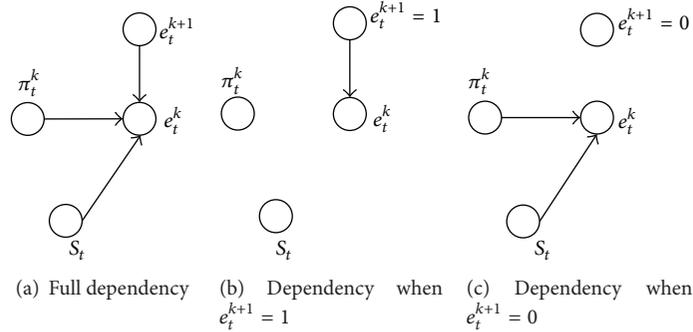

 FIGURE 4: Subnetwork for k -level ($k < K$) policy termination.

Figure 4(c), $e_t^{k+1} = 0$, and the value of e_t^k depends on a conditional distribution $p(e_t^k | S_t, \pi_t^k)$. Particularly, we set $p(e_t^k = 0 | S_t, \pi_t^k) = 1$, for all the policies which are not any head of logical transition in Δ_t . In other words, this kind of policies can only be terminated when $e_t^{k+1} = 1$. Figure 5 shows subnetwork for intention termination.

Figure 5(a) shows that e_t^k depends on a ground relation symbol π_t^k and two variables d_t and S_t . In Figure 5(b), when $d_t < 1$, $e_t^k = 1$ which means the intention has been interrupted because of unknown reasons. Figure 5(c) shows the case that when $d_t \geq 1$, the value of e_t^k is determined by $p(e_t^k | S_t, \pi_t^k)$, just like policies in other levels.

Relationships between world state, observation, and primitive action are similar as Markov decision process. First, given the level-1 policy π_{t+1}^1 and the world state S_t , an abstract primitive action A_{t+1}^0 in Σ^0 will be selected by a probability $p(A_{t+1}^0 | \pi_{t+1}^1, S_t)$. Then, A_{t+1}^0 will be instantiated into π_{t+1}^0 by μ_{t+1}^0 , and π_{t+1}^0 determine how the current world state S_t transits to S_{t+1} , and the observation O_t is a function of S_t . The primitive action does not have a termination variable; it needs to be selected according to the process above in each time slice. The variables and dependency are presented in a full DBN representation, which is shown in Figure 6.

The DBN representation depicts the dependency in \mathbf{N}_t and transitions between \mathbf{N}_t and \mathbf{N}_{t+1} . We need to note that the vertical dashed lines in Figure 6 are not logical transitions but are brief representations for the nodes and edges from level-3 to level $K-1$. As is shown in the DBN representation, our model actually consists of three parts: sensors which are modeled by the world state and observation levels; a Markov

decision process depicted in the structure below level-1, if we regard the level-1 policy as a special case of state; a hierarchical chain structure above level-0, which depicts the team policies. However, the DBN is unavailable to present the logical transitions and instantiation process in the third part. Thus, we will use a hierarchical logical Markov chain to describe the decomposition and transition process of team policies in our model.

3.3. Team Policies Presented by Hierarchical Logical Markov Chain. Unlike the DBN structure, a logical hierarchical Markov chain (LHMC) ignores the variables of duration, conditions of logical transitions, primitive actions, world states, and observations but only depicts the decomposition of policies and logical transitions among them. A LHMC is a tree structure whose each node is a logical Markov chain (LMC); the LMC can be regarded as a LHMM without any observation, just like the state transition process in a standard HMM is a Markov chain. In this paper, an abstract state in the LMC represents an abstract policy and its child is a LMC in its lower lever except for the policies in leaf nodes. Figure 7 shows an example of team policies presented by a hierarchical logical Markov chain.

The team policies in Figure 7 consist of two layers: level-1 and level- K ($K = 2$). The LMC in level-2 presents the possible intentions transitions between them. In team intention recognition problems, the relation symbols in highest level usually represent different team working modes. For example, Intention B indicates a team working mode, and T_3 and T_2 are specific constants. The Start node and End node in the level- K LMC are used to control the

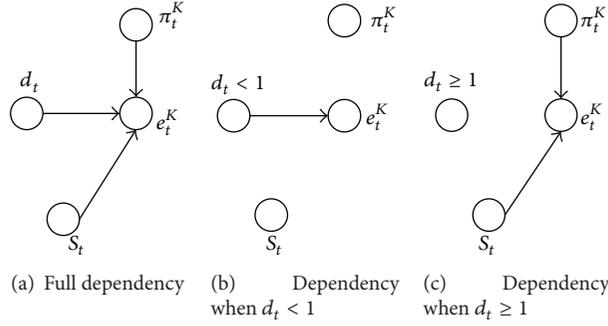


FIGURE 5: Subnetwork for intention termination.

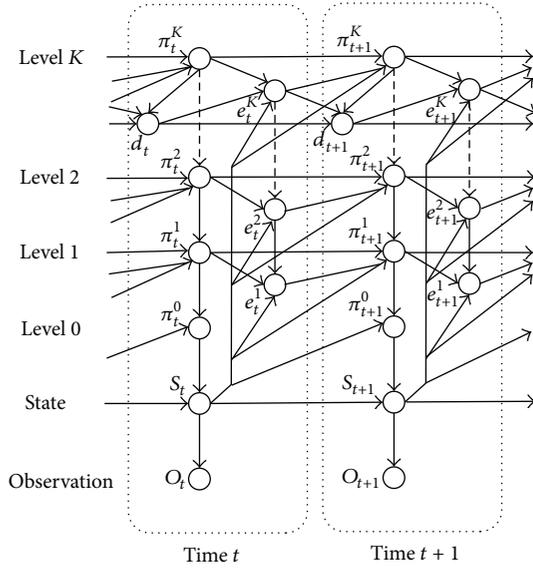


FIGURE 6: The full DBN representation in two time slices.

simulation process. If the intention reaches the End node, the simulation will be ended. The outlines from the Start node are associated with an initial distribution of abstract intention. When agents choose a level- k policy ($k > 1$), they need to execute policies in its associated LMC, as shown in the eclipse (Figure 6 ignores mappings of other policies for simplicity). For example, to finish Intention $B(T_3, T_2)$, the execution in level-1 will begin according to the initial distribution defined by Intention $B(T_3, T_2)$, and the first executed level-1 abstract policy may be Subtask $A(X_1)$ or Subtask $B(X_2)$; when the level-1 policy is terminated but Intention $B(T_3, T_2)$ is not, level-1 policy will be transited in the LMC. But when the world state satisfies the terminating conditions of Intention $B(T_3, T_2)$ or it is interrupted, we will return to abstract policy in level-2 and transit to another abstract intention in level- K LMC immediately.

4. Approximate Inference

In this section, we will discuss how to infer team intentions presented by the LHHSM. Online intention recognition is essentially treated as a filtering problem. The policies,

primitive actions, duration, and termination can be regarded as states of a dynamic system. They cannot be observed directly but can produce some observations sequentially. Our goal is to infer the real state at each time according to these observation series. Since there are noise and data missing when we observe the state, we will use a particle filter (PF) to solve the approximate inference problem. However, the standard PF does not define any logical transition and instantiation process. Thus, we will propose a new logical particle filter by introducing logical definitions and dependency in LHHSM.

4.1. Standard PF with Simplest Sampling. Suppose that we have a dynamic system whose real state at time t is \mathbf{x}_t , and \mathbf{y}_t is an observation of \mathbf{x}_t . $p(\mathbf{x}_0)$ is the prior initial probability. $\{\tilde{\mathbf{x}}_t^i, w_t^i\}_{i=1}^{N_s}$ is a particle set which describe the posterior distribution $p(\tilde{\mathbf{x}}_t | \tilde{\mathbf{y}}_t)$, where $\tilde{\mathbf{x}}_t = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$ and $\tilde{\mathbf{y}}_t = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t\}$. N_s is the number of particles, w_t^i is the normalized weight of $\tilde{\mathbf{x}}_t^i$, and $\sum_i w_t^i = 1$. Then, the posterior distribution of the state can be computed by

$$p(\mathbf{x}_t | \tilde{\mathbf{y}}_t) = \sum_{i=1}^{N_s} w_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i), \quad (1)$$

where δ is the Dirac function and w_t^i is got by

$$w_t^i \propto \frac{p(\tilde{\mathbf{x}}_t^i | \tilde{\mathbf{y}}_t^i)}{q(\tilde{\mathbf{x}}_t^i | \tilde{\mathbf{y}}_t^i)}, \quad (2)$$

in the PF, the q is known as *the important distribution* which is used to sample $\tilde{\mathbf{x}}_t^i$, and it can be factorized by

$$q(\tilde{\mathbf{x}}_t | \tilde{\mathbf{y}}_t) = q(\mathbf{x}_t | \tilde{\mathbf{x}}_{t-1}, \tilde{\mathbf{y}}_t) q(\tilde{\mathbf{x}}_{t-1} | \tilde{\mathbf{y}}_{t-1}). \quad (3)$$

And the posterior distribution can be represented by

$$p(\tilde{\mathbf{x}}_t | \tilde{\mathbf{y}}_t) \propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\tilde{\mathbf{x}}_{t-1} | \mathbf{y}_{t-1}). \quad (4)$$

After substituting formula (3) and formula (4) into formula (2), we can update the importance of weights by

$$\begin{aligned} w_t^i &\propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^i) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i) p(\tilde{\mathbf{x}}_{t-1}^i | \tilde{\mathbf{y}}_{t-1}^i)}{q(\mathbf{x}_t^i | \tilde{\mathbf{x}}_{t-1}^i, \tilde{\mathbf{y}}_t^i) q(\tilde{\mathbf{x}}_{t-1}^i | \tilde{\mathbf{y}}_{t-1}^i)} \\ &= w_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_t^i) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i | \tilde{\mathbf{x}}_{t-1}^i, \tilde{\mathbf{y}}_t^i)}, \end{aligned} \quad (5)$$

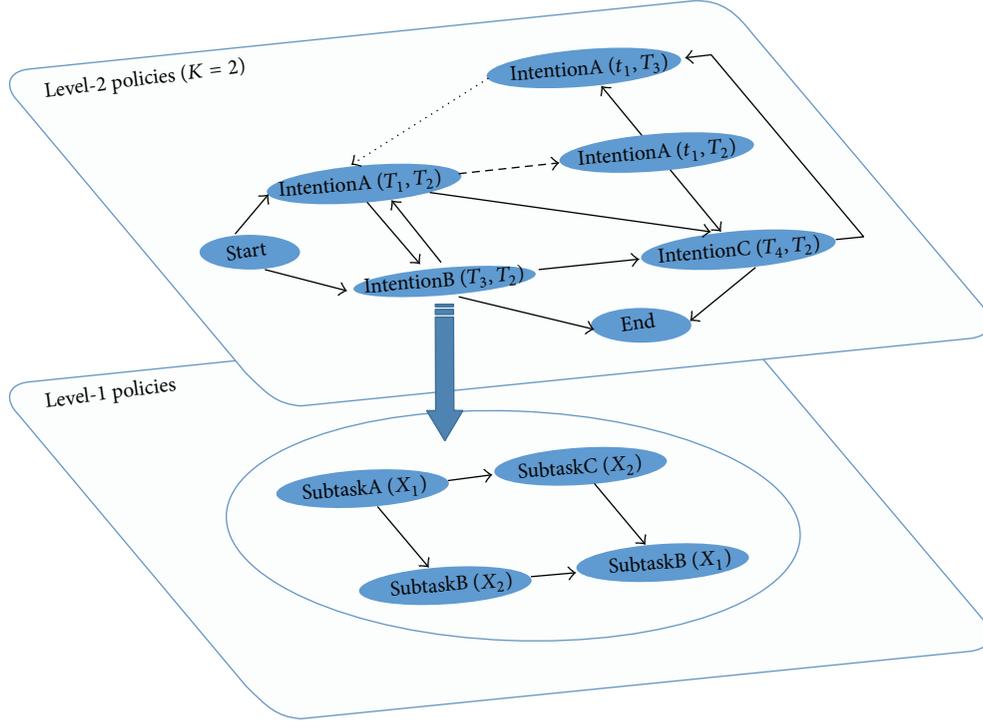


FIGURE 7: An example of team policies represented by a two-layer LHM.

when $\{\tilde{\mathbf{x}}_t, \tilde{\mathbf{y}}_t\}$ is a Markov system, $q(\mathbf{x}_t | \tilde{\mathbf{x}}_{t-1}, \tilde{\mathbf{y}}_t) = q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$, and formula (4) can be simplified into

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_t^i) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{y}_t)}. \quad (6)$$

When we choose optimal importance distribution $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$, $w_t^i \propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_{t-1}^i)$; however, it will be very hard to compute $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$ in our models. Thus, we use a simplest importance distribution $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t | \mathbf{x}_{t-1})$, and $w_t^i \propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i)$ in this case. After normalizing w_t^i , we need to resample the particles to solve the problem of particle degeneration. This process will abandon the particles with small weight and copy particles with large weight. The weights of the new particles after resampling will be set equal and we can compute the poster probability of the system state by using formula (1). Other details about this process can be found in [38].

4.2. Logical Particle Filtering in the LHSMM. The standard PF cannot be applied to recognize intentions in our LHSMM, since it does not allow logical transitions and instantiation process. In this section, we presented a logical particle filtering based on logical definitions and dependency of our model. A particle \mathbf{x}_t^i in a LHSMM is a tuple $\{\pi_t^K, d_t, e_t^K, \pi_t^{K-1}, e_t^{K-1}, \dots, \pi_t^0, e_t^0, S_t\}$, where π_t^K is an intention, and observation of a \mathbf{x}_t is O_t . Since O_t only depends on S_t , $p(O_t | \mathbf{x}_t) = p(O_t | S_t)$. To sample \mathbf{x}_t^i from \mathbf{x}_{t-1}^i , we use

the simplest importance distribution and forward sampling. Steps of LPF are shown as follows.

Step 1 (initialization). Set time $t = 0$, $w_0^i = 1/N_s$, $A_0^{K,i} = \text{Start}$, and $e_0^{K:1,i} = 1$; sample S_0^i from a prior distribution $p(S_0)$ for each particle.

Step 2 (forward sampling). Set $t = t + 1$, sample elements in each particle \mathbf{x}_t^i ($i = 1, 2, \dots, N_s$). Based on dependency in the DBN structure, the forward sampling process can be further decomposed into five steps: sampling intentions and duration, policies, primitive actions, the world state, and termination variables. The pseudocode is shown in Pseudocode 1.

There are two points to be noted: one is that when an abstract policy is selected and executed for the first time, it should update the alphabets, instantiation functions, and logical transitions in its corresponding lower level LMC. However, when a policy has not been terminated, elements in LMC do not need to be changed, and those in the top level LMC will keep; the other point is that when sampling an abstract policy from logical transitions, we must follow the specific logical transition and unified logical transition if conditions are satisfied and then get the new abstract policy by conditional probabilistic logical transition immediately.

Step 3 (updating weights and resampling). Update the weight of each particle by $w_t^i = w_{t-1}^i \cdot p(o_t^i | s_t^i)$, and get new set of particles $\{\mathbf{x}_t^i\}_{i=1}^{N_s}$ by resampling and set $w_t^i = 1/N_s$.

```

%%Sampling intentions and duration
If  $e_{t-1}^{K,i} = 0$ 
   $\pi_t^{K,i} = \pi_{t-1}^{K,i}$ ,  $d_t^i = d_{t-1}^i - 1$ 
Else
  If  $A_{t-1}^{K,i} = \text{Start}$ 
    Sample  $A_t^{K,i}$  from  $\Sigma^K$  by  $\gamma_t(A^K)$ 
  Else
    Sample  $A_t^{K,i}$  from  $\Sigma^K$  by  $\Delta_t(A_t^K | A_{t-1}^{K,i}, C(A_{t-1}^{K,i}))$ 
  End
  Instantiate  $A_t^{K,i}$  to  $\pi_t^{K,i}$  using  $\mu(\pi_t^K | A_t^{K,i}, S_{t-1}^i)$ 
  Sample  $d_t^i$  from  $p(d | \pi_t^{K,i})$ 
End
%%Sampling policies
From  $k = K - 1$  to  $k = 1$ 
  If  $e_{t-1}^{k,i} = 0$ ,
     $\pi_t^{k,i} = \pi_{t-1}^{k,i}$ 
  else
    If  $e_{t-1}^{k+1,i} = 0$ 
      Sample  $A_t^{k,i}$  from  $\Sigma^k(A_t^{k+1,i})$  by  $\Delta_t(A_t^k | A_{t-1}^{k,i}, C(A_{t-1}^{k,i}))$ ,
    Else
      Sample  $A_t^{k,i}$  from  $\Sigma^k(A_t^{k+1,i})$  by  $\gamma_t(A_t^k | A_t^{k+1,i})$ 
    End
    Instantiate  $A_t^{k,i}$  to  $\pi_t^{k,i}$  using  $\mu(\pi_t^k | A_t^{k,i}, S_{t-1}^i)$ .
  End
%%Sampling primitive actions
Sample  $A_t^{0,i}$  from  $\Sigma^0$  by  $p(A_t^0 | A_t^{1,i}, S_{t-1}^i)$ 
Instantiate  $\pi_t^{0,i}$  by  $\mu(\pi_t^0 | A_t^{0,i}, S_{t-1}^i)$ 
%%Sampling the world state
Sampling  $S_t^i$  by  $p(S_t | \pi_t^{0,i}, S_{t-1}^i)$ 
%%Sampling termination variables
If  $d_t^i < 1$ 
   $e_t^{K,i} = 1$ 
Else
  Sample  $e_t^{K,i}$  from  $p(e_t^K | \pi_t^{K,i}, S_t^i)$ 
End
From  $k = K - 1$  to  $k = 1$ 
  If  $e_t^{k+1,i} = 1$ 
     $e_t^{k,i} = 1$ 
  Else
    Sample  $e_t^k$  from  $p(e_t^k | \pi_t^{k,i}, S_t^i)$ 
  End

```

PSEUDOCODE 1: Pseudocode of sampling a particle in the LHSMM.

Step 4 (computing poster distribution of intentions). In this paper, we only care about the abstract intention and the constants in the instantiated intention. Since they are discrete, we can compute them by $p(A_t^K) = (1/N_s) \sum_{i=1}^{N_s} I_0(A_t^{K,i} - A_t^K)$ and $(\text{Constant}_t^K) = (1/N_s) \sum_{i=1}^{N_s} I_0(\text{Constant}_t^{K,i} - \text{Constant}_t^K)$, where $I_0(\cdot)$ is an indicator function and Constant_t^K is a possible constant in π_t^K . If the simulation does not end, go to Step 2.

With the four steps above, we can compute the probabilities of the team working mode and goals of agents at each time.

5. Experiments

5.1. Background. To evaluate the performance of applying the LHSMM and LPF in team intention recognition, we design a battle scenario. In this scenario, two agents execute an attacking mission individually or cooperatively in a known environment. Their team intention consists of two parts: the team working mode and the specific target of each agent. Our recognition is to compute the probabilities of the team intention sequentially according to the continuous observation of agent traces.

There are some characters of this scenario: first, the agents can act individually or constitute a team; second,

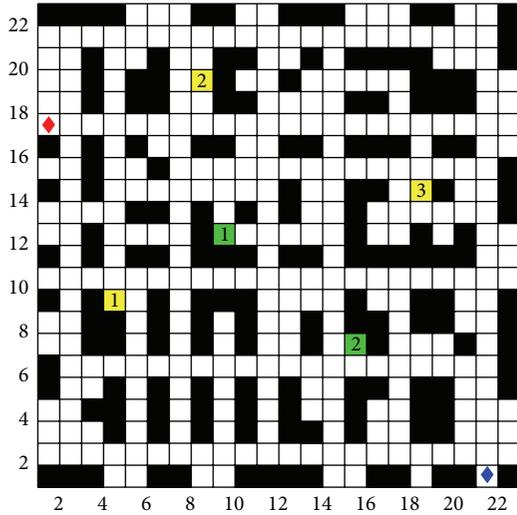


FIGURE 8: The initial situation of the battlefield.

the attacking mission can be decomposed into subtasks and primitive actions; third, the team intentions can be interrupted because of new orders or other unknown reasons; last, the observed traces have noise and we may not have any position record at all at some time; Furthermore, the observed data is got sequentially, and we need to update our recognition results when new evidence arrives. The initial situation of the battlefield is shown in Figure 8.

The battlefield map consists of 22×22 grids; the blue and red diamond points are the initial positions of agent A and agent B, respectively; the black points indicate buildings which the agents cannot get through; the two green grids are assembling positions; the three yellow grids are targets which may be attacked by agents; the white grids make up passable ways for motion. In each time step, the agent can move to an adjacent blank grid or stay in its current position. Before inferring intentions, we give a decomposed LHMC representation of the team policies in this scenario, as is shown in Figure 9.

Figure 9(a) depicts the logical Markov chain in the top level ($K = 2$); there are two relation symbols: (a) Attack.C (T_1) means the two agents will attack cooperatively, and their common target is T_1 ; (b) Attack.I (T_1, T_2) means they will execute their missions individually; that is, agent A attacks T_1 and agent B attacks T_2 . Variables T_1 and T_2 represent the targets, and the possible value could be t_1 , t_2 , or t_3 . We also depict the initial distribution of level- K LMC in Figure 9(a). Figures 9(b) and 9(c) depict the LMCs under Attack.C (T_1) and Attack.I (T_1, T_2), respectively. In level-1, there are two relation symbols; they are (a) Assemble (As), which means agent A and agent B are trying to get together at the assembling position As and (b) Destroy (T_1, T_2) which means agent A is going to destroy T_1 and agent B is going to destroy T_2 . As is a variable, and its possible value could be as_1 or as_2 . T_1 and T_2 have been instantiated in the top level. According to transitions in Figures 9(b) and 9(c), the agents will first assemble at a point and then go to destroy the target together, when they attack a target cooperatively. But when

Attack.I (T_1, T_2) is executed, agents have different targets and do not affect each other.

The team has only one abstract primitive action Move (Direction A, Direction B), where Direction A and Direction B indicate the moving directions of agents A and B, respectively, and there are 5 possible directions: north, south, east, west, and null. If the direction is instantiated as null, the agent will stay at the current position. Selection and instantiation of abstract primitive action depend on the level-1 policy and the previous state.

In this scenario, the selection of policies and actions is only related to the positions of agents. Thus, the world state is defined as the set of $\text{pos}(\text{agent A})$ and $\text{pos}(\text{agent B})$, where $\text{pos}(X)$ returns the current grid of X . The observations are functions of them which will be introduced when we explain the observation model.

5.2. Settings

5.2.1. *Transition Probabilities in LMCs.* There is only one possible logical transition in level-1 and the transition probability is always 1, and the conditional transition probabilities of level- K LMC are shown in

$$\begin{aligned}
 & C \mapsto \begin{cases} \text{if } \text{IsOccupied}(T_1) \wedge \text{IsOccupied}(T_2), & \begin{cases} p_{CD} = 1, \\ p_{CB} = 0, \\ p_{CE} = 0, \end{cases} \\ \text{else,} & \begin{cases} p_{CD} = 0, \\ p_{CB} = 0.4, \\ p_{CE} = 0.6, \end{cases} \end{cases} \quad (7) \\
 & B \mapsto \begin{cases} \text{if } \text{IsOccupied}(T_1), & \begin{cases} p_{BD} = 1, \\ p_{BC} = 0, \\ p_{BF} = 0, \end{cases} \\ \text{else,} & \begin{cases} p_{BD} = 0, \\ p_{BC} = 0.4, \\ p_{BF} = 0.6. \end{cases} \end{cases}
 \end{aligned}$$

B, C, D, E, and F represent policies shown in Figure 9(a), $\text{IsOccupied}(T_1)$ means that the target T_1 is occupied, and this proposition will be true when any agent reaches the grid of T_1 . p_{CB} is the probability of transiting from abstract policy C to abstract policy B. p_{CD} is the probability of terminating the simulation after executing abstract policy C, and other transition symbols have the similar meaning. The level- K LMC shows that the team working mode may alter between cooperation and independence, and the simulation can only end when one of the targets is destroyed.

5.2.2. *Policy Termination.* In our LHSM, termination of policy π_t^k depends on the probability $p(e_t^k | \pi_t^k, S_t)$. When π_t^k is Attack.C (T_1), we set $p(e_t^k = 1 | \pi_t^k, S_t) = 1$ for all S_t which satisfy that the positions of the two agents and target T_1 are

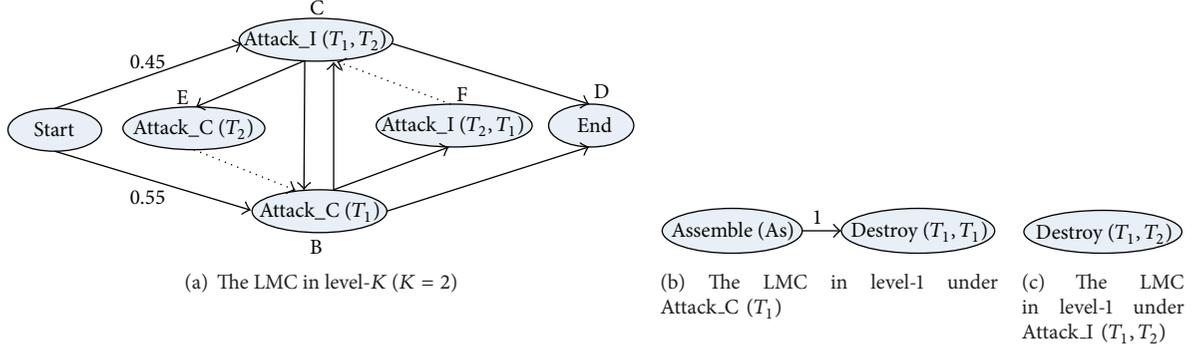


FIGURE 9: A decomposed LHMC representation of the team policies.

all the same, and $p(e_t^K = 1 \mid \pi_t^K, S_t) = 0$ for other S_t . When π_t^K is Attack.I (T_1, T_2), we set $p(e_t^K = 1 \mid \pi_t^K, S_t) = 1$ for all S_t which satisfy that agent A reaches target T_1 or agent B reaches target T_2 , and $p(e_t^K = 1 \mid \pi_t^K, S_t) = 0$ for other S_t .

When π_t^K is Assemble (As), we set $p(e_t^K = 1 \mid \pi_t^K, S_t) = 1$ for S_t which satisfy that the positions of the two agents and assembling point As are all the same, and $p(e_t^K = 1 \mid \pi_t^K, S_t) = 0$ for other S_t . When π_t^K is Destroy (T_1, T_2) or Destroy (T_1, T_1), we set $p(e_t^K = 0 \mid \pi_t^K, S_t) = 1$ for all S_t , because these two policies do not have outlines in their LMCs.

5.2.3. Selection and Instantiation of Primitive Actions. In this scenario, selection of primitive actions can be neglected since there is only one abstract primitive action. Instantiation function $\mu^0(\pi_t^0 \mid A_t^0, S_{t-1})$ defined by π_t^1 is set as follows:

- (1) When π_t^1 is Destroy (T_1, T_2) or Destroy (T_1, T_1), agent A will move in the direction which can make it on the shortest way to its instantiated target; if there are two directions satisfying this condition, agent A will choose one with a probability 0.5, and so will agent B.
- (2) When π_t^1 is Assemble (As) and S_{t-1} shows that both agent A and agent B have not reached the assembling position, agent A will move along with the direction which ensures it moving on the shortest way to As; if there are two directions satisfying this condition, agent A will choose one with a probability 0.5, and so will agent B.
- (3) When π_t^1 is Assemble (As) and S_{t-1} shows that one agent has reached the assembling position, but the other one has not, the agent who is still on its way will move on the shortest way to As; if there are two directions satisfying this condition, this agent will choose one with a probability 0.5, and the other agent will choose the direction as null.

5.2.4. The Duration Model. After the abstract intention Attack.C (T_1) or Attack.I (T_1, T_2) keeps for time d , an interruption event will happen with a probability $p(d) = \int_{d-1}^d f(x)dx$, $d = 1, 2, \dots, ds$, ds is the necessary time to realize the intention which depends on the state when the

intention begins, $f(x)$ is the probability density function of the lognormal distribution $\text{lognorm}(\mu, \sigma, \theta)$, and

$$f(x) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(x-\theta)} \exp\left(-\frac{(\log(x-\theta)-\mu)^2}{2\sigma^2}\right), & x > \theta, \\ 0, & x \leq \theta. \end{cases} \quad (8)$$

We set $\mu = 25$, $\sigma = 100$, and $\theta = 0$.

5.2.5. Observation Model. Observation at one time is a set $\{\text{obs}(\text{pos}(\text{agent A})), \text{obs}(\text{pos}(\text{agent B}))\}$, where $\text{obs}(G)$ is a function which returns the true grid G with a probability 0.6, returns null with a probability 0.2, and returns a false grid $G' \neq G$ with a probability $0.2 \cdot w_{GG'}$; $w_{GG'}$ is computed by

$$w_{GG'} = \begin{cases} \text{if } G' \in \mathbf{Go}, & 0, \\ \text{if } G' = G, & 0, \\ \text{else,} & \frac{((x_G - x_{G'})^4 + (y_G - y_{G'})^4)^{-1}}{\sum_{G'' \in \mathbf{Gs}} ((x_G - x_{G''})^4 + (y_G - y_{G''})^4)^{-1}}, \end{cases} \quad (9)$$

where \mathbf{Go} is the set of grids containing buildings and \mathbf{Gs} is the set of all grids except G and grids in \mathbf{Go} . x_G and y_G represent the horizontal and vertical coordinates of grid G , respectively. When $\text{obs}(G)$ returns null, we will have no information about the observed agent, and the observations of two agents are independent.

5.2.6. Other Settings. To simplify the instantiation functions of intentions, the specific targets are selected according to their tactical values. We set the normalized values of t_1, t_2 , and t_3 as 0.3, 0.4, and 0.3, respectively. The agents will instantiate the target variable in abstract intention one by one. The probability of choosing a target is proportionate to its tactical value. For example, when we instantiate a variable T_1 whose possible constants are t_2 and t_3 (we have known the value of T_2 is t_1), the probability of instantiating T_1 as t_2 will be $0.4/(0.4 + 0.3) = 0.5714$. For instantiation function in level-1, there is only one variable As; the probability of its value depends on the value of T_1 , as is shown in Table 1.

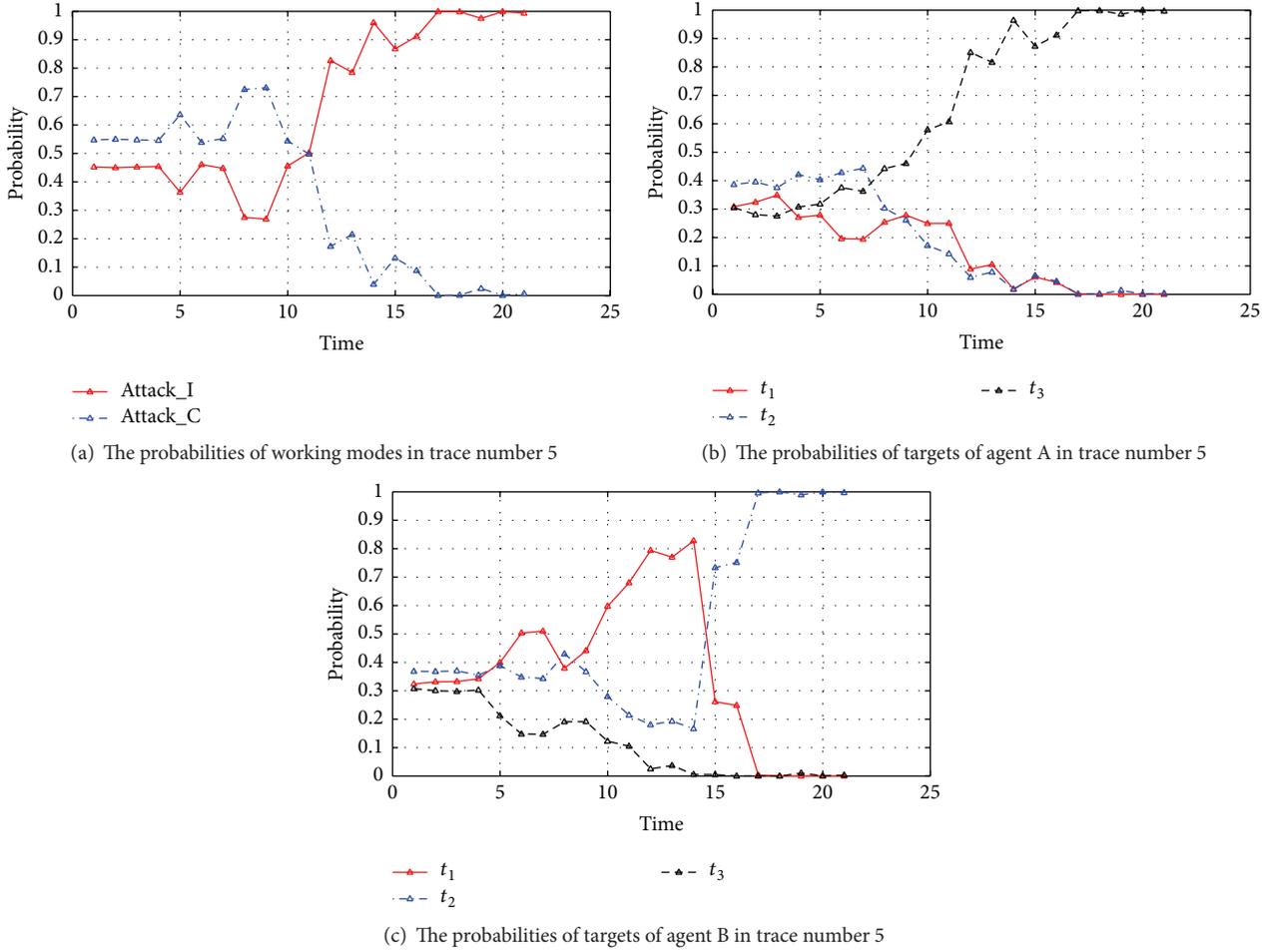


FIGURE 10: The recognition results of trace number 5 computed by the LHHSM.

TABLE 1: The instantiation probabilities of As given value of T_1 .

As	T_1		
	t_1	t_2	t_3
as ₁	0.4	0.6	0.7
as ₂	0.6	0.4	0.3

In the approximate inference, we set the number of particles $N_s = 2000$.

5.3. Results and Discussion. We run the scenario repeatedly and produce a test dataset consisting of 100 traces. With this data set, we compute the recognition results of specific traces to validate the LHHSM and the LPF and compare the performances when intention duration distributions are different.

5.3.1. The Recognition Results of Specific Traces. Three traces in test data set are selected to compute the probabilities of team intentions by LHHSM. The details of these three traces are shown in Table 2.

As shown in Table 2, agents in trace number 5 execute the attacking mission individually: the target of agent A is t_3 and the target of agent B is t_2 . For trace number 5 there is no interruption and the mission is completed at $t = 21$. In trace 17, the agents choose the cooperation working mode and do not change their intention during mission either. They occupy the target t_1 successfully at $t = 36$. Trace number 57 is a bit more complicated. The agents decide to attack t_3 and t_2 individually first, but they change the intention and attack t_2 together at $t = 14$. They successfully assemble and go to t_2 together. However, before they reach t_2 , their intention is interrupted again and agent A attacks t_3 alone. This intention is terminated at $t = 44$ because agent B occupied t_2 . We use LHHSM to recognize both team working modes and intentions in these three cases, and the results are shown in Figures 10, 11, and 12.

Figure 10(a) shows the probabilities of working modes in trace number 5. Even though the prior probability of the real working mode, that is, Attack-I (T_1, T_2), is lower in the initial phase, it increases very fast, and the value exceeds 0.8 at $t = 12$. The reason of the shake at $t = 8$ is that the agent B is turning left at a cross, but there is noise in the observation of agent B at this time. Figures 10(b) and 10(c)

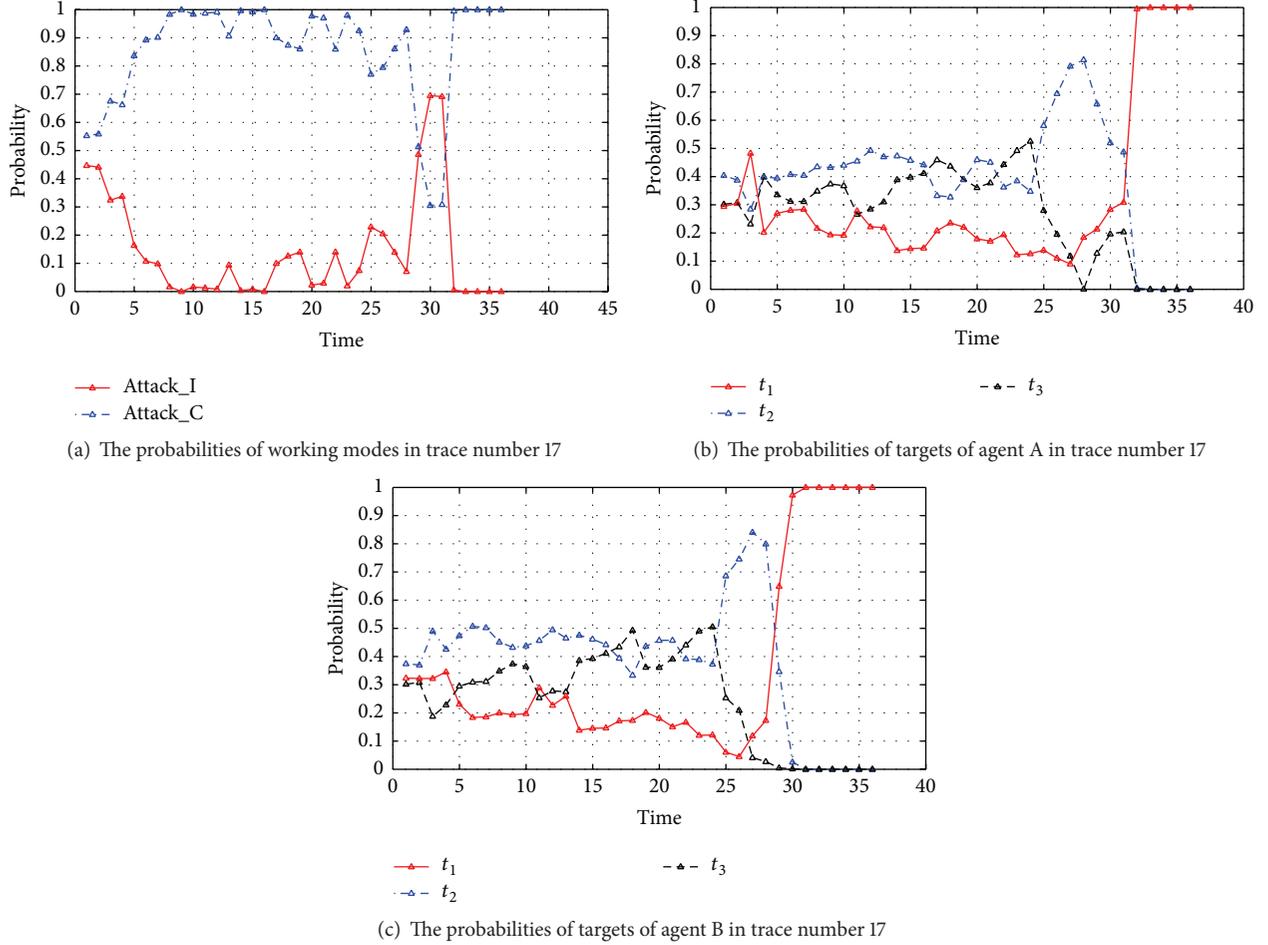


FIGURE 11: The recognition results of trace number 17 computed by the LHHMM.

TABLE 2: The details of three traces.

Trace number	Durations	Working modes	Targets	Interrupted
5	$t \in [1, 21]$	Attack_I (T_1, T_2)	agent A: t_3 agent B: t_2	No
17	$t \in [1, 36]$	Attack_C (T_1)	agent A: t_1 agent B: t_1	No
57	$t \in [1, 14]$	Attack_I (T_1, T_2)	agent A: t_3 agent B: t_2	Yes
57	$t \in [15, 41]$	Attack_C (T_1)	agent A: t_2 agent B: t_2	Yes
57	$t \in [42, 44]$	Attack_I (T_1, T_2)	agent A: t_3 agent B: t_2	No

show the probabilities of targets of agents in trace number 5. In Figure 10(b), our models recognize the target of agent A very well, and the probability of target t_3 increases very fast. In Figure 10(c), the probability of t_1 is larger than that of the real target t_2 from $t = 8$ to $t = 13$, because agent B chooses a path which is indispensable to reach t_1 in this period. When

agent B leaves this path at $t = 13$, the probability of the real target is higher again.

Figure 11(a) shows the probabilities of working modes in trace number 17. Except for the shaking near $t = 30$, the probability of the real working mode is always high. The reason of the failure is that some observations are missing in that period. At the same time, the observed noisy data support that the working mode has been interrupted. Fortunately, the probability of Attack_C (T_1) recovers fast when new evidences arrive. Figures 11(b) and 11(c) show the probabilities of targets of agents in trace number 17. Since agent A and agent B have the same target, the curves in Figures 11(b) and 13(c) are very similar. The reason of the confusion before $t = 25$ is that the agents are assembling and their traces cannot provide new information about their targets; we can find that, even the prior probability of t_2 is quite higher, the red curve increase fast.

Figure 12(a) shows the probabilities of working modes in trace number 57. Generally, our models can recognize the working modes well even if it is interrupted at $t = 14$ and $t = 41$. The reason of two shakes near $t = 10$ and $t = 35$ is that the position of agent A is near the cross but the

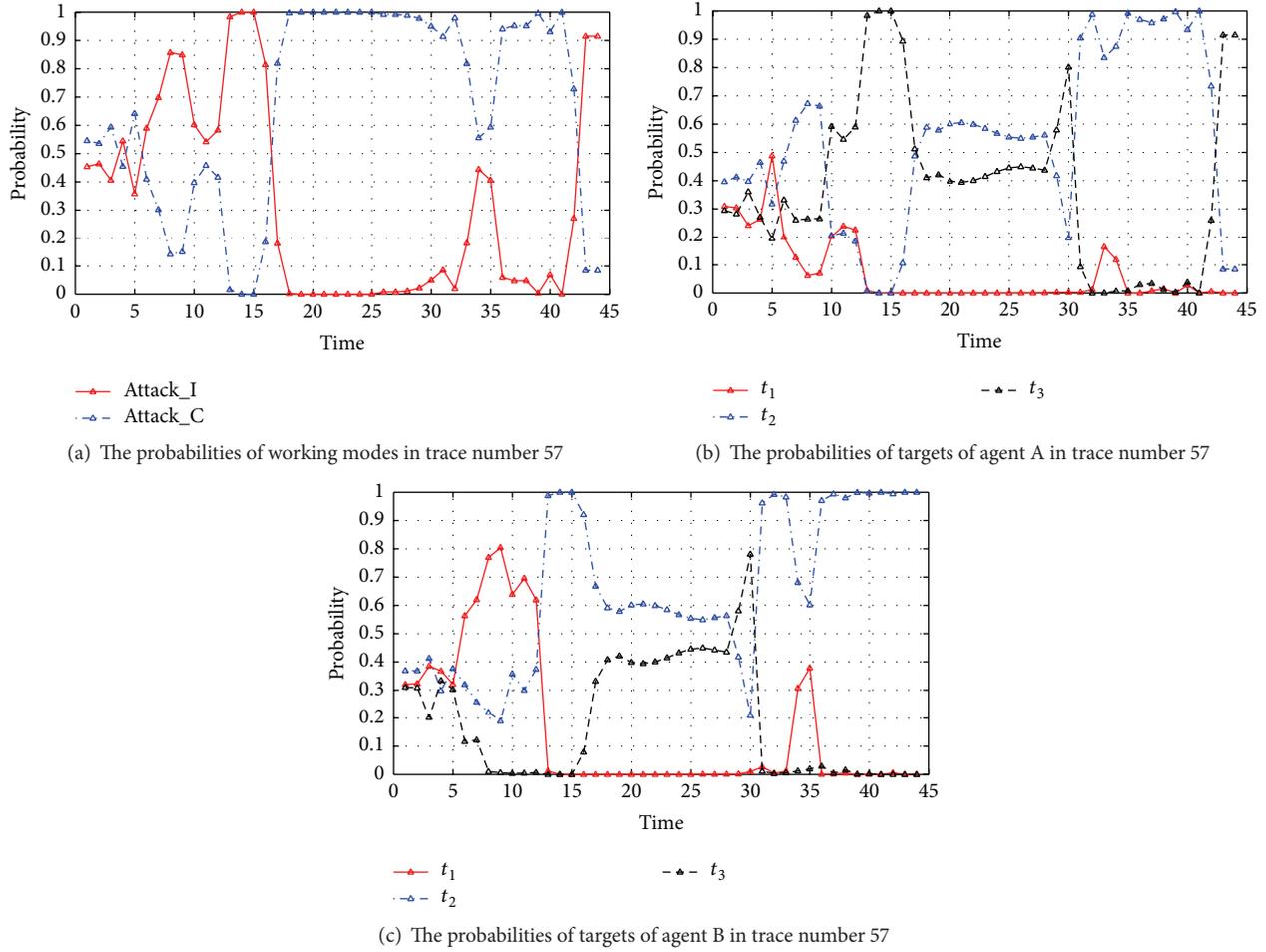


FIGURE 12: The recognition results of trace number 57 computed by the LHSMM.

observation is not accurate. Figures 12(b) and 12(c) show the probabilities of targets of agents in trace number 57. We can find that the results after $t = 14$ can reflect the real results well except some shake; even the target of agent A changes twice. However, the real targets are not the highest in most case before $t = 14$. In Figure 12(b), the probability of t_2 is generally higher before $t = 8$, because agent A and agent B are on the indispensable paths to t_2 and t_1 , respectively. After they leave the indispensable paths, the probability of t_3 which is the real target of agent A increases a lot, which makes the target of agent B much like t_2 , since their working mode is attacking individually. We can also find that although the probabilities of the real targets are not the highest during this period, they are not the lowest at least.

5.3.2. Comparison of the LHSMM and a Modified LHHMM. To show the advantages of LHSMM compared with the LHHMM, we add our observation model to the standard LHHMM (we still call it the LHHMM later) and also use a particle filtering to infer intentions. However, since the LHHMM does not model intention interruption, the weights of all particles may be 0 after an interruption happens. In this situation, we reset the particle weights to $1/N_s$. Then,

inference can be continued to update the probabilities of intentions until the end of the simulation.

The performances of our model and the LHHMM are compared statistically: their performances are evaluated by three metrics: precision, recall, and F -measure. These metrics are computed by

$$\begin{aligned} \text{precision} &= \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TI_i}, \\ \text{recall} &= \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TT_i}, \end{aligned} \quad (10)$$

$$F\text{-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where N is the number of possible classes, TP_i , TI_i , and TT_i are the true positives, total of true labels, and total of inferred labels for class i , respectively. Formulas (10) show that precision is used to scale the reliability of the recognized results; recall is used to scale the efficiency of the algorithm applied in the test data set; and F -measure is an integration of precision and recall. We can find that the value of all these

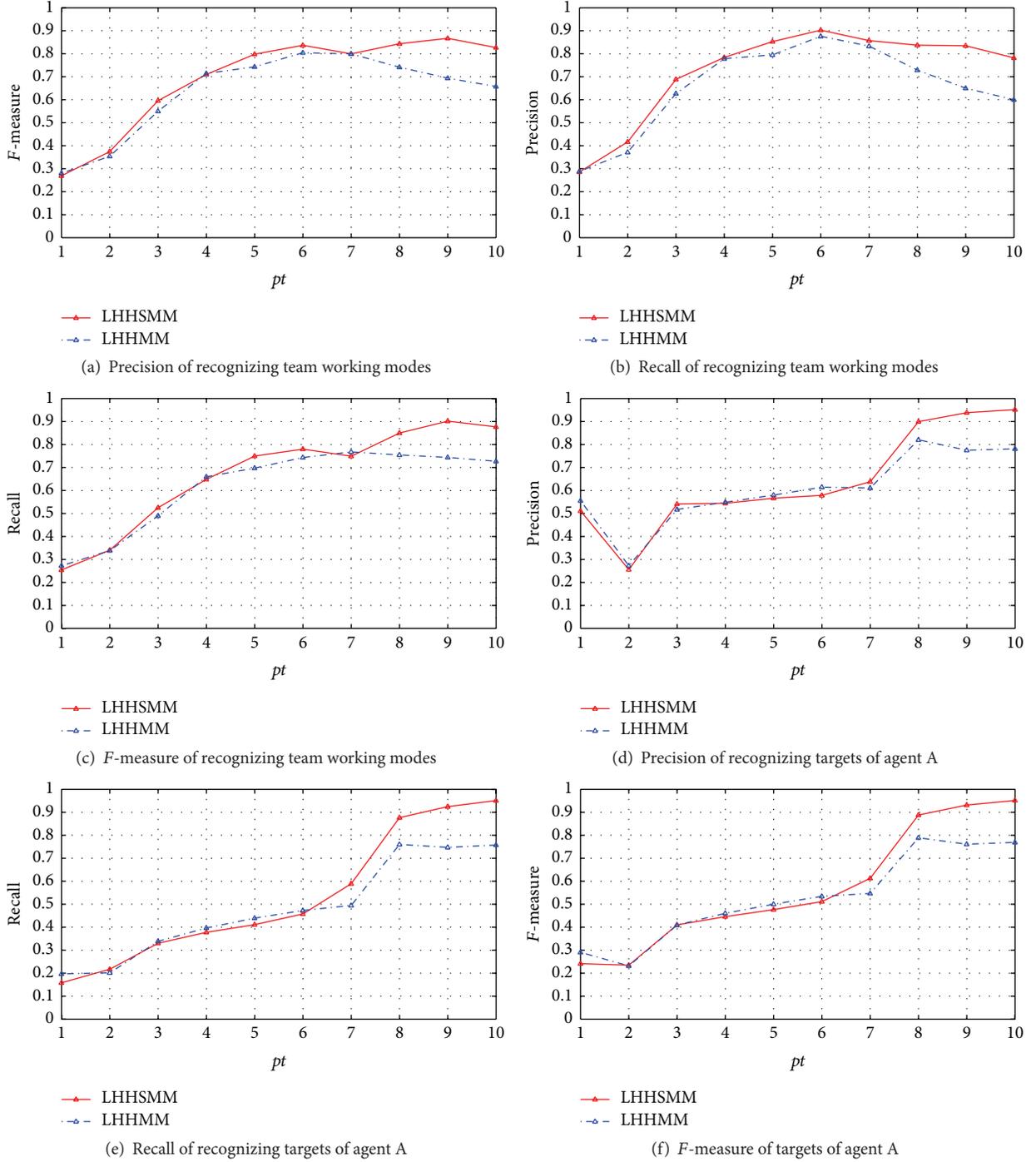


FIGURE 13: Metrics of recognizing team working modes and targets of agent A computed by the LHHMM and the LHSMM.

metrics will be between 0 and 1, and a higher metric means a better performance.

Since the traces in the dataset have different time lengths, we define a variable pt which is a positive integer. $pt \in [1, 10]$, and the metrics at pt means that their recognizing object set is $\{\text{object}_{t=\lceil pt * \text{length}^k / 10 \rceil}^{k=1:100}\}$, where k is the index of the trace and length^k is the length of the trace number k . Thus, metrics with different pt show the performances of algorithms in different phases of simulation. To give the final recognition results,

we further define two thresholds $\alpha = 0.1$ and $\beta = 0.05$. When $p(\text{mode A}) - p(\text{mode B}) > \alpha$, we can regard the recognition result as mode A, where mode A and mode B are the two working modes; otherwise, the result is unknown. When $p(\text{target A}) - p(\text{target B}) > \beta$ and $p(\text{target A}) - p(\text{target C}) > \beta$, we can regard the recognition result as target A, where target A, target B, and target C are the three targets; otherwise, the result is unknown. The three metrics of recognizing working modes and targets of agent A computed by the LHHMM and the LHSMM are shown in Figure 13.

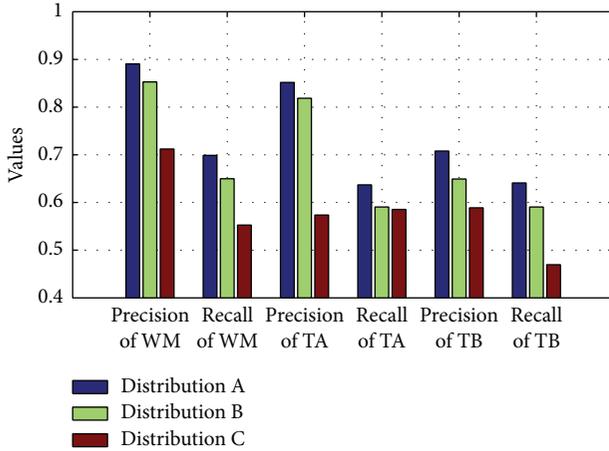


FIGURE 14: Recognition metrics with different duration distributions.

The red solid curves and the blue dash dotted curves are computed by the LHHSMM and the LHHMM, respectively. We can find that the LHHSMM outperforms the LHHMM in all the three metrics, especially in the second half of the simulation. In the starting phase, the intention just keeps for a short time and the probability of changing intentions is not large. Thus, the LHHSMM and the LHHMM have similar performances. However, with more and more interruptions of intention occurring, the LHHSMM shows its advantage: it will generally improve the recognition performance when there are more observations.

5.3.3. The Effects of Duration Modeling. Our LHHSMM is semi-Markov because it uses a specific distribution to model the duration that the team intention is not interrupted explicitly. To evaluate the effects of duration modeling, we compare recognition metrics computed by LHHSMMs with three different duration distributions: Distribution A is the real distribution of the working mode introduced in Section 5.2; Distribution B is a geometric distribution whose expectation is equal to that of $f(x)$ in Distribution A; Distribution C is the same as Distribution A except that $\mu = 45$. The test dataset is a subset of the 100 traces, which consists of all of 51 traces where the working mode is changed at least once. Besides, we only evaluate the recognition results of the second half of these traces, because, in the beginning phases of the traces, the lack of evidences may affect the recognizing performances. The recognition results are still evaluated by the metrics precision and recall. Because we do not need to show the metrics at different phases, the recognizing object set is not with the parameter pt , and we recognize the working modes (WM), targets of agent A (TA), and targets of agent B (TB) at all steps of the second half of the 51 traces. Recognition metrics with different duration distributions are shown in Figure 14.

The blue, green, and red bars indicate the performances of our LHHSMMs with Distribution A, Distribution B, and Distribution C, respectively. Obviously, the comparison results of six terms are the same: (a) Distribution A

performs the best, which shows that the type of duration distribution is important to recognize the working mode and the target of each agent; (b) Distribution B performs better than Distribution C, which shows that the expectation of the duration distribution has a large impact on the recognizing results. These comparisons prove that a precise duration modeling of the working mode is necessary and our semi-Markov framework is effective in team intention recognition.

6. Conclusions and Future Works

In this paper, we proposed a LHHSMM to recognize team intentions. As a fusion of the LHHMM, LHSM, and MDP, the LHHSMM possesses advantages to solve the team intention recognition problems in a complex environment: first, it uses a logical predicate to represent the team working mode, which can be recognized together with each goal of agents; second, it has a hierarchical structure and can make use of domain knowledge to present complex tasks; third, due to the modeling of intention duration, LHHSMM can update the probabilities of results correctly even if the intention is interrupted and changed; last, LHHSMM can deal with noisy and partially missing observations.

To solve inference problems of the LHHSMM, a LPF is proposed based on logical definitions and the dependency of the LHHSMM. We also design a combat scenario to evaluate the LHHSMM and LPF; the results show the following: first, no matter intentions are changed or not, our methods can effectively recognize the team working mode and the targets of each agent; second, the LHHSMM outperforms the LHHMM in precision, recall, and F -measure in case that intentions are interrupted within a high probability; last, an explicit duration modeling of the team working mode is effective in team intention recognition.

In the future, we would like to continue our research on two aspects: (a) to learn parameters in the LHHSMM and (b) to discuss how to get the optimal importance distribution in the LPF. Moreover, applying our model to recognize intentions in a real scenario is also absorbing.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

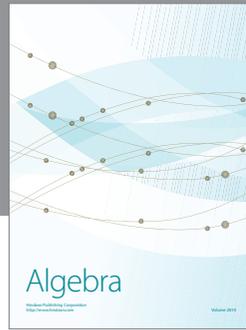
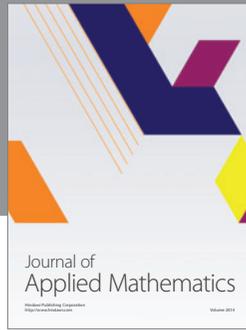
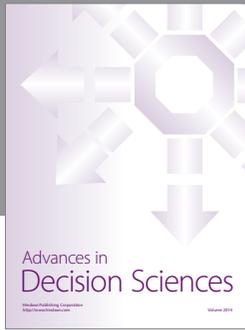
The work described in this paper is sponsored by the National Natural Science Foundation of China under Grants no. 61473300 and no. 61573369.

References

- [1] F. Sadri, "Logic-based approaches to intention recognition," in *Handbook of Research on Ambient Intelligence: Trends and Perspectives*, pp. 346–375, IGI Global, 2011.
- [2] G. Synnaeve and P. Bessière, "A Bayesian model for plan recognition in RTS games applied to starcraft," in *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '11)*, pp. 79–84, October 2011.

- [3] S. Z. Elloumi, V. Roy, J. Marmorat, and N. Maïzi, "Securing harbors by modeling and classifying ship behaviour," in *Proceedings of the 20th Annual Conference on Behavior Representation in Modeling and Simulation (BRiMS '11)*, pp. 114–121, Sundance, Utah, USA, March 2011.
- [4] L. R. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.
- [5] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.
- [6] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.
- [7] H. H. Bui, S. Venkatesh, and G. West, "Policy recognition in the abstract hidden Markov model," *Journal of Artificial Intelligence Research*, vol. 17, pp. 451–499, 2002.
- [8] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artificial Intelligence*, vol. 173, no. 7-8, pp. 830–856, 2009.
- [9] K. Kersting, S. Natarajan, and D. Poole, "Statistical relational AI: logic, probability and computation," in *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR '11)*, pp. 1–9, Vancouver, Canada, May 2011.
- [10] K. Kersting, L. De Raedt, and T. Raiko, "Logical hidden Markov models," *Journal of Artificial Intelligence Research*, vol. 25, pp. 425–456, 2006.
- [11] Y.-B. Zha, S.-G. Yue, Q.-J. Yin, and X.-C. Liu, "Activity recognition using logical hidden semi-Markov models," in *Proceedings of the IEEE 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP '13)*, pp. 77–84, Chengdu, China, December 2013.
- [12] S. Natarajan, H. H. Bui, P. Tadepalli, K. Kersting, and W.-K. Wong, "Logical hierarchical hidden Markov models for modeling user activities," in *Inductive Logic Programming: 18th International Conference, ILP 2008 Prague, Czech Republic, September 10–12, 2008 Proceedings*, vol. 5194 of *Lecture Notes in Computer Science*, pp. 192–209, Springer, Berlin, Germany, 2008.
- [13] S. G. Yue, Y. B. Zha, Q. J. Yin, and L. Qin, "Multi-agent intention recognition using logical hidden semi-Markov models," in *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pp. 701–708, August 2014.
- [14] C. F. Schmidt, N. S. Sridharan, and J. L. Goodson, "The plan recognition problem: an intersection of psychology and artificial intelligence," *Artificial Intelligence*, vol. 11, no. 1, pp. 45–83, 1978.
- [15] T. A. Han and L. M. Pereira, "State-of-the-art of intention recognition and its use in decision making," *AI Communications*, vol. 26, no. 2, pp. 237–246, 2013.
- [16] H. A. Kautz and J. F. Allen, "Generalized plan recognition," in *Proceedings of the 5th National Conference on Artificial Intelligence*, pp. 32–37, Philadelphia, Pa, USA, August 1986.
- [17] D. Avrahami-Zilberbrand and G. A. Kaminka, "Fast and complete symbolic plan recognition," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pp. 79–84, Edinburgh, Scotland, August 2005.
- [18] D. W. Albrecht, I. Zukerman, A. E. Nicholson, and A. Bud, "Towards a Bayesian model for keyhole plan recognition in large domains," in *Proceedings of the 6th International Conference on User-Modeling*, pp. 365–376, Sardinia, Italy, June 1997.
- [19] D. V. Pynadath, *Probabilistic grammars for plan recognition [Ph.D. thesis]*, University of Michigan, Ann Arbor, Mich, USA, 1999.
- [20] E. W. Dereszynski, J. Hostetler, A. Fern, T. G. Dietterich, T. T. Hoang, and M. Udarbe, "Learning probabilistic behavior models in real-time strategy games," in *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 20–25, 2011.
- [21] J. Yin, D. H. Hu, and Q. Yang, "Spatio-temporal event detection using dynamic conditional random fields," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1321–1327, Pasadena, Calif, USA, July 2009.
- [22] D. Avrahami-Zilberbrand and G. A. Kaminka, "Hybrid symbolic-probabilistic plan recognizer: initial steps," in *Proceedings of the AAAI Workshop on Modeling Others from Observations*, pp. 1321–1327, Boston, Mass, USA, July 2006.
- [23] C. W. Geib and R. P. Goldman, "A probabilistic plan recognition algorithm based on plan tree grammars," *Artificial Intelligence*, vol. 173, no. 11, pp. 1101–1132, 2009.
- [24] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [25] E. Y. Ha, J. P. Rowe, B. W. Mott, and J. C. Lester, "Goal recognition with Markov logic networks for player-adaptive games," in *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '11)*, pp. 32–39, October 2011.
- [26] N. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Boston, Mass, USA, 2009.
- [27] D. Masato, T. J. Norman, W. W. Vasconcelos, and K. Sycara, "Agent-oriented incremental team and activity recognition," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1402–1407, Barcelona, Spain, July 2011.
- [28] W. J. Mao, J. Gratch, and X. C. Li, "Probabilistic plan inference for group behavior prediction," *IEEE Intelligent Systems*, vol. 27, no. 4, pp. 27–36, 2012.
- [29] A. Pfeffer, S. Das, D. Lawless, and B. Ng, "Factored reasoning for monitoring dynamic team and goal formation," *Information Fusion*, vol. 10, no. 1, pp. 99–106, 2009.
- [30] S. Saria and S. Mahadevan, "Probabilistic plan recognition in multi-agent systems," in *Proceedings of the 24th International Conference on Automated Planning and Scheduling*, pp. 287–296, Portsmouth, NH, USA, June 2004.
- [31] K. Gaitanis, *Team behavior recognition using dynamic Bayesian networks [Ph.D. thesis]*, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium, 2008.
- [32] A. Sadilek and H. Kautz, "Location-based reasoning about complex multi-agent behavior," *Journal of Artificial Intelligence Research*, vol. 43, pp. 87–133, 2012.
- [33] B. Auslander, K. M. Gupta, and D. W. Aha, "Maritime threat detection using probabilistic graphical models," in *Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference (FLAIRS '12)*, pp. 2–7, May 2012.
- [34] S. Hladky and V. Bulitko, "An evaluation of models for predicting opponent positions in first-person shooter video games," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG '08)*, pp. 39–46, IEEE, Perth, Australia, December 2008.
- [35] F. Southey, W. Loh, and D. Wilkinson, "Inferring complex agent motions from partial trajectory observations," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, pp. 2631–2637, Hyderabad, India, January 2007.

- [36] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Activity recognition using semi-Markov models on real world smart home datasets," *Journal of Ambient Intelligence and Smart Environments*, vol. 2, no. 3, pp. 311–325, 2010.
- [37] H. H. Bui, S. Venkatesh, and G. West, "Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 177–196, 2001.
- [38] Y. Hu, P. Baraldi, F. D. Maio, and E. Zio, "A particle filtering and kernel smoothing-based approach for new design component prognostics," *Reliability Engineering and System Safety*, vol. 134, pp. 19–31, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

