Information Sciences 418–419 (2017) 13–33

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A framework of mining semantic-based probabilistic event relations for complex activity recognition

Li Liu[a,b,*], Shu Wang[c], Guoxin Su[d], Bin Hu[e,**], Yuxin Peng[f], Qingyu Xiong[b], Junhao Wen[b]

[a] Ministry of Education, Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing 400044, PR China
[b] School of Software Engineering, Chongqing University, Chongqing 400044, PR China
[c] Faculty of Materials and Energy, Southwest University, Chongqing 400715, PR China
[d] School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia
[e] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, PR China
[f] Department of Physical Education and Sports Science, Zhejiang University, Hangzhou 310028, PR China

ARTICLE INFO

Article history:
Received 18 October 2016
Revised 12 July 2017
Accepted 14 July 2017
Available online 15 July 2017

Keywords:
Activity model
Semantic-based representation
Probabilistic event relation learning
Pattern mining

ABSTRACT

Human activity recognition has become a key research topic in a variety of applications. Modeling activity events and their rich relations using high-level human understandable activity models such as semantic-based knowledge base hold promise. However, formulas in current semantic-based approaches are generally manually encoded, which is rather unrealistic in situations where event relations are intricate. Moreover, current approaches for learning event relations often lack the capability to handle uncertainties. To address these issues, we present a framework to learn an event knowledge base (EKB) of probabilistic interval-based event relations and use them to infer varied semantic-level queries about activity occurrences under uncertainty. Specifically, we formalize an activity model to represent eight temporal and hierarchical event relations and four commonly performed queries. We leverage pattern mining techniques to learn an EKB associated with these relations and queries in a unified way. Experimental results show that the proposed framework with the learned EKB involving temporal and hierarchical dependencies leads to a significant performance improvement on activity recognition, particularly in the presence of incomplete or incorrect observations.

1. Introduction

The maturity and prevalence of powerful sensors and high-speed processors facilitate advanced human activity recognition capabilities in a variety of applications. The data collected from these sensors is semantically rich, but scalable approaches to answer varied semantic-based queries are limited due to their inability to automatically generate representations of events that encode abstract temporal and hierarchical structure. Although *simple events* can often be inferred by sensor

* Corresponding author at: School of Software Engineering, Chongqing University, Chongqing 400044, PR China.
** Corresponding author.
E-mail addresses: dcsliuli@cqu.edu.cn (L. Liu), wangshu@lut.cn (S. Wang), guoxin@uow.edu.au (G. Su), bh@lzu.edu.cn (B. Hu), yxpeng@zju.edu.cn (Y. Peng), xiong03@cqu.edu.cn (Q. Xiong), jhwen@cqu.edu.cn (J. Wen).

http://dx.doi.org/10.1016/j.ins.2017.07.022
0020-0255/

data directly, many *complex activities*, which consist of temporally and coherently related events, require more structured models that are not easy to handcraft.

First of all, an activity model should account for the temporal structures of complex activities. Also, an activity model should convey hierarchical relations among activities, as it is found that most human behaviors are hierarchically structured [48] and the hierarchical structures are necessary for improving recognition performance [6]. Besides, owing to the diversity and complexity in human activities, the model is required for handling uncertainties over events and their temporal dependencies. Last, a useful activity model should be applied to answer varied semantic-level queries associated with temporal and hierarchical relations under *uncertainty*. For example, "if the event `drill ball before jump` is observed, is it highly likely that a player is doing the complex event `layup`?", or "if the event `catch ball` is observed, what is the possibility that a player will `jump` next?".

Recent works have tried to address the modeling and recognition of complex activities. They generally can be categorized into two kinds of approaches, namely, graphical models and semantic models. Among various recognition methodologies, graphical models–hidden Markov models, dynamic Bayesian networks and topic models–have become the most popular approaches for modeling and understanding human activities. Semantic models have also gained attention for addressing complex activity recognition problems in recent years. However, all these approaches face one or more of the following issues:

1. Most of the graphical-based approaches are adept at managing uncertainties, but not expressive enough to describe rich temporal relationships among activities [25,49]. Besides, they lack flexibility and scalability to model the hierarchical relationships among activities at multiple levels.
2. Current graphical-based approaches mainly address the models for classification. They often need to construct a separate network structure for each class of activities. It would be difficult for these approaches to build a unified model for answering semantic queries such as "*is an activity occurring*", "*will an activity occur*" and "*had an activity already occurred*".
3. Semantic-based approaches are often rich in modeling temporal and multilevel relationships, but they often lack the expressive power to capture and propagate the uncertainties associated with their temporal dependencies.
4. Formulas or rules in semantic-based approaches are often hand-coded or based on common knowledge. They need to be carefully defined by domain experts. It would be rather difficult to handcraft each formula whose temporal relations among activities are intricate.

To address these issues, we present a framework that can automatically learn event relations and their weights from a training dataset, and use them to answer varied semantic-level queries under uncertainty. Our framework formulates semantic-based representations of event models to describe eight interval-based event relational properties by combining seven Allen's temporal relations [2] and one hierarchical relation. Additionally, we define four predicates that can be used for answering the commonly performed queries in activity recognition applications:

—*High-level activity classification*: If a set of events are observed, what is the possibility of the occurrence of a certain high-level activity?
—*Future Activity Prediction*: If a set of events are observed, what is the possibility that a certain activity will occur?
—*Past Activity Verification*: If a set of events are observed, what is the possibility that a certain activity had occurred?
—*Parallel Activity Discovery*: If a set of events are observed, what is the possibility that a certain activity is co-occurring?

We leverage pattern mining techniques to automatically learn an *event knowledge base* (EKB) of formulas and their weights associated with the aforementioned queries in a unified way. Specifically, frequent and confident formulas are defined to capture and propagate the uncertainties of event relational properties. Based on these mined formulas, our framework draws upon probabilistic semantics of Markov logic networks to infer multiple types of tasks under uncertainty.

The main contributions of the current paper are as follows:

1. A semantic-based probabilistic framework is presented to explicitly learn propositional formulas and their weights for complex activity recognition that, together with other formulas collected from domain or expert knowledge, forms an EKB.
2. Pattern mining techniques are introduced to learn formulas with intricate temporal and hierarchical relations in the presence of uncertainty. It is worth noting that the goal of our framework was not to enhance the pattern mining techniques per se, but rather to employ the existing techniques for formulas learning in the field of complex activity recognition.
3. By representing as and reasoning on Markov logic networks, various semantic-based queries in activity recognition can be inferred based on EKB under uncertainty in a unified way.
4. By leveraging EKB with the learned formulas under uncertainty, our framework is robust against the incomplete or incorrect observations of events.

Our focus in this paper is at the semantic level of probabilistic activity models. In what follows we start by introducing briefly, in Section 2, the background of primitive event recognition from sensor data, complex activity recognition models and formula learning by employing pattern mining techniques. Then, we present the representations of event models in Section 3. Section 4 introduces the details of learning probabilistic event-based formulas under uncertainty using pattern

mining techniques. Section 5 describes a framework for answering semantic-based queries in activity recognition applications. The experimental evaluation and results are presented in Sections 6 and 7. Section 8 summarizes the main contributions and concludes this paper.

2. Background

2.1. Inferring simple events from sensor data

Simple events or *primitive events* are referred to as primitive activities that can be inferred from sensors and cannot be further decomposed under application semantics [40]. The *interval* of a primitive event can also be obtained as the period of time over which the corresponding status remains unchanged. Many approaches have been proposed in the literature to recognize simple events which can be inferred from various sensors. Computer vision-based approaches have been at the forefront of this research field. These approaches have contributed significantly to detect gestures or movements from video sequences using cameras. Recently, the maturity and prevalence of powerful sensors and high-speed processors offer advanced capabilities to build sensor-based human activity recognition systems in a pervasive and ubiquitous environment. In fact, cameras can also be viewed as a specific sensor providing visual data. So far hundreds of approaches have been proposed in the literature to detect simple events from various sensors, including wearable or on-body sensors such as accelerometers, gyroscopes, microphones and physiological sensors, and environmental sensors such as infrared sensors, reed switches and RFID tags. We refer the interested readers to the excellent tutorials and reviews (Aggarwal and Ryoo [1], Chen et al. [9] and Bulling et al. [8]) for recognizing primitive events from sensors.

2.2. From simple events to complex activities

We focus on complex activity recognition in our framework by assuming that primitive events and their corresponding intervals have already been recognized from sensors. Recent works have tried to address the modeling and recognition of complex activities. They can generally be categorized into two types, graphical model-based and semantic-based.

2.2.1. Graphical models

Graphical models such as the hidden Markov model (HMM) family, conditional random field (CRF) family and Bayesian network (BN) family utilize network structures to model complex activities. HMMs are more suitable for purely sequential activities. CRFs have shown limitations in recognizing concurrent activities and dynamic BNs (DBNs) impose more computational burden than other graphical models. Most importantly, the above graphical models can only capture three time point-based relations: precedes, follows, and equals. To solve this issue, many variants have been proposed to handle more complex event relations, such as during and overlaps. It is worth mentioning that the interval temporal Bayesian network (ITBN) [49] can capture thirteen Allen's temporal relations. However, since the Bayesian network structure is a directed acyclic graph, ITBN has to remove some temporal relations from the training dataset in order to make the Bayesian network temporally consistent, resulting in information loss. Also, checking temporal consistency of such triangle relationships and evaluating all possible network structures (i.e. which relation should be ignored or not) are computationally expensive.

Overall, the graphical models are adept at handling uncertainties, but they face the following issues:

- They often suffer from inflexibility of expressing intricate temporal and hierarchical event relations.
- They have to construct a separate network structure for each class of activities. It is in general difficult to build a unified model for varied queries.

2.2.2. Semantic-based models

Semantic-based approaches, on the contrary, are often rich in modeling temporal and hierarchical relationships. Ryoo and Aggarwal [38] used a context-free grammar (CFG) based representation scheme, which is manually encoded, to represent complex events. Although the CFG-based framework can represent rich event relations, it cannot handle uncertainty efficiently. Also, it is difficult to accurately handcraft CFG grammars for a large number of human activities. Brendel et al. [7] formulated a probabilistic event logic (PEL) knowledge base consisting of confidence-weighted formulas from an interval-based event logic. Their PEL knowledge base can handle Allen's relations, but its formulas have to be manually configured in advance. Helaoui et al. [17] used a log-linear model to implement probabilistic ontological reasoning, which can handle uncertainties for multilevel activities. However, it cannot support complex temporal modeling and reasoning, and is difficult to manually encode formulas and their weights efficiently. In another work, Helaoui et al. [16] exploited Markov logic formulations for inferring time point-based events. Like most of the semantic-based approaches, the formulas are hand-coded and need to be carefully defined. Their experimental results show that the appropriate selection of formulas is significant to final recognition results. Saguna et al. [40] used activity signatures and their weights, which are mainly encoded from domain knowledge, to handle variations in sequence and concurrency of complex activities. If temporal relations among activities are intricate, it would be rather difficult to handcraft each formula accurately. Many other semantic-based approaches [22,23,40] can handle rich event relations, but their formulas and weights are manually encoded from the domain knowledge. The knowledge base of formulas in these semantic-based approaches could be viewed as static and incomplete. If event relations are intricate, it would be rather difficult to handcraft each formula and its weight accurately.

2.2.3. Learning event relations in semantic-based models

Recently, several semantic-based approaches have been proposed for learning event relations. Fern et al. [13] introduced the *Leonard* system that learns formulas from training dataset. Yet the system can only support *before* and *equal* temporal relations. Veeraraghavan et al. [43] proposed an approach that learns stochastic context-free grammars, which however can only describe sequential relations between activities. And-Or Graph (AOG) [15] and its variant Temporal And-Or Graph (T-AOG) [35] are used for modeling causal relationship among events. Some Allen relationships such as *during* and *overlaps* cannot be learned. Dubba et al. [12] presented a framework *Remind* for supervised relational learning of activity models containing Allen's relations using Inductive Logic Programming (ILP). The framework can learn the Boolean values of formulas, but not their weights, and thus has limitation in handling uncertainty.

In summary, current semantic-based approaches are rich in modeling event relations, but they face one or more of the following issues:

- Formulas and their weights in previous activity recognition approaches often need to be carefully hand-coded by domain experts. It would be rather difficult to handcraft each formula whose event relations are intricate.
- Current approaches for learning event relations often lack the expressive power to capture uncertainties or to learn formula weights.

2.3. Learning formulas using pattern mining techniques

To address the above issues in formula learning, we introduce pattern mining techniques in our framework. Pattern mining techniques are widely applied for finding relationships among items in a database. In our framework we employ patterns to learn semantic-based formulas involving Allen's interval relations and hierarchical relations. As mining patterns is computationally costly, the existing work on temporal pattern mining during the last decade can be used for reference during our formula-learning process. Moskovitch and Shahar [29] have contributed an excellent review in this area. In what follows we will highlight some approaches that can optimize pattern mining process. Sacchi et al. [39] presented an algorithm to learn temporal association rules involving complex temporal patterns in both their antecedent and consequent. Wu and Chen [46] presented TPrefixSpan to mine non-ambiguous temporal patterns from time interval-based events. Winarko and Roddick [45] proposed an algorithm, called ARMADA, which incorporates a *partition-and-validation* technique to calculate the supports of patterns when the dataset is too large to fit into the memory. By employing different optimization or pruning techniques to reduce the search space, several approaches can discover frequent interval-based patterns effectively, such as H-DFS [33], IEMiner [34] and CTMiner [10]. Chen et al. [10] compared the runtime and memory usage of their CTMiner to TPrefixSpan, IEMiner and H-DFS and found CTMiner to be fastest in computation and least in memory-consumption. Moskovitch and Shahar [30] introduced a family of KarmaLego to discover temporal-interval related patterns efficiently. Their results show that KarmaLego is faster than ARMADA, H-DFS and IEMiner.

After patterns are extracted, they are commonly used as features for prediction and classification. Batal et al. [5] presented a framework to generate a small set of predictive and nonspurious patterns and applied it in the application of predicting diabetic patients as well as patients who are at risk of developing heparin-induced thrombocytopenia. Moskovitch and Shahar [29] presented a framework for classification of multivariate time series analysis in the domains of diabetes, intensive care, infectious hepatitis and others from electronic health records. In the field of complex activity recognition, Gu et al. [14] used emerging patterns to recognize both sequential and concurrent activities under uncertainties. The main purpose of this approach is to discriminate sequentially and concurrently composed primitive activities from sensor data using patterns. It does not handle the temporal and multilevel relations among these activities. As a result, the computation complexity grows exponentially as the number of overlapping activities increases. Also, it is hard to capture long-term temporal dependencies. Ye et al. [47] augmented a range of learning techniques with ontological semantics to facilitate the unsupervised discovery of patterns that each user performs daily living activities. Li and Fu [21] used temporal sequence patterns for the early prediction of ongoing human activities. In one of our previous works [26], we presented an efficient algorithm to identify temporal-interval patterns among primitive events and utilized them as features to represent complex activities for automated recognition. In all aforementioned approaches the learned patterns are used as features to induce a classifier.

Our work is somewhat similar to the previous work. However, our objective differs from theirs in that we provide a unified way to answer temporal and hierarchical associated semantic queries about the probabilities of activity occurrences. This leads us to transform learned patterns into weighted formulas instead of treating them as features for classifiers. To answer various semantic queries in the presence of uncertainty, we intuitively consider Markov logic networks for inference based on those learned formulas. In this way, the learned formulas can be combined with other formulas in existing semantic-based systems to answer queries not limited to what we defined in this work, especially for the applications where temporal-interval-related formulas are intricate and difficult to hand-code. To the best of our knowledge, no unified framework has been proposed to deal with semantic queries associated with both temporal and multilevel relations under uncertainty using patterns learned from training datasets.

3. Event models for activity recognition

We first provide necessary definitions and representations of temporal and multilevel relations over activity events.

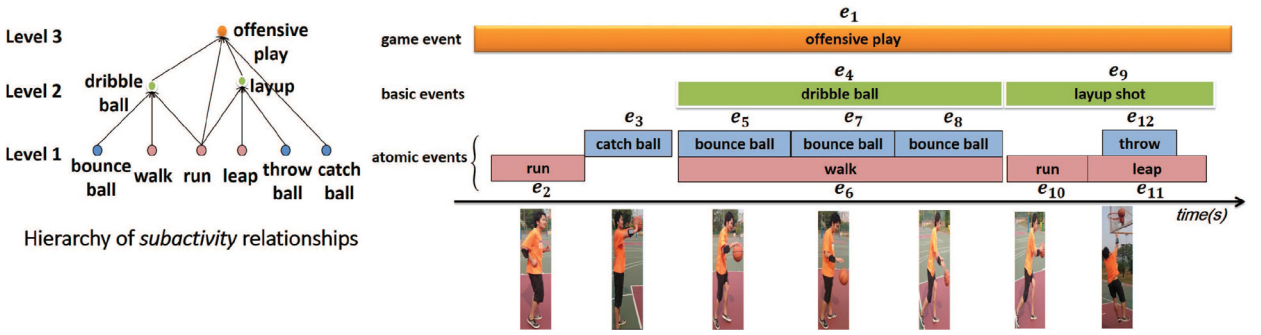


Fig. 1. An example of a record of basketball play scenarios containing 12 events with nine activity types that are hierarchically organized at three levels.

3.1. Definitions of events and relations

Let Σ be a finite set of activity types. Each activity type $\mathtt{X} \in \Sigma$ is associated to a level $\ell(\mathtt{X}) \in \mathbb{N}$. We use $\mathtt{X} \ll \mathtt{Y}$ to represent that $\mathtt{X}$ is a *subactivity* of $\mathtt{Y}$ with $\ell(\mathtt{X}) < \ell(\mathtt{Y})$. A *primitive event* is a period of time $I_i = [t_i^-, t_i^+)$ during which an activity instance a_i occurs ($i \in \mathbb{N}$), denoted by $a_i@I_i$, where $a_i \in \Sigma$, and t_i^- and t_i^+ are the start-time and end-time of a_i, respectively, with $t_i^- < t_i^+$. A *record* of primitive events $\mathcal{E} =< a_1@I_1, a_2@I_2, \ldots, a_K@I_K >$ is a sequence of K intervals sorted by start-time and end-time in ascending order, such that for any $1 \le i < j \le K$, $t_i^- < t_j^-$ or $t_i^- = t_j^-$ and $t_i^+ \le t_j^+$.

We define eight event relations, namely seven temporal relations and one hierarchical relation, as follows.

Definition 1 (Event relations)**.** Given two primitive events $e_i = a_i@I_i$ and $e_j = a_j@I_j$ in a record, we define a set of event relations $\Gamma = \{\mathtt{b}, \mathtt{m}, \mathtt{o}, \mathtt{s}, \mathtt{f}, \mathtt{c}, \mathtt{e}, \mathtt{i}\}$ such that

- (before) e_i $\mathtt{b}$ e_j: $t_i^+ < t_j^-$;
- (meets) e_i $\mathtt{m}$ e_j: $t_i^+ = t_j^-$;
- (overlaps) e_i $\mathtt{o}$ e_j: $t_i^- < t_j^- < t_i^+ < t_j^+$;
- (starts) e_i $\mathtt{s}$ e_j: $t_i^- = t_j^- < t_i^+ < t_j^+$ and $a_i \not\ll a_j$;
- (finished-by) e_i $\mathtt{f}$ e_j: $t_i^- < t_j^- < t_i^+ = t_j^+$ and $a_j \not\ll a_i$;
- (contains) e_i $\mathtt{c}$ e_j: $t_i^- < t_j^- < t_j^+ < t_i^+$ and $a_j \not\ll a_i$;
- (equals) e_i $\mathtt{e}$ e_j: $t_i^- = t_j^- < t_i^+ = t_j^+$, $a_j \not\ll a_i$ and $a_i \not\ll a_j$;
- (involves) e_i $\mathtt{i}$ e_j: $t_i^- <= t_j^- < t_j^+ <= t_i^+$ and $a_j \ll a_i$.

Note that the Allen's temporal relations [2] are redefined by appending satisfiable conditions on *subactivity* relationships between activities. We illustrate these relations with a basketball play scenario, as shown in Fig. 1. The hierarchy of activities is given based on the common knowledge of basketball play. For example, the player first runs to a position and catches the ball. The relation between the two intervals is *meets*, i.e. $e_2 \mathtt{m} e_3$. Since bounce ball and walk are the elements of dribble ball, we can see that $e_4 \mathtt{i} e_5$, $e_4 \mathtt{i} e_6$, $e_4 \mathtt{i} e_7$ and $e_4 \mathtt{i} e_8$. Notice that the relation between the intervals e_4 and e_7 cannot be *contains* according to our definition. Although run is an element of layup shot, e_9 does not involve e_2 according to the definition of multilevel relations. The relation between these two intervals is *before*, i.e. $e_2 \mathtt{b} e_9$.

3.2. Semantic-based representations of events

In the above example, all the relations of primitive events can be explained in a semantic language. We can readily represent the event relations by a set of predicates, denoted by $\Theta = \{\mathtt{before}, \mathtt{meet}, \mathtt{overlap}, \mathtt{start}, \mathtt{finished\text{-}by}, \mathtt{contain}, \mathtt{equal}, \mathtt{involve}, \mathtt{occur}\}$. For any predicate with arguments $\alpha(\mathtt{X}, \mathtt{Y})$ where $\alpha \in \Theta$ and $\mathtt{X}, \mathtt{Y} \in \Sigma$, an *interpretation* over it is the occurrence of primitive events e_i and e_j such that $e_i \mathbf{r} e_j$ where $\mathbf{r} \in \Gamma$ is the corresponding event relation of α, and $a_i = \mathtt{X}$ and $a_j = \mathtt{Y}$. For instance, the semantic-based representation of $e_i \mathtt{b} e_j$ is $\mathtt{before}(\mathtt{X}, \mathtt{Y})$. It is worth noting that $\mathtt{occur}(\mathtt{X})$ is a unitary predicate which represents the occurrence of activity $\mathtt{X}$. The satisfaction of any other binary predicates $\alpha(\mathtt{X}, \mathtt{Y})$ implies the satisfactions of $\mathtt{occur}(\mathtt{X})$ and $\mathtt{occur}(\mathtt{Y})$.

A *complex event* can be represented as a collection of these predicates concatenated by logical predicates including $\mathtt{and}(\wedge)$, $\mathtt{or}(\vee)$ and $\mathtt{not}(\neg)$. For example, "a player runs to a position and catches the ball, and then dribble the ball" can be translated into $\mathtt{meet(run,catch)} \wedge \mathtt{before(catch,dribble)}$.

A semantic query can also be represented by our defined predicates. For example, the query "if *run meets catch ball* and *dribble ball before catch ball* are observed, does *dribble ball meet layup*?" is equivalent to a semantic-based representation of $\mathtt{meet(run,catch)} \wedge \mathtt{before(catch,dribble)} \rightarrow \mathtt{meet(dribble,layup)}$. (Note that we will commonly use the shorthand $\phi \rightarrow \phi'$ for $\neg\phi \vee \phi'$). Besides such simple queries about event occurrence or event relations, we also consider complex semantic queries which are commonly asked questions. For instance, "do the observation of the event *run meets catch ball*

and *dribble ball before catch ball* imply the occurrence of the event *an offensive play*?"; or "if *run after dribble ball* is observed, will the player *layup*?". We define four *query predicates* associated with these commonly asked queries.

Definition 2 (Query predicates)**.** Given a set of observed primitive events $E = \{e_1, e_2, \ldots, e_n\}$ ($e_i = a_i@I_i$ for any $i \in [1, n]$) and another primitive event e_j ($e_j = a_j@I_j \notin E$). We define a set of query predicates $\Psi = \{\texttt{highlevel}, \texttt{predict}, \texttt{verify}, \texttt{discover}\}$ such that

- $\texttt{highlevel}(a_j; a_1 \ldots a_n)$: $\forall e_i \in E : e_j \texttt{i} e_i$;
- $\texttt{predict}(a_j; a_1 \ldots a_n)$: $\forall e_i \in E : e_i \texttt{b} e_j$ or $e_i \texttt{m} e_j$;
- $\texttt{verify}(a_j; a_1 \ldots a_n)$: $\forall e_i \in E : e_j \texttt{b} e_i$ or $e_j \texttt{m} e_i$;
- $\texttt{discover}(a_j; a_1 \ldots a_n)$: $\forall e_i \in E : e_j \texttt{s} e_i$ or $e_i \texttt{f} e_j$ or $e_i \texttt{c} e_j$ or $e_i \texttt{e} e_j$.

These query predicates can be interpreted as follows: if a set of events E are observed, then "is higher-level activity a_j occurring?" (for `highlevel`); "will activity a_j occur?" (for `predict`); "had activity a_j occurred?" (for `verify`); "is another activity a_j co-occurring?" (for `discover`). According to the definition, the aforementioned query about the occurrence of an ongoing activity can be represented as $\mathrm{r} = \texttt{before(dribble,run)} \rightarrow \texttt{predict(layup;dribble,run)}$. The predicate `predict(layup;dribble,run)` holds iff any corresponding event of the activity (i.e. run or dribble ball) either is before or meets the corresponding event of layup. In Fig. 1, `predict(layup;dribble,run)` is true on account of that $e_2 \texttt{b} e_9$ and $e_4 \texttt{b} e_9$. Furthermore, the query r holds because `before(dribble,run)` is also true on account of that $e_2 \texttt{b} e_4$. Notice that other query predicates can also be defined in this way to describe requisite questions by specifying events and their relations.

Formally, we recursively define semantic-based representations of events and their satisfiable conditions as follows.

Definition 3 (Semantic-based representation)**.** A *semantic-based representation* δ *of an event* e is either a predicate $\phi \in \Theta \cup \Psi$ (called a primitive representation), or one of the compound expressions: $\phi \wedge \phi'$, $\phi \vee \phi'$ and $\neg\phi$, where ϕ and ϕ' are semantic-based representations.

For a primitive representation δ, δ is satisfied if e is observed; $\phi \wedge \phi'$ is satisfied if both ϕ and ϕ' are satisfied; $\phi \vee \phi'$ is satisfied if ϕ or ϕ' is satisfied; $\neg\phi$ is satisfied if ϕ is not satisfied.

Note that a query can be viewed as an event and its *semantic-based representation* is of the form $\phi \rightarrow \phi'$ (i.e. $\neg\phi \vee \phi'$).

Because of the uncertain characteristic of human activities, an event or a query should be associated with a weight rather than a *hard* result that is either true or false. That is, what is the probability that an event or a query holds? An *event knowledge base* (*EKB*) is defined to represent such *soft* events as follows.

Definition 4 (Event knowledge base)**.** An *event knowledge base* $\mathcal{L}$ is a set of N weighted events, as denoted by

$$\mathcal{L} = \{(\phi_i, \omega_i) : \text{ for } i \in [1, N]\},$$

where ϕ_i is the semantic-based representation of an event e_i, which is also called a *formula*, and ω_i is a non-negative numeric weight associated with ϕ_i.

One common strategy for building an EKB is to manually define the semantic-based representation of each event and its associated weight based on common knowledge. However, it is not desirable when a large number of activity types or intricate event relations are involved in EKB. We intend to learn EKB from a given training dataset consisting of records. It is worth noting that the learned formulas can be combined together with other formulas encoded by domain experts or based on common knowledge to form a generalized EKB for inference.

4. Learning an event knowledge base

We employ pattern mining techniques to learn formulas and their weights in an EKB from a training set of records.

4.1. Event pattern

We first give the definition of event pattern as follows.

Definition 5 (Event pattern)**.** Given a record $\mathcal{E}$ of K events $< e_1, \ldots, e_K >$ ($e_i = a_i@I_i$ and $a_i \in \Sigma$, for any $i \in [1, K]$), an *event pattern* (or *pattern* for short) of $\mathcal{E}$ is defined by a $K \times K$ upper triangular matrix $\mathbf{M}$ whose element $\mathbf{M}[i, j] \in \Gamma$ denotes the event relation between e_i and e_j, for any $1 \le i < j \le K$.

A pattern of size $K \times K$ is called a *K-pattern*. Given two patterns p and p', p' is a *subpattern* of p if there is a 1-1 mapping from events in p' to a subset of events in p that preserves relations between any pair of events, denoted by $p' \sqsubseteq p$. If additionally p' is a $(K-1)$-pattern and p is a K-pattern, we denote it by $p' \sqsubseteq_1 p$. As shown in Table 1, we can see that all the 2-patterns and 3-patterns are subpatterns of $p_{8,1}$. Meanwhile, $p_{2,1} \sqsubseteq_1 p_{3,1}$, and $p_{2,3} \sqsubseteq_1 p_{3,2}$. Note that any matrix is inversely symmetric over relations and hence we ignore its lower triangle that contains inverse relations.

We can readily convert a pattern p to its corresponding semantic-based representation, written by $\mathcal{H}(p)$. For instance, $p_{3,1}$ is translated to $\mathcal{H}(p_{3,1}) = \texttt{involve}(a_1, a_2) \wedge \texttt{involve}(a_1, a_3) \wedge \texttt{meet}(a_2, a_3)$. As for 1-patterns, we use the predicate `occur` to represent the occurrence of a single primitive event.

Table 1
An example of event patterns of different sizes.

2-pattern	$p_{2,1}=\begin{matrix}&1&2\\1&-&i\\2&-&-\end{matrix}$, $p_{2,2}=\begin{matrix}&2&3\\2&-&m\\3&-&-\end{matrix}$, $p_{2,3}=\begin{matrix}&6&8\\6&-&f\\8&-&-\end{matrix}$
3-pattern	$p_{3,1}=\begin{matrix}&1&2&3\\1&-&i&i\\2&-&-&m\\3&-&-&-\end{matrix}$, $p_{3,2}=\begin{matrix}&5&6&8\\5&-&s&b\\6&-&-&f\\8&-&-&-\end{matrix}$
8-pattern	$p_{8,1}=\begin{matrix}&1&2&3&4&5&6&7&8\\1&-&i&i&i&i&i&i&i\\2&-&-&m&b&b&b&b&b\\3&-&-&-&b&b&b&b&b\\4&-&-&-&-&v&v&v&v\\5&-&-&-&-&-&s&m&b\\6&-&-&-&-&-&-&c&f\\7&-&-&-&-&-&-&-&m\\8&-&-&-&-&-&-&-&-\end{matrix}$

In this way, we can use patterns to generate formulas. Given a record, if there exist a $(K-1)$-pattern p' and a K-pattern p $(K>1)$ such that $p' \sqsubseteq_1 p$, the appearance of p' implies the appearance of p in the record. Let e_k be an event contained in p but not in p'. We can generate $K-1$ formulas that for all $1 \le i \ne k \le K$, $\mathcal{H}(p') \to \mathcal{H}(p^*)$, where p^* is a 2-pattern that contains e_i and e_k. It can be understood as that the observation of p' implies the occurrence of p^* in the record. In the above example where $p_{2,2} \sqsubseteq_1 p_{3,1}$, we can obtain two formulas, i.e. $\texttt{meet}(a_2, a_3) \to \texttt{involve}(a_1, a_2)$ and $\texttt{meet}(a_2, a_3) \to \texttt{involve}(a_1, a_3)$. Furthermore, to generate formulas with query predicates, we will continue checking whether every obtained formula is satisfiable to the query predicate. If all are true, a formula with query predicate is generated by merging all of them. For example, a new formula $\texttt{meet}(a_2, a_3) \to \texttt{highlevel}(a_1; a_2, a_3)$ is built by merging the above two formulas.

Reversely, given a formula ϕ, we can also find a set of all possible patterns, denoted by $\mathcal{J}(\phi)$. Suppose that there are K events associated with ϕ. Every pattern $p \in \mathcal{J}(\phi)$ is a K-pattern such that (1) for each binary predicate $\alpha(a_i, a_j) \in \Theta$ in ϕ, $\mathbf{M}[i, j]$ is its corresponding event relation; (2) for each query predicate $\beta(a_j; a_1, \ldots, a_K) \in \Psi$ in ϕ, $\mathbf{M}[i, j]$ $(i<j)$ or $\mathbf{M}[j, i]$ $(j<i)$ is the corresponding relation satisfying the query predicate for all $1 \le i \ne j \le K$. Taking an example of a formula

$$\phi = \texttt{meet}(a_2, a_3) \to \texttt{verify}(a_1; a_2, a_3),\ \mathcal{J}(\phi) = \left\{ \begin{matrix}&1&2&3\\1&-&b&b\\2&-&-&m\\3&-&-&-\end{matrix}, \begin{matrix}&1&2&3\\1&-&b&m\\2&-&-&m\\3&-&-&-\end{matrix}, \begin{matrix}&1&2&3\\1&-&m&b\\2&-&-&m\\3&-&-&-\end{matrix}, \begin{matrix}&1&2&3\\1&-&m&m\\2&-&-&m\\3&-&-&-\end{matrix} \right\}.$$

4.2. Frequent and confident formulas

Normally, only the formulas that frequently appear in a training dataset is considered to be contained in an EKB. We give the definition of *frequent formula* as follows.

Definition 6 (Frequent formula)**.** Given a training dataset $\mathcal{D}$ of records, the *support* of a formula ϕ is defined as $supp(\phi) = \sum_{p \in \mathcal{J}(\phi)} \sigma(p)$, where $\sigma(p)$ is the *support* of pattern p. Given a minimum support *minsup*, a formula ϕ is called a *K-frequent formula* if p is a K-pattern and $supp(\phi) \ge minsup$.

The *support* $\sigma(p)$ of a pattern p is often defined as the number of records in which the pattern p appears. That is to say, if a pattern appears in a record, only one occurrence is counted, no matter how many times the pattern appears in the record. In activity recognition, activities may periodically occur. Only counting the occurrence of a pattern at most once in a record may cause the loss of such information. However, counting all possible occurrence of a pattern is computationally demanding, especially when the length of a record is long. Besides, only counting the number of times a pattern appears may lose the information of its duration.

In order to retain these information, we use the definition proposed by Höppner [18], that is, the *duration time* of a pattern p appearing in a record $\mathcal{E}$, denoted by $\mathcal{T}_{\mathcal{E}}(p)$, is the total time in which the pattern p can be observed within a sliding window. Fig. 2 shows an example of $e_i \circ e_j$. We can observe an instance of pattern $p = \begin{matrix}&i&j\\i&-&o\\j&-&-\end{matrix}$ as soon as event e_j moves in the window, and loose it when e_i leaves the window. The observation duration of pattern p is actually the length of the pattern plus the length of sliding window. According to Höppner's proof, we have the fact that $\forall$patterns $p, p' : p' \sqsubseteq p \to \mathcal{T}_{\mathcal{E}}(p') \ge \mathcal{T}_{\mathcal{E}}(p)$.

In our work, the *support* of a pattern p in a record $\mathcal{E}$ is defined as the time duration proportion of pattern p being observed in $\mathcal{E}$, i.e. $\sigma_{\mathcal{E}}(p) = \frac{\mathcal{T}_{\mathcal{E}}(p)}{L_{\mathcal{E}}}$, where the normalization constant $L_{\mathcal{E}}$ is the length of the record plus the length of sliding window. We also define $\sigma(p) = \frac{1}{|D|} \sum_{\mathcal{E} \in D} \sigma_{\mathcal{E}}(p)$. The size of sliding window may limit the extension of a pattern, because

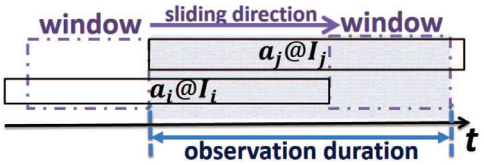


Fig. 2. An illustration of the observation duration of a pattern.

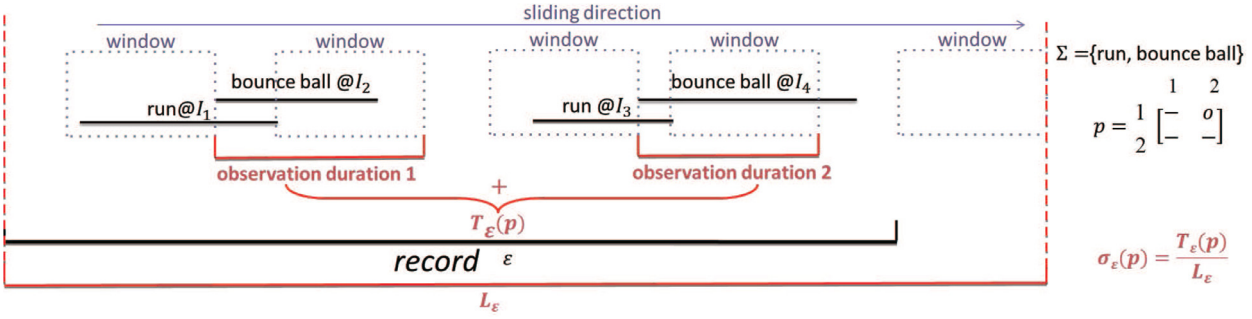


Fig. 3. An example of the occurrence of the pattern *run overlaps bounce ball* in a record.

the sliding window may not be large enough to observe the whole pattern. We refer the interested readers to the topic of sliding window selection, and duration calculation as well [19]. We consider only those patterns that can be observed within a chosen sliding window. Fig. 3 shows an example of computing the support of the pattern *run overlaps bounce ball* given a record $\mathcal{E}$. The support $\sigma_\mathcal{E}(p)$ is the proportion of the total duration that p is observed in $\mathcal{E}$. Given a set of records, we can compute out the total support $\sigma(p)$. Finally, to estimate the support of a given formula ϕ, we first find its corresponding set of all possible patterns $\mathcal{J}(\phi)$, then compute the sum of the supports of each pattern in $\mathcal{J}(\phi)$, i.e. $supp(\phi) = \sum_{p\in\mathcal{J}(\phi)}\sigma(p)$.

Now we define the *weight* of a formula over patterns as follows.

Definition 7 (Weight)**.** Given a formula ϕ which is generated by a $(K-1)$-pattern p' and a K-pattern p $(K>1)$ such that $p' \sqsubseteq_1 p$, the *weight* of ϕ is defined as $\omega_\phi = \frac{supp(\phi)}{supp(\mathcal{H}(p'))}$.

The weight of a formula indicates the possibility of the occurrence of a certain event (i.e. the event that is in p' but not in p) if a set of events and their relations (i.e. pattern p') are observed. Besides frequent formula, it is also important to choose a *confident formula*.

Definition 8 (Confident formula)**.** Given a minimum confidence *minconf*, a formula ϕ is called a *confident formula* if $\omega_\phi \geq minconf$.

Only the formulas that are both frequent and confident will be contained in an EKB.

4.3. Learning formulas and their weights over patterns

An EKB learning algorithm is provided over a training dataset of records, as shown in Fig. 1. Let $\mathcal{FS}(k)$ be a set of all possible k-patterns extracted from the training dataset. A pattern in $\mathcal{FS}(k)$ $(k\geq 2)$ is selected if its corresponding formulas are frequent and confident. Any formulas whose support and confidence reach their respective minimum thresholds are added into the EKB. The algorithm stops until no more k-patterns are found.

If the training dataset is large or the sequence in the dataset is long, calculating the supports of all possible formulas would be a challenge work. A few technologies are designed to optimize the pattern support calculation [27,28,45]. Unfortunately, these technologies cannot be utilized directly in our algorithm, because their optimization strategies are due to the definition that the occurrence of a pattern is the number of records the pattern appears in the dataset. Similar to these optimization strategies for finding frequents, like Apriori, we optimally generate $(k+1)$-frequent formulas from the set of k-frequent formulas. Those infrequent formulas will not be considered in the next iteration of formula generation. We give the following property of frequent formulas.

Theorem 1. *Let p' be a k-pattern. For any primitive representations φ and φ', if a formula $\phi' = \mathcal{H}(p') \to \varphi'$ is not a frequent formula, then for any $(k+1)$-pattern $p \in \mathcal{J}(\phi)$, $\phi = \mathcal{H}(p) \to \varphi$ is not a frequent formula.*

Proof. (By contradiction) First, it can be seen that $\sqsubseteq$ is a partial order relation on patterns. Suppose ϕ is a frequent formula, then $\sum_{q\in\mathcal{J}(\phi)}\sigma(q) = \sum_{q\in\mathcal{J}(\phi)}\sum_{\mathcal{E}\in D}\sigma_\mathcal{E}(q) = \sum_{q\in\mathcal{J}(\phi)}\sum_{\mathcal{E}\in D}\frac{T_\mathcal{E}(q)}{L_\mathcal{E}} \geq |D| \times minsup$. For any $q \in \mathcal{J}(\phi)$ and $q' \in \mathcal{J}(\phi')$, we have $p' \sqsubseteq q'$ and $p \sqsubseteq q$. Since $p' \sqsubseteq_1 p$, we have that (1) if pattern q appears, pattern q' certainly appears, i.e. $|\mathcal{J}(\phi)| \leq |\mathcal{J}(\phi')|$; (2)

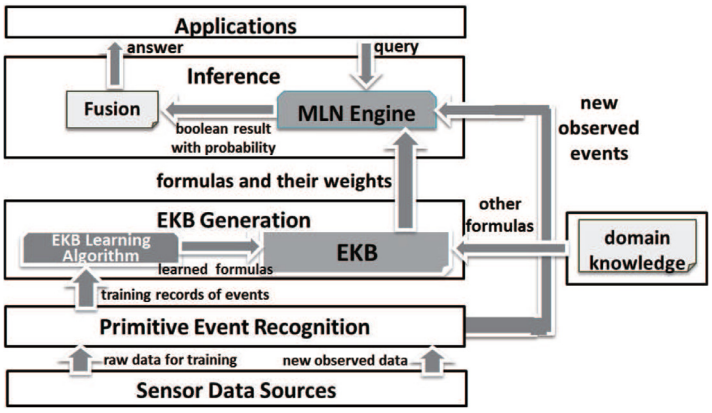


Fig. 4. Our proposed Activity Recognition Framework (ARF). In this work we focus on the components marked in dark colored boxes.

pattern q contains q', i.e. $q' \sqsubseteq q$. Since $q' \sqsubseteq q$, $\mathcal{T}_{\mathcal{E}}(q') \geq \mathcal{T}_{\mathcal{E}}(q)$. Then, we have $\sum_{q' \in \mathcal{J}(\phi')} \sigma(q') \geq |D| \times minsup$, which contradicts to that ϕ' is not a frequent formula. □

Based on this property, if a pattern $p \in \mathcal{J}(\phi)$ is selected for generating formulas at the next iteration, the formula $\phi = \mathcal{H}(p') \rightarrow \varphi$ must be a frequent formula. Instead of searching all $(k+1)$-patterns, the algorithm only needs to update $\mathcal{FS}(k+1)$ by selecting patterns in $\mathcal{FS}(k)$ whose associated formulas are frequent (line 12). We can see that the formulas discovered by our algorithm are lossless.

5. Framework for answering semantic query

Our approach can be deployed in various application domains that require complex activity recognition associated with event relations. Moreover, we envision that the formulas learned by our algorithm can be combined with other formulas encoded from existing domain knowledge in an EKB, especially in the situation where applications require temporal related queries but the temporal formulas are intricate or difficult to hand-code. Our proposed framework, named **A**ctivity **R**ecognition **F**ramework (ARF for short), consists of five components, as shown in Fig. 4.

5.1. Sensor data sources

The sensor data sources include wearable sensors that are positioned directly or indirectly on human body (e.g. accelerometer, GPS and biosensor), and ambient sensors that are attached to objects to monitor user-object interactions (e.g. RFID, infrared sensors, reed and pressure mat). Cameras and other virtual sensors can also be regarded as sensor data sources. For example, browsing and phone activities can be retrieved from virtual sensors on the user's devices (e.g. a logging app).

5.2. Primitive event recognition

The primitive event recognition component can handle different types of sensor data, which are required to infer events of the occurrences of *primitive activities* and their *durations*. Primitive activities are defined as unit-level activities that can be inferred by sensors and cannot be broken down further under application semantics [40]. The durations of these primitive activities can also be detected as the period of time over which the corresponding sensor states remain unchanged. There are mature techniques to recognize primitive events from sensors. For instance, sensor-based body-motion and gesture related primitive activities, such as walking, sitting and standing, are inferred using a general-purpose recognition system containing signal processing and machine learning techniques developed by Bulling et al. [8]. Aggarwal and Ryoo [1] reviewed vision-based recognition methodologies of the primitive activities of a single person and person-object interactions. Specifically in our sensor-based experiment presented in Section 6.2, we utilized the tool developed by Bulling et al. [8], which implements several classifiers including kNN, DT, C4.5 and SVM, to detect primitive events from sensors. In our vision-based experiment, we adopted the approach of primitive event recognition proposed by Zhang et al. [49], where kinematic and image based features are extracted from the bounding box of the targets in the videos and multiple DBNs are used to model each primitive activity. Note that it is common that a missing or false recognition of a primitive event often occurs due to sensor noises or low-level prediction errors.

5.3. EKB generation

The EKB generation component builds an EKB using training records of primitive events inferred from the primitive event recognition component. The pattern mining technique enables our approach to discover the weighted formulas involving both temporal-interval-related and hierarchical relations. It is worth mentioning that although we employ pattern mining techniques in Algorithm 1, frequent and confident formulas but not event patterns are discovered as out-

Algorithm 1 The EKB learning algorithm.

Input:
D - the training dataset of records
Σ - the set of activity types
$minsup$, $minconf$ - the predefined minimum support and minimum confidence
Output:
$\mathcal{L} = \{(\phi, \omega_\phi)\}$
1: $\mathcal{L} = \phi$, $k = 1$;
2: $\mathcal{FS}(1) = \{p : p \text{ is 1-pattern}\}$;
3: **while** $\mathcal{FS}(k) \neq \phi$ **do**
4: **for all** φ that is a primitive representation and $p' \in \mathcal{FS}(k)$ **do**
5: $\phi = \mathcal{H}(p') \rightarrow \varphi$;
6: **if** any $p \in \mathcal{J}(\phi)$ has $p' \sqsubseteq_1 p$ **then**
7: Calculate $supp(\phi)$ and ω_ϕ;
8: **if** $supp(\phi) \geq minsup \wedge \omega_\phi \geq minconf$ **then**
9: $\mathcal{L} = \mathcal{L} \bigcup \{(\phi, \omega_\phi)\}$;
10: **end if**
11: **if** $supp(\phi) \geq minsup$ **then**
12: $\mathcal{FS}(k+1) = \mathcal{FS}(k+1) \bigcup \{p : p \in \mathcal{J}(\phi)\}$;
13: **end if**
14: **end if**
15: **end for**
16: $k = k + 1$;
17: **end while**
18: **return** $\mathcal{L}$;

puts. This is because our objective is to provide an EKB that can be used for inferring various semantic-based queries. Event patterns can be viewed as intermediate variables helping find targeting formulas in the algorithm. Since we have specific definitions on frequent formulas, many existing approaches of discovering temporal-interval-related patterns cannot be used directly in our algorithm. We find our own way to accelerate the formula discovery process as depicted in Theorem 1. After formulas associated with temporal-interval-related and hierarchical relations are learned, other formulas encoded based on domain knowledge or from other existing semantic-based systems can also be contained in the EKB. For example, our algorithm discovers a formula `before(dribble,run)` $\rightarrow$ `predict(layup;dribble,run)` with the weight of 0.7. According to the domain knowledge, eighty percent of players use their right hand for layup, which can be formalized as `use_right_hand(dribble)` $\wedge$ `use_right_hand(run)` $\rightarrow$ `predict(layup;dribble,run)` with the weight of 0.8. The predicate *use_right_hand* indicates whether a player is doing an action using his/her right hand. By integrating these formulas and employing a Markov logic network as elaborated in the following section, we can answer our previously defined semantic-based queries in the presence of uncertainty, including (but not limited to) `highlevel`, `predict`, `verify` and `discover`.

5.4. Probabilistic inference

The inference component exploits Markov logic network (MLN for short) to answer arbitrary queries from various applications under uncertainty. Given an EKB $\mathcal{L}$, the procedure for estimating the probability that a new query holds is identical to the inference on a MLN. Although several other approaches, such as Bayesian inference and Markov chain monte carlo methods, are provided for probabilistic inference, we leverage MLNs for inference by three reasons: MLNs can robustly handle uncertainty; Their representations are more flexible to the number of activity types and predicates compared to other probabilistic models, such as HMMs or DBNs; MLN can estimate the exact probability of an event, while other approaches such as PEL inference [7] can only answer a query like which activity is the most probable one through a defined score.

Let $\mathcal{MLN}_\mathcal{L}$ be a MLN together with a finite set of constants Σ, which refers to the set of all possible activity types. Each event representation $\phi \in \mathcal{L}$ can be converted to a formula in conjunctive normal form [36], written as $\mathcal{F}_\phi$. Each formula $\mathcal{F}_\phi$ is also called a ground formula in MLN, and each predicate in a ground formula is called a ground atom. $\mathcal{MLN}_\mathcal{L}$ contains one node for each ground atom and one edge between two ground atoms appearing in the same ground formula. A set of ground atoms is called a possible world. A possible world is a true grounding of a formula $\mathcal{F}_\phi$ if $\mathcal{F}_\phi$ is true in the possible

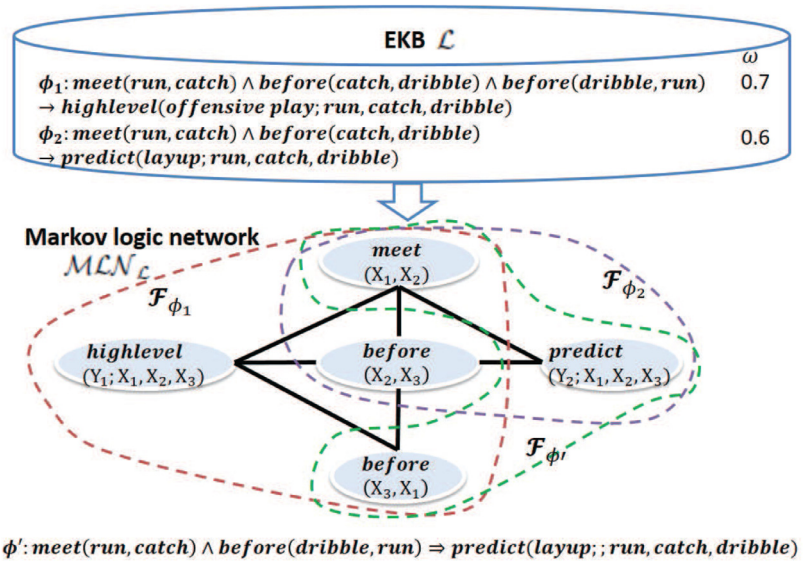


Fig. 5. An example of EKB and its corresponding MLN where $\mathtt{X}_1 = \mathtt{run}$, $\mathtt{X}_2 = \mathtt{catch}$, $\mathtt{X}_2 = \mathtt{catch}$, $\mathtt{X}_3 = \mathtt{dribble}$, $\mathtt{Y}_1 = \mathtt{offensive\ play}$, $\mathtt{Y}_2 = \mathtt{layup}$.

world. The probability distribution over a possible world x specified by $\mathcal{MLN}_{\mathcal{L}}$ is given by

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{(\phi, \omega_\phi) \in \mathcal{L}} \omega_\phi f(\phi, x) \right)$$

where $f(\phi, x)$ is a $0-1$ function that

$$f(\phi, x) = \begin{cases} 1 & \text{if } x \text{ is a true grounding of } \mathcal{F}_\phi \\ 0 & \text{otherwise} \end{cases}$$

and Z is a normalization value, i.e. $Z = \sum_{x \in \mathcal{X}} \exp(\sum_{(\phi, \omega_\phi) \in \mathcal{L}} \omega_\phi f(\phi, x))$, where $\mathcal{X}$ is the set of all possible worlds. Then, the probability that a new event or a query ϕ' holds is estimated as follows:

$$P(\phi') = \sum_{x \in \mathcal{X}_{\mathcal{F}_{\phi'}}} P(X = x)$$

where $\mathcal{X}_{\mathcal{F}_{\phi'}}$ is the set of all possible worlds where $\mathcal{F}_{\phi'}$ holds. Fig. 5 illustrates the possible world of a new query ϕ' inferred on the corresponding MLN of a given EKB.

Afterwards, the fusion component is used for integrating the results inferred by MLN in order to answer more complicate queries. For example, one of the commonly asked query is about the classification of high-level activities, i.e. if an event ϕ happens, what high-level activity (e.g. at level l) is occurring? Such queries can be answered by fusing the results of the queries about `highlevel` on all possible activities at level l. The most probable activity is considered as the classification result, i.e.

$$\hat{\mathtt{Y}} = \arg\max_{\mathtt{Y}_i : \ell(\mathtt{Y}_i) = l} P(\phi \rightarrow \mathtt{highlevel}(\mathtt{Y}_i; \mathtt{X}_1, \ldots, \mathtt{X}_n)).$$

6. Evaluation

The objective of the evaluation of the ARF framework, whose goal is to use the discovered EKB to answer various semantic-based queries in the presence of uncertainty, is to estimate the effectiveness of using the mined formulas to answer our defined semantic-based queries in the field of complex activity recognition. Although we focus on high-level complex activity recognition by assuming that primitive events have already been recognized from sensors, we carried out experiments not only in the ideal situation where the all primitive events are correctly detected, but also in the real scenarios of different situations where primitive events are incorrectly recognized or partially observed. Our research questions are related to the performance of the ARF framework compared with other existing methods in complex activity recognition (see more details about the competing methods in the descriptions of experiment design in Section 6.2):

- Can the ARF framework provide a unified model to answer the commonly performed queries in the presence of uncertainty in activity recognition applications, e.g. "If a set of events are observed, what is the possibility that a certain activity will occur"?

Table 2
Summary of the publicly-available datasets.

	Opportunity	OSUPEL
Application type	Daily living	Basketball play
Recording devices	72 on-body sensors	One ordinary camera
E.g. of primitive activities	Sit, open door, etc.	Shoot, jump, dribble, etc.
E.g. of complex activities	Relax, cleanup, etc.	Two offensive play types
No. of types of primitive activities	211	6
No. of complex activities	5	2
No. of records	125	72
No. of events per record	1–78	2–5

- Can the learned formulas be combined with the formulas obtained from domain knowledge in the ARF framework to infer semantic-based queries efficiently?
- Is the ARF framework robust enough to the incomplete or incorrect observations of primitive events originated from sensor-level noise or low-level predication?

6.1. Datasets

Two publicly-available complex activity recognition datasets are considered. As shown in Table 2, both datasets contain unique properties: The Opportunity dataset [37] involves a number of records with intricate relations, while the OSUPEL dataset [7] is comprised of a small number of primitive activities with simple relations.

6.1.1. Opportunity daily living activity dataset

Opportunity dataset[1] contains human activities recorded in a room with kitchen, deckchair, and outdoor deployed 72 sensors of 10 modalities in objects and on the body. Four subjects performed daily morning activities for six "daily living activities" runs (ADL1-6). In each run of ADL1-5, a subject executes activities with freedom about the record of individual primitive activities. Subjects execute a scripted sequence of activities in a "drill" run (ADL6). All activities are organized in a hierarchy of three levels. The annotations describe 211 primitive activities, 17 mid-level activities and five high-level activities. The primitive activities (i.e. primitive events) can be divided into locomotions (e.g. `walk`, `lie`), and the actions of the left and right hands (e.g. `reach`, `release`) and their targeted objects (e.g. `table`, `door`). The mid-level activities are gestures generated from the primitive hand actions (e.g. `open door`). The five high-level activities related to the ADL runs are `relax`, `coffee time`, `early morning`, `cleanup` and `sandwich time`. The complete list of these activities are depicted in Table 3.

The dataset records a total length of 268 h daily living activities, with about an average of 11 h recordings for each subject in each run. The dataset contains a total number of 28,976,744 sensor data records (sampling at 30 Hz). Each record has five tracks of annotations, i.e. locomotion, left-hand action and object, right-hand action and object, mid-level gesture, and high-level activity. About 73% of the records do not belong to any activities. We label these records as *null* activity in our experiments.

6.1.2. OSUPEL basketball play dataset

The OSUPEL dataset[2] is a publicly available video-recorded dataset of actual two-on-two basketball games, consisting of multiple players playing against each other. The players are tracked and labelled with six primitive activities: pass, catch, hold ball, shoot, jump, and dribble. It is suitable for evaluating complex activities such as different offensive play types characterized by rich spatiotemporal constraints. In our experiment, we intend to use these six basic types of complex basketball play activities to recognize two higher offensive play activities, as defined in [49]. That is, PT-I: player 1 receives the ball from throw-in and passes to player 2, and player 2 attacks the rim; PT-II: player 1 receives the ball from throw-in and attacks the rim directly. The numbers of the samples for the two offensive play types are 56 and 16, respectively. We adopt the approach for primitive event recognition proposed by Zhang et al. [49]. The recognition accuracies of these six basic activities are 83% (pass), 76% (catch), 81% (hold), 64% (shoot), 58% (jump) and 58% (dribble), respectively.

6.2. Experiment design

We compare our ARF approach with several graphical model approaches for complex activity recognition, including HMM [20], dynamical skip-chain CRF (SCCRF) [42], DBN [11] and ITBN [49]. We also compare our ARF approach, which learns intricate event-based formulas from training datasets, with the approach proposed by Helaoui et al. [16] (named CK), which

[1] http://www.opportunity-project.eu/challengeDataset.
[2] http://blogs.oregonstate.edu/osupel/dataset.

Table 3
Event levels and names in opportunity dataset.

Event level		Activity name
primitive(211)	Locomotion(4)	stand, walk, sit, lie
	left-hand action and object(95)	unlock bottle, open dishwasher, move sugar, unlock milk, open milk, move milk, unlock door1, open drawer3, move spoon, lock bottle, open drawer2, move knife cheese, lock door1, open cheese, move glass, close bottle, open door1, move cheese, close salami, open door2, move chair, close bread, open drawer1, move plate, close dishwasher, open fridge, move cup, close milk, sip glass, move knife salami, close drawer3, sip cup, move lazychair, close drawer2, clean table, close cheese, bite salami, close door1, bite bread, close door2, cut salami, close drawer1, cut bread, close fridge, release bottle, reach bottle, release salami, reach salami, release bread, reach bread, release sugar, reach sugar, release dishwasher, reach dishwasher, release switch, reach switch, release milk, reach milk, release drawer3, reach drawer3, release spoon, reach spoon, release knife cheese, reach knife cheese, release drawer2, reach drawer2, release table, reach table, release glass, reach glass, release cheese, reach cheese, release chair, reach chair, release door1, reach door1, release door2, reach door2, release plate, reach plate, release drawer1, reach drawer1, release fridge, reach fridge, release cup, reach cup, release knife salami, reach knife salami, release lazychair, reach lazychair, move bottle, open bottle, move salami, open salami, move bread
	right-hand action and object(112)	reach drawer2, release salami, reach table, release bread, reach glass, release sugar, reach cheese, release dishwasher, reach chair, release switch, reach door1, release milk, reach door2, release drawer3, reach plate, release spoon, reach drawer1, release knife cheese, reach fridge, release drawer2, reach cup, release table, unlock knife cheese, reach knife salami, release glass, unlock door1, reach lazychair, release cheese, unlock door2, open bottle, release chair, stir spoon, open salami, release door1, stir glass, open bread, release door2, stir cheese, open dishwasher, release plate, stir cup, open milk, release drawer1, lock door1, open drawer3, release fridge, lock door2, open drawer2, release cup, close bottle, open cheese, release knife salami, close salami, open door1, release lazychair, close bread, open door2, move bottle, close dishwasher, open drawer1, move salami, close milk, open fridge, move bread, close drawer3, sip glass, move sugar, close drawer2, sip cup, move dishwasher, close cheese, clean salami, move milk, close door1, clean table, move drawer3, close door2, clean plate, move spoon, close drawer1, bite bread, move knife cheese, close fridge, cut salami, move glass, reach bottle, cut bread, move cheese, reach salami, cut cheese, move chair, reach bread, cut knife salami, move plate, reach sugar, spread bottle, move fridge, reach dishwasher, spread bread, move cup, reach switch, spread sugar, move knife salami, reach milk, spread milk, move lazychair, reach drawer3, spread knife cheese, reach spoon, spread cheese, reach knife cheese, release bottle
high-level(5)	-	relax, coffee time, early morning, cleanup, sandwich time

requires simple point-based formulas to be hand-coded. In CK, formulas are designed based on common knowledge, describing the implication of the occurrence of an activity between start-time and end-time, and the dependence between the occurrences of a primitive activity and a higher-level activity. We use Tuffy [31], which is an open-source MLN inference engine, to achieve required inferences.

6.2.1. Primitive event recognition

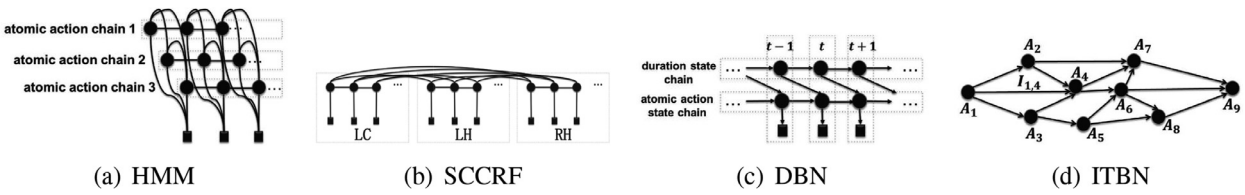
For the Opportunity dataset, we conducted two separate experiments, i.e., in the ideal situation where all primitive events are assumed to correctly detect and in real scenarios where primitive events are inferred by sensors. In the second case, we used the activity recognition chain (ARC) system developed by Bulling et al. [8] to implement the primitive event recognition from sensors. The observation sliding window is set to 2 min, which is the maximum duration time of continuous occurrences of primitive events in the dataset. In particular, we used kNN, Decision Table(DT), C4.5 and SVM for primitive event recognition in our experiments. We first classify each sensor record to locomotion, left-hand action and right-hand action, respectively. For each of these three primitive tracks, eight classifiers are built by combining the four approaches and two time-sliced windows for feature extraction (i.e. 0.33 s and 1 s). We adopt features such as mean, variance, energy, entropy and correlation. After classification, each sensor record is assigned to three labels, i.e. a locomotion, a left-hand action and a right-hand action.

We adopt the approach for primitive event recognition proposed by Zhang et al. [49]. Two categories of features (i.e. kinematic and image based) are first extracted from the bounding box of the computed tracks of the players in the videos. Then, a collection of dynamic Bayesian network models are used to model each primitive activity. During the six basic basketball play activity recognition, a sliding window moves across a video and at each segment in the sliding video each player is tested against the collection of the DBN models, and is assigned to the primitive activity that corresponds to the model with the highest likelihood.

6.2.2. Implementation of the competing approaches

In what follows we illustrate how to implement the competing approaches, including ARF, for complex activity recognition on the Opportunity dataset. Note that the rational and procedure to implement them for OSUPEL dataset is similar to that for Opportunity dataset.

In HMM, sensor-extracted features or sensor tagged objects are often used as observations whilst activities are defined as hidden states. HMMs are more suitable for purely sequential activities. To model the events of locomotions, left-hand



(a) HMM (b) SCCRF (c) DBN (d) ITBN

Fig. 6. (a) HMM, where the observations of primitive events (square-shaped nodes) and several chains of hidden states (round-shaped nodes) are used to handle overlapping; (b) SCCRF with three chains, where the round-shaped nodes represent the hidden states and the square-shaped nodes represent the observations associated with the three chains (i.e. locomotion, left-hand action, right-hand action); (c) DBN, where duration states and primitive event states are represented as chains of nodes; (d) ITBN, where each node represents a type of primitive activities and each link $I_{i,j}$ represents the set of temporal relations between i and j. The triangles 124 and 134 have a common link $I_{1,4}$ that must be temporally consistent with each other.

Table 4
The set of formulas in CK and ARF.

CK	ARF
①-③,⑪-⑫	①-⑪,⑬
hard formulas	
∀ primitive activities X, Y, Z, timesteps t, $t1$, $t2$, $t3$, $t4$:	
① $occurLC(X,t) \vee occurLH(X,t) \vee occurRH(X,t) \Leftrightarrow occur(X)$	
② $X \neq Y \Rightarrow (occurLC(X,t) \Rightarrow \neg occurLC(Y,t)) \wedge (occurLH(X,t) \Rightarrow \neg occurLH(Y,t)) \wedge (occurRH(X,t) \Rightarrow \neg occurRH(Y,t))$	
③ $startEvent(X,t1) \wedge endEvent(X,t2) \wedge t \geq t1 \wedge t \leq t2 \Rightarrow occur(X)$	
④ $endEvent(X,t1) \wedge startEvent(Y,t2) \wedge t1 < t2 \Rightarrow before(X,Y)$	
⑤ $startEvent(X,t1) \wedge startEvent(Y,t2) \wedge endEvent(X,t3) \wedge endEvent(Y,t4) \wedge t3 > t2 \wedge t1 \neq t2 \wedge t4 \neq t3 \Rightarrow overlap(X,Y)$	
⑥ $endEvent(X,t1) \wedge startEvent(Y,t2) \wedge t1 = t2 \Rightarrow meet(X,Y)$	
⑦ $startEvent(X,t1) \wedge startEvent(Y,t2) \wedge endEvent(X,t3) \wedge endEvent(Y,t4) \wedge t1 = t2 \wedge t3 < t4 \Rightarrow start(X,Y)$	
⑧ $startEvent(X,t1) \wedge startEvent(Y,t2) \wedge endEvent(X,t3) \wedge endEvent(Y,t4) \wedge t1 < t2 \wedge t3 > t4 \Rightarrow contain(X,Y)$	
⑨ $startEvent(X,t1) \wedge startEvent(Y,t2) \wedge endEvent(X,t3) \wedge endEvent(Y,t4) \wedge t1 < t2 \wedge t3 = t4 \Rightarrow finished-by(X,Y)$	
⑩ $startEvent(X,t1) \wedge startEvent(Y,t2) \wedge endEvent(X,t3) \wedge endEvent(Y,t4) \wedge t1 = t2 \wedge t3 = t4 \Rightarrow equal(X,Y)$	
∀ high-level activity X, Y:	
⑪ $X \neq Y \Rightarrow (highlevel(X) \Rightarrow \neg highlevel(Y))$	
soft formulas	
∀ primitive activities X, Y, high-level activity Z:	
⑫ $occur(X) \Rightarrow highlevel(Z)$	
⑬ All learned formulas from EKB.	

actions and right-hand actions that may occur in parallel, we used the interleaved HMM with three chains that can capture both inter-event and intra-event dynamics, as shown in Fig. 6(a). SCCRF is an extension of the linear-chain CRF to model high-level activities with concurrent and interleaved primitive activities by multiple linear chains. As shown in Fig. 6(b), we define three linear chains associated with the locomotions (LC), the left-hand actions (LH) and the right-hand actions (RH), respectively. Each chain can only capture simple temporal relations between primitive activities (i.e. precede and follow). DBN represent the primitive event and their duration as two separate chains of nodes, as shown in Fig. 6(c). Oliver and Horvitz [32] investigated HMM and dynamic BN (DBN) and found that DBN can learn dependencies between variables that were assumed independent in HMM, but HMM imposes less computational burden than arbitrary DBN. ITBN is a graphical model that combines the Bayesian network with the Allen's interval algebra to explicitly model the temporal relations between primitive activities. Each node represents a primitive activity, and each link represents the temporal relations between two activities. Since Bayesian network is a directed acyclic graph, as shown in Fig. 6(d), any relations that may make the graph temporally inconsistency must be removed from the training dataset. In addition, ITBN only can contain an activity appearing at most once in the acyclic graph. In our experiment, if an activity occurs multiple times its corresponding node represents the first interval only. It is worth noting that all the graphical models mentioned above can only be applied as classifiers to recognize high-level activities from primitive events.

Table 4 shows the *hard* and *soft* formulas contained in the two semantic-based methods. The predicate *occurLC*(X, t) models the occurrence of a locomotion X at time step t. The *occurLH*(X, t) and *occurRH*(X, t) have the same meaning for left-hand and right-hand action, respectively. The predicate *startEvent*(X, t) and *endEvent*(X, t) indicate the start time and the end time of the event, respectively. CK only contains three hard formulas from common knowledge that describe the equivalence of the event of a locomotion (or a left-hand action, or a right-hand action) and the event of a primitive activity (formula ①), the impossibility of two locomotions (or left-hand actions, or right-hand actions) occurring simultaneously (formula ②), and the implication of the event of an activity between start-time and end-time (formula ③). CK includes one soft formula (⑫) that models the probabilistic dependencies between the events of a primitive activity and a high-level activity. The weights are calculated by counting the frequencies at which the primitive activity appears in the high-level activity. ARF not only uses the learned formulas, but also combines the formulas from common knowledge to form an EKB.

Table 5
Examples of highly confident learned formulas and their weights (with $minconf = 0.5$, $minsup = 0.0005$).

Formulas	Weights
occur(lie) → highlevel(relax)	1.0
occur(reach lazychair) → highlevel(relax)	1.0
occur(clean table*) → highlevel(cleanup)	1.0
occur(bite bread) → highlevel(sandwich time)	1.0
finished-by(move cup, stand) → highlevel(cleanup)	1.0
meet(stand, walk) ∧ before(stand, stand^2) ∧ before(stand, move cup) ∧ meet(walk, stand^2) ∧ before(walk, move cup) ∧ contain(stand^2, move cup) → highlevel(coffee time)	0.86
occur(unlock door) → predict(walk)	1.0
contain(sit, move cup) ∧ contain(sit, move bread) ∧ before(move cup, move bread) → predict(drink from cup*)	1.0
occur(reach bread) → predict(open fridge)	0.62
occur(unlock door) → verify(stand)	1.0
meet(sip cup, move cup) ∧ before(sip cup, release cup) ∧ meet(move cup, release cup) → verify(drink from cup*)	1.0
contain(stand, release bottle) ∧ meet(stand, walk) ∧ before(release bottle, walk) → verify(walk)	0.64
occur(open fridge) → discover(stand)	1.0
occur(close fridge) → discover(walk)	0.93
occur(reach switch) → discover(stand)	0.90

Note that we use the abbreviated term, for example, highlevel(Y) to represent highlevel(Y; $X_1, \ldots, X_n$) for space saving. * - mid-level activity. ^{2} - the second time an activity appears in the same formula.

Formulas ④-⑩ model the relationships between temporal relations and their start-time and end-time. An additional hard formula ⑪ is added in all the methods to ensure that any two different high-level activities cannot occur simultaneously.

(MLN inference tool) We use Tuffy [31], which is an open-source MLN inference engine, to achieve required inferences. We refer the reader to the users' manual[3] of Tuffy. Patterns are discovered by ARPG algorithm from the training dataset and are converted to CNF formulae in Tuffy input format[4].

7. Results

In this section, we report the activity recognition results from the sensor-based opportunity dataset and the vision-based OSU basketball play dataset.

7.1. Experiment I - opportunity dataset

7.1.1. Learned formulas in EKB

In this experiment, we consider to generate a user-independent and date-independent EKB. Formulas discovered by our learning algorithm are converted to Tuffy input format. The observation sliding window is set to 2 min, which is the maximum duration time of continuous occurrences of primitive events in the dataset. Since *null* event occupies a high percentage of time in the dataset, we set *minsup* to a small value of 0.0005. The minimum confidence value *minconf* is set to 0.5. A total number of 2364 formulas with their weights are learned from ADL1-5 as training data. Particularly, the learned EKB contains 706, 1135, 315 and 138 formulas associated with the four query predicates, respectively.

Table 5 shows several examples of highly confident formulas. It can be seen that some primitive activities or mid-level activities are specific to a high-level activity. For instance, `lie` and `reach lazychair` are connected with `relax` only, and `clean table` and `bite bread` are only connected with `cleanup` and `sandwich time`, respectively. Also, it can be seen that event relations play an essential role in high-level activity recognition. For example, it is hard to discriminate between the occurrence of `coffee time` and `cleanup` if `move cup` and `stand` are observed without their relationship. If `move cup` is observed during `stand`, it is more confident that `coffee time` is occurring. It is quite straightforward to understand the semantic meanings of these formulas. Notice that activities of different levels may appear in the same formula.

The learning algorithm with various settings on *minsup* and *minconf* may generate different EKBs. We changed the values of *minsup* from 0.0001 to 0.001, and *minconf* from 0.3 to 0.8. The total number of generated formulas varies from 998 to 4862. A set of more formulas can make the EKB describing event relations more precisely, but it may also lead to a computational disaster for inference on MLN. In this case, we found that a set of more than 800 formulas associated with

[3] http://i.stanford.edu/hazy/tuffy/doc/.
[4] http://alchemy.cs.washington.edu/user-manual/.

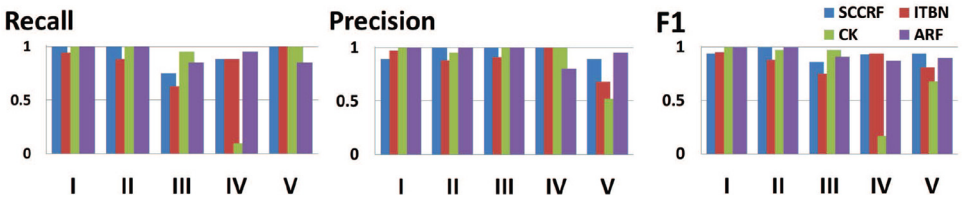


Fig. 7. Comparison results without error on primitive event recognition. I-`relax`; II-`coffee time`; III-`early morning`; IV-`cleanup`; V-`sandwich time`.

Table 6
Performance comparison under detection errors of primitive events.

Low-level classifiers		Error(%)	High-level recognition approaches			
			SCCRF	ITBN	CK	ARF
No Error		0	0.94	0.88	0.76	**0.94**
kNN	w20	16.5	0.69	0.54	0.71	**0.81**
	w60	19.4	0.50	0.79	0.76	**0.86**
SVM	w20	24.2	0.10	0.46	0.72	**0.79**
	w60	25.0	0.10	0.75	0.74	**0.81**
C4.5	w20	24.3	0.13	0.29	0.64	**0.69**
	w60	24.1	0.24	0.41	0.64	**0.69**
DT	w20	31.5	0.10	0.38	0.52	**0.69**
	w60	33.5	0.10	0.54	0.38	**0.64**

* w20=sliding window size of 0.33s; w60=sliding window size of 1s.

`highlevel` query cannot significantly improve high-level activity classification performance. Therefore, we set $minsup = 0.0005$ and $minconf = 0.5$ in our following experiments.

7.1.2. Comparison against other approaches on high-level activity classification

This section reports our comparison study on high-level activity classification. We compare the classification performance of our ARF with two graphical models (i.e. SCCRF and ITBN) and one semantic-based model (i.e. CK). We use leave-one-run-out cross-validation for evaluation, that is to say, any five runs are used as the training dataset and the remaining one is used as the testing dataset. We apply three metrics for the evaluation: recall, precision and F1 score.

Fig. 7 depicts the comparison results of the average recall, precision and F1 over six-fold cross-validations without primitive event recognition errors. It can be seen that CK falsely recognizes most of `cleanup` as `sandwich time`. This is because these two activities involve many common hand actions. It is hard to distinguish between them using the occurrence frequencies of primitive activities only. By contrast, ITBN and ARF, both of which consider the event relations, achieve better performance on F1. SCCRF and ARF achieve similarly good results, which coincide with the conclusion [20] that SCCRF can capture long-range dependencies. However, the performance of SCCRF is significantly affected by the observations that are insufficiently or erroneously provided, as shown in the following experiments. **Performance under incorrect primitive event recognition.** Table 6 depicts the comparison results of F1 score on high-level activity classification under various primitive event recognition errors. It is clear from the table that the semantic-based approaches outperform the graphical model-based approaches. Regarding semantic-based approaches, classification results are influenced by incorrect observations and the weights assigned to formulas. Particularly, the incorrect recognition of event relations may cause ARF to receive incorrect evidences for inference. We found that ARF is especially sensitive to two types of event recognition errors, that is, the error that a *null* is predicted as an activity, and the error that an activity is predicted as *null*. ITBN is also significantly affected by these two types of errors as it uses event relations for inference. SCCRF is highly sensitive to the errors of missing recognition, which leads to insufficient observations in chains. In summary, our experiment shows that ARF is robust and sufficient in handling errors resulting from noisy records. Compared with other approaches, ARF consistently achieves higher performance under realistic primitive event recognition with varying data noise.

Performance under incomplete observations of primitive events. In the previous experiments, the high-level activity classification rests on the entire records of primitive events are fully observed. In Opportunity dataset, a full length of *coffee time, cleanup* and *sandwich time* contain greater than 80 events on average. It is impractical to provide classification results after all the primitive events are detected. For clarity, we conduct an experiment to compare the results over primitive events are partially observed. We divide each training record into a collection of *sub-records* of continuous primitive events of length k, where $1 \leq k \leq K$ and K refers to the total number of events in the record.

Fig. 8 shows the comparison results of the high-level classification on ADL5 over incomplete observations. Overall, the semantic-based approaches outperform the graphical model-based approaches. The trend of F1 score on SCCRF is nearly convex as the number of observed events increases, while the trend on ITBN is concave. It can be seen that both SCCRF and ITBN require enough observations for inference. Any missing information may result in failure in their structure con-

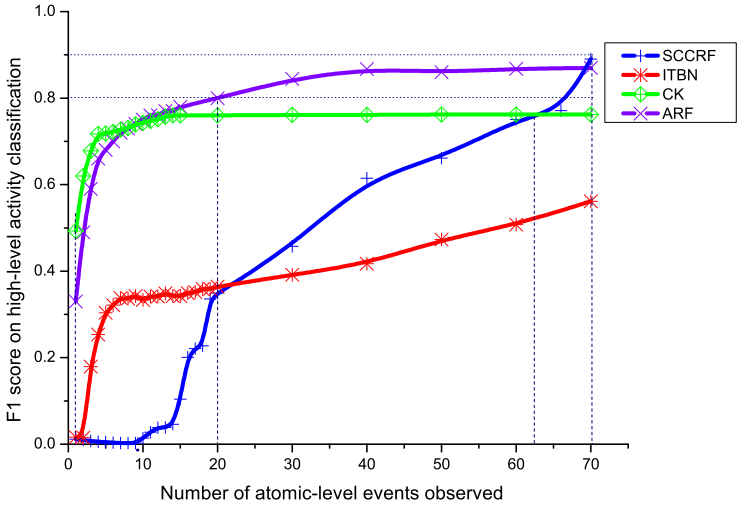


Fig. 8. Performance comparison over varied numbers of detected primitive events.

Table 7
The comparison results on two types of offensive basketball play recognition.

	HMM		SCCRF		DBN		ITBN		CK		ARF	
Actual(%)/Predicted(%)	PT-I	PT-II	PT-I	PT-II	PT-I	PT-II	PT-I	PT-II	PT-I	PT-II	PT-I	PT-II
PT-I	50	50	64.29	35.71	53.57	46.43	60.71	39.29	89.29	10.71	75	25
PT-II	32.5	67.5	25	75	25	75	25	75	100	0	32.5	67.5
Accuracy (%)	52.78		66.67		58.33		69.44		69.44		72.22	

structions. It is clear from the figure that ITBN needs more observations than SCCRF for inference although it utilizes event relations. The reason is that ITBN requires more complex event relations than the chain-based SCCRF to construct its structure. It can also be seen that the semantic-based approaches are more robust on incomplete observations. The F1 score increases sharply to 0.8 until the number of observed events reaches 20. ARF gets more accurate results than CK when more than 20 events are observed. This is because ARF can take advantage of event relations, which do not contained in CK. In summary, the exploitation of event relations can improve the high-level classification performance under incomplete observations of primitive events.

7.2. Experiment II - OSUPEL basketball play dataset

A total of 226 formulas have been learned in EKB. Table 7 shows the recognition results of the two offensive plays produced by the six approaches, i.e. HMM, SCCRF, DBN, ITBN, CK and ARF. It seems that the semantic-based approaches perform better than the graphical model-based approaches. We can further see that our ARF approach outperforms other models. For clarity, we also provide the classification accuracies of each model. We found that DBN can learn temporal dependencies between two primitive events that are assumed independent in HMM. SCCRF is able to catch the dependencies between two overlapping primitive events. DBN and ITBN recognizes PT-I slightly better than ARF but much worse for PT-II. This is because both of them are built on Bayesian network that is a directed acyclic graph. They have to remove some temporal relations from the training dataset in order to make the Bayesian network temporally consistent, resulting in information loss during model training. Generally, ARF's capacity of performing accurate recognition is mainly due to its ability to take advantage of temporal relations among events. In addition, in order to recognize new activities, the graphical model-based approaches have to reconstruct network structures. The semantic-based model only need to append formulas related to new activities into EKB. Additionally, the learned formulas can be reused, and the inference system can be implemented efficiently.

7.3. Runtime

All of the competing approaches are different in computational complexity. Three variables may affect the time complexities of these approaches for recognizing high-level activities, i.e. the number of primitive activity types, the number of events per record, and the number of training (or testing) records per high-level activity. Fig. 9 shows the comparison results of the empirical runtime performance on different settings by varying one variable while fixing others. We investigate the training and inference stages of each approach separately. Note that the runtime for pattern support calculation is also

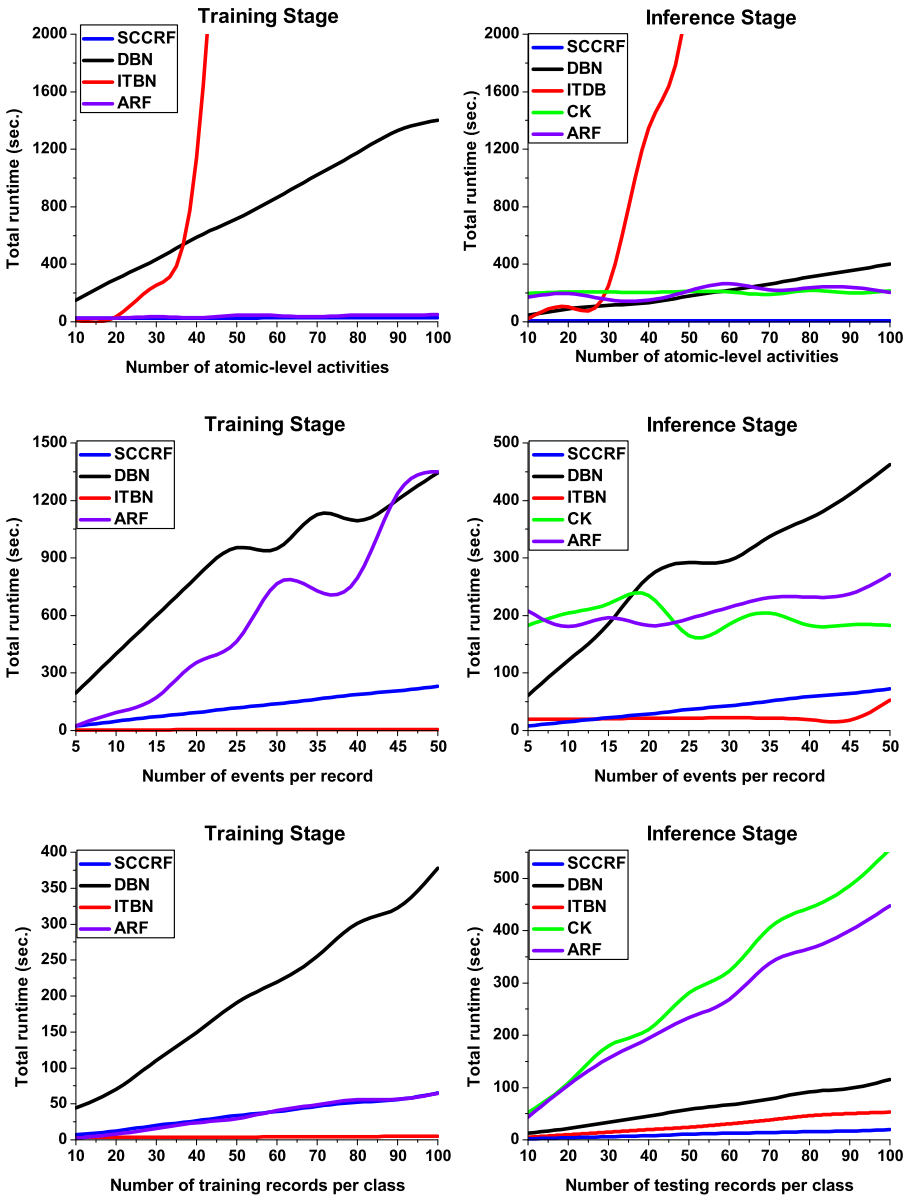


Fig. 9. Runtime comparison results under different settings.

counted in the training stage of ARF, and the training stage of CK is not presented because the formulas in CK are manually encoded.

We can see that ARF performs well at training stage on average. Its time complexity is mainly dependent on the number of events. The reason is that the number of patterns generated by our algorithm will increase as the number of events in a record increases. For graphical model-based approaches, SCCRF performs better than DBN and ITBN overall on runtime performance because it is a probabilistic model over undirected graph and needs not learn a large number of parameters compared with the directed graph. Particularly, it is clear that ITBN is extremely and only sensitive to the number of primitive activity types. This is because ITBN needs to check temporal consistency and evaluate all possible network structures, and thus takes much more time as the number of nodes increases. We found that it is impractical to train an ITBN for a dataset containing more than 40 activity types. As for inference stage, we found that ARF approximately takes five seconds for each testing, including the time for database connection in Tuffy. Considering the usually long duration of complex activities, the runtime of five seconds can be deemed to be an acceptable delay in most applications. As part of future work, we will improve the runtime performance of ARF at the inference stage.

8. Conclusion

In this paper, we provide a novel probabilistic event relational learning framework to learn semantic-level activity models and address temporally and hierarchically related semantic queries under uncertainty. It extends the Allen's temporal relations by appending one more hierarchical relation. As such, it is able to answer various queries about the probabilities of activity occurrences in a unified manner.

Understanding complex activities requires not only capturing their temporal dependencies among primitive events, but also conveying hierarchical relations. Psychologists have found that most human behaviors are hierarchically structured [48]. This inspires us to define a unified way to represent the relations between events at different levels. Also, semantic-level understanding of human activity is essential for applications. As shown in our experiments, our framework can be applied to answer various semantic queries about events and their relations, e.g., "if washing cups and cleaning tables are observed, is it highly likely that Alice is doing cleanup?", or "if washing cups and plates are observed, what is the possibility that Alice will clean tables next?". These semantic queries are associated with temporal and multilevel event relations, and have practical usage and vast potential in human activity recognition applications such as healthcare [4], aged care [41], sports [3], smart homes [24] and product recommendations [44]. For example, in elderly care applications, not only is an activity recognition system needed to detect whether a basic daily activity is occurring, but the system can also forecast an ongoing abnormal activity to avoid possible injury through a warning. In addition, when the detection of primitive activities is poor because of sensor failure or background noise, verifying recognition results inferred in the past few minutes can improve a recognition system's robustness by estimating the probability of the missing or false recognition of an activity. As shown in our experiments, a number of frequent and confident formulas were discovered by the ARF framework that employs pattern mining techniques. By employing MLN, the ARF framework is capable of integrating diverse formulas either hand-coded by domain experts or learned from the training dataset and inferring various semantic-based queries in a unified way.

With respect to assessing the efficiency and robustness of different approaches, we evaluated several situations where primitive events are detected. In general, the exploitation of interval relations can improve the high-level classification performance under partial low-level activities are observed. The competing approaches, including ARF, which consider the temporal-interval-related relations between events achieved better performance than others. The ARF framework can also capture long-range dependencies efficiently. Most importantly, the ARF framework achieves the best performance under realistic primitive event recognition with varying data noise. Other approaches, especially the graphic models, are significantly affected by the false-detection errors or sensitive to the errors of missing primitive events.

Our evaluation of the ARF framework was not designed to examine or enhance the pattern mining technique per se, but rather, using them in formulas learning, to answer the specific research questions that we raised. However, as part of our evaluation, we have quantitatively measured the runtime of the competing approaches. We found that the ARF framework needs more time in both training and testing stages than most of other approaches. Especially, when the number of events per records increases, the time complexity of the ARF framework grows sharply. This is due to the fact that most of computational time are spent on formula learning. Although many existing work on accelerating the process of pattern mining cannot be applied in formula learning directly, we will investigate these approaches to find out a better solution to reduce runtime and memory usage in future research. An alternative is to employ the transitivity-based method for pattern discovery [28].

In summary, our novel ARF framework brings about several benefits in recognizing complex activities:

1. Our framework can automatically learn, and thus avoids to manually encode, formulas and their weights associated with event relations. Our experiments show that these learned formulas involving temporal and hierarchical dependencies have significant impact on the performances of activity inferences, especially with noise and missing observations.
2. Our framework can be applied for answering varied queries about the probabilities of activity occurrences. In this paper, we focus on four commonly performed queries. Other queries can also be implemented through satisfying their corresponding event relational properties.
3. We envision that our mined event relational formulas can be combined with existing semantic-based systems, especially for the applications where event relations are intricate and difficult to hand-code.

As part of future work, we will continue exploring learning EKBs that contains more complex formulas with quantifiers such as first-order logic formulas rather than only propositional logic formulas in this work. One strategy would be to use of universal formula generalization technique, in which formulas can be generalized into a smaller set without information loss. For example, we can leverage the transitive properties among event relations in Allen's algebra, e.g. $\mathtt{X}\,before\,\mathtt{Y}$ and $\mathtt{Y}\,before\,\mathtt{Z}$ imply $\mathtt{X}\,before\,\mathtt{Z}$. These properties can be used for the generalization. Following the example, a generalized formula $\forall x, y, z : x \texttt{ before } y \wedge y \texttt{ before } z \rightarrow x \texttt{ before } z$ is included in an EKB, and any instances of the form $x \texttt{ before } z$, e.g. $\mathtt{X}\,before\,\mathtt{Z}$, can be removed. Future work includes the generalization of probabilistic event relational formulas. In this way, it can condense the EKB and also strengthen the expressive power to describe human understandable knowledge.

Acknowledgment

This work was supported by grants from the Fundamental Research Funds for the Central Universities in China (grant nos. CQU903005203326, CQU0225001104447), the Science and Technology Innovation Project of Foshan City in China (grant

no. 2015IT100095), the National Basic Research Program of China (973 Program) (grant no. 2013CB328903, 2014CB744600), the National Natural Science Foundation of China (grant nos. 61401183, 60973138, 61003240), the International Cooperation Project of Ministry of Science and Technology (grant no. 2013DFA11140), the Key Research Programm of Chongqing Science & Technology Commission (grant no. cstc2017jcyjBX0025) and the Major Science and Technology Program of Guanxi Province (grant no. GKAA17129002).

References

[1] J.K. Aggarwal, M.S. Ryoo, Human activity analysis: a review, ACM Comput. Surv. 43 (3) (2011) 16.
[2] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1983) 832–843.
[3] K. Altun, B. Barshan, Human activity recognition using inertial/magnetic sensor units, in: Human Behavior Understanding, Springer, 2010, pp. 38–51.
[4] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, P. Havinga, Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey, in: Architecture of Computing Systems (ARCS), 2010 23rd International Conference on, VDE, 2010, pp. 1–10.
[5] I. Batal, H. Valizadegan, G.F. Cooper, M. Hauskrecht, A temporal pattern mining approach for classifying electronic health record data, ACM Trans. Intell. Syst. Technol. 4 (4) (2013) 63.
[6] U. Blanke, B. Schiele, Remember and transfer what you have learned-recognizing composite activities based on activity spotting, in: Wearable Computers (ISWC), 2010 International Symposium on, IEEE, 2010, pp. 1–8.
[7] W. Brendel, A. Fern, S. Todorovic, Probabilistic event logic for interval-based event recognition, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3329–3336.
[8] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Comput. Surv. 46 (3) (2014) 33.
[9] L. Chen, J. Hoey, C.D. Nugent, D.J. Cook, Z. Yu, Sensor-based activity recognition, Syst. Man Cybern. Part C 42 (6) (2012) 790–808.
[10] Y.C. Chen, J.C. Jiang, W.C. Peng, S.Y. Lee, An efficient algorithm for mining time interval-based patterns in large database, in: ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October, 2010, pp. 49–58.
[11] Y. Du, F. Chen, W. Xu, Y. Li, Recognizing interaction activities using dynamic bayesian network, in: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 1, IEEE, 2006, pp. 618–621.
[12] K.S.R. Dubba, A.G. Cohn, D.C. Hogg, M. Bhatt, F. Dylla, Learning relational event models from video, J. Artif. Intell. Res. 53 (1) (2015) 41–90.
[13] A. Fern, R. Givan, J.M. Siskind, Specific-to-general learning for temporal events with application to learning event definitions from video, J. Artif. Intell. Res. 17 (1) (2011) 2002.
[14] T. Gu, L. Wang, Z. Wu, X. Tao, J. Lu, A pattern mining approach to sensor-based human activity recognition, Knowl. Data Eng. IEEE Trans. 23 (9) (2011) 1359–1372.
[15] A. Gupta, P. Srinivasan, J. Shi, L.S. Davis, Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos, in: IEEE Conference on Computer Vision & Pattern Recognition, 2009, pp. 2012–2019.
[16] R. Helaoui, M. Niepert, H. Stuckenschmidt, Recognizing interleaved and concurrent activities: a statistical-relational approach, in: Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on, IEEE, 2011, pp. 1–9.
[17] R. Helaoui, D. Riboni, H. Stuckenschmidt, A probabilistic ontological framework for the recognition of multilevel human activities, in: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, ACM, 2013, pp. 345–354.
[18] F. Höppner, Discovery of temporal patterns, in: Principles of Data Mining and Knowledge Discovery, Springer, 2001, pp. 192–203.
[19] F. Höppner, F. Klawonn, Finding informative rules in interval sequences, in: Advances in Intelligent Data Analysis, Springer, 2001, pp. 125–134.
[20] E. Kim, S. Helal, D. Cook, Human activity recognition and pattern discovery, Perv. Comput. IEEE 9 (1) (2010) 48–53.
[21] K. Li, Y. Fu, Prediction of human activity by discovering temporal sequence patterns, Pattern Anal. Mach. Intell. IEEE Trans. 36 (8) (2014) 1644–1657.
[22] L. Liu, Y. Peng, M. Liu, Z. Huang, Sensor-based human activity recognition system with a multilayered model using time series shapelets, Knowl. Based Syst. 90 (2015) 138–152.
[23] L. Liu, Y. Peng, S. Wang, M. Liu, Z. Huang, Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors, Inf. Sci. 340–341 (2016) 41–57.
[24] L. Liu, S. Wang, Y. Peng, Z. Huang, M. Liu, B. Hu, Mining intricate temporal rules for recognizing complex activities of daily living under uncertainty, Pattern Recognit. 60 (2016) 1015–1028.
[25] L. Liu, S. Wang, G. Su, Z. Huang, M. Liu, Towards complex activity recognition using a bayesian network-based probabilistic generative framework, Pattern Recognit. 68 (2017) 295–309.
[26] Y. Liu, L. Nie, L. Liu, D.S. Rosenblum, From action to activity: sensor-based activity recognition, Neurocomputing 181 (2016) 108–115.
[27] F. Mörchen, Unsupervised pattern mining from symbolic temporal data, ACM SIGKDD Explor. Newslett. 9 (1) (2007) 41–55.
[28] R. Moskovitch, Y. Shahar, Fast time intervals mining using the transitivity of temporal relations, Knowl. Inf. Syst. (2013) 1–28.
[29] R. Moskovitch, Y. Shahar, Classification of multivariate time series via temporal abstraction and time intervals mining, Knowl. Inf. Syst. 45 (1) (2015) 35–74.
[30] R. Moskovitch, Y. Shahar, Fast time intervals mining using the transitivity of temporal relations, Knowl. Inf. Syst. 42 (1) (2015) 1–28.
[31] F. Niu, C. Ré, A. Doan, J. Shavlik, Tuffy: scaling up statistical inference in markov logic networks using an rdbms, Proc. VLDB Endow. 4 (6) (2011) 373–384.
[32] N. Oliver, E. Horvitz, A comparison of hmms and dynamic bayesian networks for recognizing office activities, in: User Modeling 2005, Springer, 2005, pp. 199–209.
[33] P. Papapetrou, G. Kollios, S. Sclaroff, D. Gunopulos, Mining frequent arrangements of temporal intervals, Knowl. Inf. Syst. 21 (2) (2009) 133–171.
[34] D. Patel, W. Hsu, M.L. Lee, Mining relationships among interval-based events for classification, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ACM, 2008, pp. 393–404.
[35] M. Pei, Z. Si, B.Z. Yao, S.C. Zhu, Learning and parsing video events with goal and intent prediction, Comput. Vision Image Understanding 117 (10) (2013) 1369–1383.
[36] M. Richardson, P. Domingos, Markov logic networks, Mach. Learn. 62 (1–2) (2006) 107–136.
[37] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Troster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al., Collecting complex activity datasets in highly rich networked sensor environments, in: Networked Sensing Systems (INSS), 2010 Seventh International Conference on, IEEE, 2010, pp. 233–240.
[38] M.S. Ryoo, J.K. Aggarwal, Semantic representation and recognition of continued and recursive human activities, Int. J. Comput. Vis. 82 (1) (2009) 1–24.
[39] L. Sacchi, C. Larizza, C. Combi, R. Bellazzi, Data mining with temporal abstractions: learning rules from time series, Data Min. Knowl. Discov. 15 (2) (2007) 217–247.
[40] S. Saguna, A. Zaslavsky, D. Chakraborty, Complex activity recognition using context-driven activity theory and activity signatures, ACM Trans. Comput. Human Interact. 20 (6) (2013) 32.
[41] N. Suryadevara, S.C. Mukhopadhyay, R. Wang, R. Rayudu, Forecasting the behavior of an elderly using wireless sensors data in a smart home, Eng. Appl. Artif. Intell. 26 (10) (2013) 2641–2652.
[42] D.L. Vail, M.M. Veloso, J.D. Lafferty, Conditional random fields for activity recognition, in: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, 2007, p. 235.
[43] H. Veeraraghavan, N. Papanikolopoulos, P. Schrater, Learning dynamic event descriptions in image sequences, in: IEEE Conference on Computer Vision & Pattern Recognition, 2007, pp. 1–6.

[44] X. Wang, D. Rosenblum, Y. Wang, Context-aware mobile music recommendation for daily activities, in: Proceedings of the 20th ACM International Conference on Multimedia, ACM, 2012, pp. 99–108.
[45] E. Winarko, J.F. Roddick, Armada–an algorithm for discovering richer relative temporal association rules from interval-based data, Data Knowl. Eng. 63 (1) (2007) 76–90.
[46] S.Y. Wu, Y.L. Chen, Mining nonambiguous temporal patterns for interval-based events, IEEE Trans. Knowl. Data Eng. 19 (6) (2007) 742–758.
[47] J. Ye, G. Stevenson, S. Dobson, Usmart: an unsupervised semantic mining activity recognition technique, ACM Trans. Interact. Intell. Syst. 4 (4) (2014) 16.
[48] J.M. Zacks, B. Tversky, Event structure in perception and conception., Psychol. Bull. 127 (1) (2001) 3.
[49] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, Q. Ji, Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (10) (2013) 2468–2483.