Journal of Reliable Intelligent Environments manuscript No.
(will be inserted by the editor)

Complex Behavioral Pattern Mining in Non-intrusive Sensor-based Smart Homes using an Intelligent Activity Inference Engine

Jianguo Hao · Abdenour Bouzouane · Sébastien Gaboury

Received: 2 February 2017 / Accepted: 16 May 2017

Abstract With the help of cutting-edge software, hardware and communication technologies, future living environments such as smart homes are expected to provide intelligent services to make user experience much better than ever. Ideally, indoor multi-modal interactions and ambient changes in smart homes could be monitored using ambient intelligence technology for instantaneous human behavior analysis and human-centered interactions. Such quick feedbacks are significant to the elderly and disabled people. Qualified smart environments could greatly reduce the burdens on personal home cares by offering appropriate assistance according to recognized human behaviors. However, due to varied behavioral patterns and imprecise data acquisition, realizing complex human activity in real-time without controversy over privacy is one of the biggest challenges for the scientific community. In this paper, we propose an innovative inference engine based on Formal Concept Analysis to recognize complex human activities, including sequential, interleaved and concurrent patterns performed in a non-intrusive smart home. Besides the on-the-fly recognition, our engine adopts a loose coupling architecture that strengthens the software robustness, reliability and knowledge reusability.

Keywords Ambient intelligence · Smart homes · Pervasive computing · Behavioral pattern mining · Formal concept analysis

J. Hao · A. Bouzouane · S. Gaboury
LIARA Laboratory, Université du Québec à Chicoutimi
555 Boulevard de l'Université, Chicoutimi, G7H 2B1, Canada
e-mail: jianguo.hao1@uqac.ca

A. Bouzouane
e-mail: abdenour.bouzouane@uqac.ca

S. Gaboury
e-mail: sebastien.gaboury@uqac.ca

1 Introduction

As people live longer and low birth rates all over the world, population aging has become a worldwide crucial issue in the recent decades [1]. When cheering for the great improvement in health and longevity, we also need to calmly think about its tremendous challenges to the social and family aspects [2]. Elderly people usually require more long-term home cares because of the gradually weakened physical conditions, especially the aging-related disabilities. As one of the most typical ones, cognitive disability has attracted much attention from scientific communities [3]. According to the recent research [4], most cognitive impairment symptoms happened in the middle-aged and elderly population are related to the Alzheimer's disease, which is also the most common cause of dementia. Progressively severe deterioration in cognitive skills makes elderly people have difficulties in living independently. Comparing with healthy people, they tend to produce more abnormal behaviors which make normal activities become threatening as well [5]. The most direct and efficient intervention is to employ family caregivers to take care of elderly people day and night. However, it will increase the burdens of family and lead to a significant escalation in home care expenditures.

To get rid of this dilemma, smart environments, a new paradigm of future living environments, have become a new promising solution [6, 7]. With huge commercial prospects and fast development of information and communication technologies, smart environments incorporate multiple cutting-edge techniques such as ambient intelligence (AmI) [8, 9], Internet of things (IoT) [10], social robotics [11] and human-computer interaction [12, 13]. They try to take full advantage of computer science to help elderly and disabled people live on their own with less nursing care. Based on the real-time analysis of indoor conditions, appropriate feedback such as intervention or assistance will be duly given out. Thus, the future living environments which rebuilt into smart homes using AmI technique will become more and more comfortable. They will offer a range of assisted services due to more consideration about context awareness, human factors and ergonomics between occupants and environments [14].

As one of the core techniques of smart environments, IoT could build a network of objects embedded with electronic components such as sensors, actuators or RFID tags to monitor indoor conditions in real-time [7, 10, 15] (see Fig. 1). These objects include home appliances, wearable devices, household furniture, components of an apartment (e.g. doors, windows) and the other daily commodities.

As a consequence, pervasive electronic components continuously generate enormous amounts of data describing environmental conditions (positions, movements, temperature and pressure, etc.) and consumption (energy or resources) [13]. Generated data constitute temporal and sequential patterns which are usually heterogeneous and chaotic. With the help of ubiquitous sensor networks, most of human behaviors performed in smart homes could trigger passive sensor events and will be captured by specific IoT components. Thus, it is possible to realize real-time activity monitoring as well as behavioral pattern analysis to avoid potential threats in their earliest stages. Moreover, for the purpose of providing appropriate assistance, smart homes should also understand an occupant's real intentions (i.e. activities willing to do) hidden behind observable behaviors as early as possible [16].

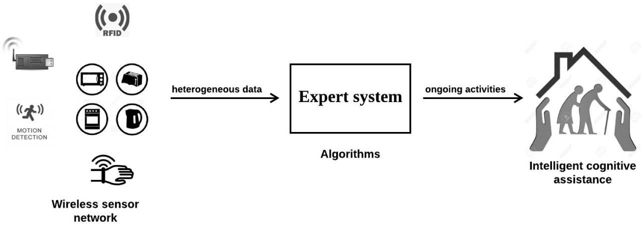

Fig. 1: Architecture about intelligent assistance based on a sensor network and an expert system in smart homes.

Because of the periodicity caused by people's habits, the majority of behavioral patterns describing an activity are regular and predictable by historical data [17]. A series of cognitive assistance can be offered based on the activity recognition and prediction given observations, namely, the captured sensor data.

The objective of behavioral pattern analysis is to recognize ongoing activities rather than classify the completed ones [18]. In other words, activity prediction is more essential to help occupants prevent dangerous situations before they occur. Once current behavioral pattern tends to be anomalous or threatening, we hope the smart homes could determine whether it is necessary to implement preventive interventions or to notice the family members, neighbors or caregivers.

However, complex behavioral pattern analysis is always a great challenge to smart home applications [19]. In most cases, human behaviors are basically planned and performed in continuous and complex ways. Compared with the behavioral patterns of independent activities, the complex ones are usually sequential without clear boundaries, sometimes even interweaved due to personal thinking ability and advanced planning such as interleaved or concurrent modes.

Besides, before a widespread commercialisation, the reliability of smart homes is another important factor [20]. As summarized in [21], a good smart home must provide reliable positioning and measurement of sensor data and have reliable algorithms for evaluating occupant's lifestyles. Considering these requirements in reliability, our main contribution of this paper is to propose a robust activity inference engine mining complex behavioral patterns in non-intrusive sensor-based smart homes. The innovative inference engine is based on formal concept analysis and could recognize ongoing activities using partially observed unreliable sensor data from a heterogeneous sensor network. Its modeling is an attempt to retrospect sequential patterns of historical sensor data to construct a reliable case-based model. It takes advantage of context-aware rules to prompt individuals for well scheduling and carrying out complex activities. Moreover, the design of inference engine uses a loose coupling architecture to strengthen the reliability in robustness and knowledge reusability.

The remainder of this paper is organized as follows. Section 2 outlines the related works about complex activity recognition in smart environments. Section 3 introduces the preliminaries of activity recognition as well as the knowledge representation and inference by formal concept analysis. Section 4 presents our experimental results. Section 5 discusses the performances and emphasis the reliability of our methods. Finally, the conclusion is reported in Section 6.

2 Related Work

Because of numerous generated data and adopted ubiquitous computing technique, data analysis, especially large-scale data mining has become an efficient solution to handle with AmI problems. Most of the raw data contain valuable information including regular patterns or useful cases. At the same time, they are usually difficult to be used directly to solve practical problems due to the lack of efficient knowledge discovery and retrieval strategies. Thus, it is essential to choose an effective representative form to index, organize and retrieve unstructured information [15].

With the help of knowledge representation and data mining techniques, useful information can be discovered from large-scale continuous data streams. On the one hand, different types of data streams need to be analyzed by suitable approaches on the basis of varied data features. On the other hand, different features are normally related to the different ways of data acquisition.

For example, the vision-based one uses facilities like cameras, webcams, kinects or social robots to capture sequences of images [22]. Each image is a set of pixel values. Vision-based solutions capture more details about environments and contexts in the data streams containing high information entropy [23]. Thus, their performances are also better than others. Those pixel values indicate the meta-data such as the real-time positions of monitored objects and as the latest states of moving objects. Image processing and pattern recognition [24, 25, 26] are common methods to handle with pixel data streams. However, because vision-based solutions directly acquire highly sensitive personal information, the trade-off between privacy and excellent performances has always aroused controversy [27]. That's why more and more applications of smart homes have adopted the non-intrusive data acquisition such as sensor-based solutions.

The sensor-based one depends on the wireless sensor network to build ubiquitous computing environments. Most of the non-intrusive components only capture the states of attached or embedded objects, react passive activation and transfer numerical, binary or categorical values through wireless communication. All the captured data contain little personal information and generate sensor data streams. Unlike homogeneous pixel values of vision-based data acquisition, it is difficult to represent and process large-scale heterogeneous sensor data by a unified data structure. In this paper, we focus on the sensor-based pattern analysis to exploit useful behavioral patterns from historical data and to recognize activities from real-time data.

Due to the difficulties of complex behavioral analysis in non-intrusive smart environments, there are only a few researches in this domain. Ruotsalainen et al. [28] introduced a genetic algorithm for detecting interleaved patterns from event sequences. The algorithm has been utilized in partitioning event sequences and matching subsequences with the specific pattern templates. Thus, this method is limited by the low performance in generalization.

Gu et al. [29] built activity models based on Emerging Patterns describing significant changes of itemset supports and differences between two classes to recognize sequential, interleaved and concurrent activities. Rashidi et al. [30] introduced an unsupervised method to discover frequent interesting activity patterns and to group similar discovered patterns. A boosted version of a hidden Markov model is created to represent and recognize the activities and their variations. The limitation of these methods is that they only considered specific subsequences occurring frequently, and ignored some important ones with imbalanced distribution in the dataset.

Interleaved hidden Markov models were introduced to recognize multitasked activities [31]. After a small modification of the classical HMM model, the improved models were able to better predict transition probabilities by recording the last object observed in each activity. Hu and Yang [32] proposed a two-level probabilistic framework for multiple-goal recognition including concurrent and interleaved activity recognition. They used skip-chain conditional random fields (SCCRF) and a correlation graph for modeling interleaved and concurrent activities. A detailed performance comparison of different techniques involving naive Bayes and the variations of hidden Markov Model was given in [33]. These methods usually have strong noise immunity. The imperfection mainly surrounds computational difficulties in the training phase. It is usually difficult to train models with a large number of parameters or large state spaces.

For the other methods, Hall et al. [34] used finite-state automaton to decompose the total power load and distinguish each individual usage of appliances. Thus, interleaved activities related to the energy consumption are indirectly discriminated. However, it could not handle with the other kind of activities without the usage of electrical devices.

Riboni et al. [35] proposed an unsupervised method to recognize complex activities by exploiting the semantics of activities, context data, and sensing devices through ontological and probabilistic reasoning. Roy et al. [36] proposed a hybrid recognition model based on the probabilistic description logic. Okeyo et al. [37] combined ontological and temporal knowledge representation to recognize composite activities. The ontological activity model established relationships between activities and involved background knowledge. The temporal one defined correlations between constituent activities of a composite activity. Saguna et al. [38] proposed a conceptual framework for spatiotemporal context-aware systems to infer interleaved and concurrent activities. However, these knowledge-based methods require more extra knowledge or predefined inference rules defined by domain experts. Their high requirement about domain knowledge makes the maintenance or extension difficult for domain freshmen.

A logic framework based on four-level lattice was defined for plan recognition in our early studies [39]. Although its complexity determines that it can not be widely used in ubiquitous computing, it also prompts the research to solve AmI issues through maturer lattice-based models. Another interesting research introduced by Ye and Dobson [40] investigated the performance of semantic data structure called context lattices to recognize activities in smart homes. Their further research proposed a knowledge-driven approach for concurrent activity recognition [41]. However, their methods largely depend on domain knowledge, predefined logic expressions and operations. These factors greatly reduce the efficiency and flexibility.

Based on the above considerations, formal concept analysis (FCA) has become our first choice to represent and extract knowledge from a large volume of unstructured data for the reason of its excellent performance [42]. It is widely used in various domains like knowledge discovery, ontology learning [43], semantic annotation [44], information retrieval and recommender system [45], to construct a knowledge graph or graphical ontology for data organization, visualization or mining. Similar to association rule learning, formal concept analysis can fast discovers useful itemsets. Nevertheless, it lacks efficient method to retrieve them. Thus, in [5, 46, 47], FCA firstly introduced to recognize singleton activity with a clear boundary from sensor event streams and to detect abnormal patterns on-the-fly. The treated streams are required to be well segmented with clear boundaries. Moreover, each sensor stream must only have one activity. These assertions are too ideal to be satisfied in reality. Thus, in this paper, we propose an improved algorithm based on FCA to recognize complex behavioral patterns from continuous sensor events.

3 Inference Engine for Activity Recognition

In this section, first of all, we introduce some notions about simple and complex activities of daily living in Section 3.1. Then, a common multilevel granularity to solve activity-centered problems of smart environments is highlighted in Section 3.2. And then, we apply the formal concept analysis theory to the AmI issues and explain the principles in Section 3.3. Next, we import an innovative search algorithm to retrieve knowledge in the FCA-based knowledge base in Section 3.4. After that, on the basis of our previous work, we demonstrate how to recognize complex activities on-the-fly by our new approach in Section 3.6. Lastly, we highlight a method called RMSD to refine the predictive activities in Section 3.7.

3.1 Preliminaries of Activity Recognition in Smart Homes

To live independently in their own homes, elderly people should at least be able to individually complete basic Activities of Daily Living (ADLs) [48]. ADLs refer to people's daily self-care activities. As a term used in healthcare, the quality of ADLs performance has been regarded as an important measurement of one person's functional status. It is also the minimum requirement for living independently.

According to the comprehensive research of Lawton et al. [49] in the behavioral assessment of elder people, ADLs could be classified by human functionality as two types:

- Basic ADLs: simple tasks about functional mobility and physical self-maintaining activities such as toilet, feeding, dressing, grooming, ambulation and bathing, etc.
- Instrumental ADLs: complex tasks that are not necessary, but more complicated than basic ADLs which need more thinking, planning and interactions. For instance, telephone, food preparation, housekeeping, laundry, responsibility for own medication, etc.

Basic ADLs are usually the short-term behaviors with short execution times. Moreover, they seem to be independent and not collaborative with each other.

Furthermore, two or more instrumental activities could generate more complex behavioral patterns. These patterns with collaborative interactions are more close to the real scenarios [19]. In the following definitions, we suppose that each mode is taken by only one occupant. Similar definitions have also been proposed in the other AmI researches [32, 50, 51].

- Sequential ADLs: activities are performed one after another in a continuous and sequential mode without interweaving execution (see the execution of behaviors a_x, b_y belonging to two ADLs a, b in Fig. 2). For example, an occupant prepared toasts after preparing a cup of coffee. Moreover, each ADL is independent and there is no shared behavior between any two successive ADLs.

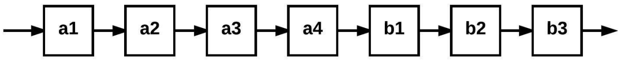

Fig. 2: Sequential mode of ADLs execution

- Interleaved ADLs: the performance of ADLs interweaves together with pauses. As shown in Fig. 3, an occupant will temporally suspend the current ongoing ADL and turn to do another one. After a while, the suspended ADL will be done sooner or later. In reality, when an ongoing ADL needs to wait for the processing or results, the occupant usually does another ADL during the waiting time (e.g. during the waiting of cooking spaghetti, the occupant turned to prepare a cup of tea). Behaviors belonging to different ADLs are frequently scheduled among the execution processes.

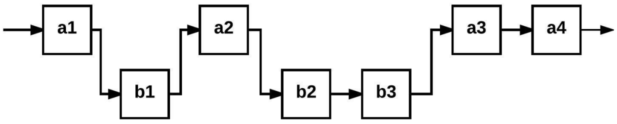

Fig. 3: Interleaved mode of ADLs execution

- Concurrent ADLs: in this mode, more than one activity is carried out simultaneously in a period of time. Although it is similar to the interleaved ones, the biggest difference is that different behaviors could be done at the same time (see Fig. 4). Because of no conflict, a person could telephone by mobile phone while cooking.

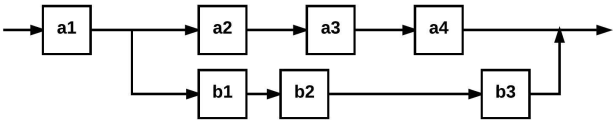

Fig. 4: Concurrent mode of ADLs execution

The issue of ADLs recognition is a very active topic in the application of ambient intelligence [13, 52, 53]. Activity tracking and recognition are also the most desirable requirement of family caregivers for smart home technology [54]. The issue refers to the long-term behavioral intention recognition from various sequential patterns which are constituted by temporal behavior-related data. Given a sequence of observations, the ADLs recognition task is to associate activities with part of observations.

For the reason of varied living habits, personal preferences or the other physical, mental and culture factors [55], an ADL could have multiple behavioral patterns to describe itself. Even if having almost the same participant actions, two patterns could be totally dissimilar because of different execution orders, recurrent or optional actions.

3.2 Multilevel Granularity in Smart Environments

Figure 6 depicts a common multilevel granularity to solve activity-centered problems of smart environments [46, 56, 57]. Three levels of granularity represent different kinds of behavioral elements. The fine-grained components are located at lower levels in the structure and the coarse-grained ones are positioned at higher levels. Each coarse-grained super component is composed of one or more fine-grained subcomponents.

The objective of ADLs recognition is to identify the most probable coarse-grained ADLs generating the observed fine-grained subcomponents. Normally, an ADL is composed of more than one observable action. In other words, several short-term targets have joined together to form a long-term intention. An action is atomic, which indicates a meaningful element describing a short-term target. Additionally, each action is measured by one or more sensor event. Thus, in the case of ADLs recognition, the coarse-grained target classes should be the high-level ADLs and the fine-grained features could be either sensor events or actions.

3.3 Knowledge Representation and Modeling

Formal concept analysis (FCA) is a mathematical theory that derives concept hierarchies from a given dataset. It aims at firstly clustering similar targets classes sharing the same ontological features, and then indexing these clusters by a structural preorder called partial order. Correlations between the individuals and features could be represented as homogeneous binary relations. In reality, binary relations describing two sets of different things widely exist in most of the scene [1]. When we use FCA to construct knowledge base, FCA requires choosing suitable target classes and corresponding features for modeling due to the restriction of binary relations.

After the brief introduction about FCA, in the following words, FCA components (including formal context, concept-forming operations, formal concept, formal lattice and Hasse diagram) will be presented as well as their roles in modeling and the constructions of the inference engine.

3.3.1 Knowledge Storage by Formal Context

In order to construct the inference engine by FCA, firstly, unstructured knowledge must be stored in a specific data structure called formal context for easier utilization and indexation in the future inferences.

A formal context $\mathcal{K}$ is the mathematical abstraction of a scene. It aims to transfer unstructured information into structured data. It could be represented as a triplet $\mathcal{K}(G,M,I)$ that consists of two disjoint sets G and M. Set I represents their Cartesian products as binary relations, which defines the different issues of ADLs recognition. The elements of G representing coarse-grained target classes are called *objects*, and the elements of M representing fine-grained features are called *attributes*. Attributes are the descriptors of objects. In order to express that object g is related with attribute m, we write gIm [58].

The triple $\mathcal{K}(G,M,I)$ could be represented and visualized as a $|G| \times |M|$ matrix. The matrix representing context $\mathcal{K}$ provides an opportunity to encode unstructured or heterogeneous information to a machine-recognizable data structure. With the help of formal context, FCA can further discover the dependencies between target classes G and feature variables M.

Application 1: In our case, both observable sensor events and atomic actions could be treated as the fine-grained features of target classes G. To facilitate the following introduction, we suppose that the fine-grained features M indicate the observed sensor events, then binary relations I define the issue as activity recognition in the stream of sensor events.

[1] For example, in linguistics, each subject establishes a binary relation with its object (or predictive) by the (linking) verb.

Activities about preparing breakfast		*a*: boil water	*b*: prepare tableware	*c*: add cocoa powder	*d*: pour cereals	*e*: take out breads	*f*: take out teabags	*g*: take out spaghetti	*h*: add sugar	*i*: add milk	*j*: add sauce	*k*: use toaster	*l*: use microwave oven
PrepareHotChocolate	g_1		×	×					×	×			×
PrepareMilkTea	g_2	×	×				×		×	×			
PrepareSpaghetti	g_3	×	×					×			×		
PrepareCaffèMocha	g_4	×	×	×					×	×			
PrepareCereals	g_5		×	×	×					×			×
PrepareToast	g_6		×			×						×	
PrepareSandwich	g_7		×			×					×		

Fig. 5: Matrix representing the activities g_i carried out in the kitchen and their atomic actions m_x.

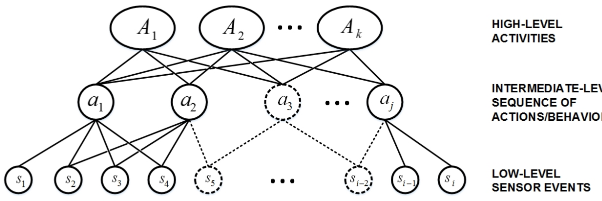

Fig. 6: Multilevel granularity in smart environments

Considering the architecture in Fig. 6, $\forall g \in G$ represents a target activity, and $\forall m \in M$ could be an indexed sensor event or atomic action. Given a concrete example, there are seven activities about preparing breakfast: *PrepareHotChocolate* (g_1), *PrepareMilkTea* (g_2), *PrepareSpaghetti* (g_3), *PrepareCaffèMocha* (g_4), *PrepareCereals* (g_5), *PrepareToast* (g_6) and *PrepareSandwich* (g_7). There are also twelve actions shared among these activities: *boil water* (a), *prepare tableware* (b), *add cocoa powder* (c), *pour cereals* (d), *take out breads* (e), *take out teabags* (f), *take out spaghetti* (g), *add sugar* (h), *add milk* (i), *add sauce* (j), *use toaster* (k) and *use microwave oven* (l). If a sensor event or action m is in the sequence describing activity g, then a cross is filled in row g and column m. For example, $\alpha = \{b \prec b \prec d \prec c \prec d \prec i \prec l\}$ is a sequence describing g_5, then five crosses are automatically filled in row g_5 (see Fig. 5).

3.3.2 Similarity Maximization by Concept-Forming Operations

In the conventional clustering methods, similarity metrics are essential to generate clusters [59]. Likewise, there also exists the metrics in FCA to cluster similar target classes sharing the same ontological features.

To exploit useful information from the FCA matrix and maximize their similarities, a pair of closure operations, so-called the *concept-forming operators* is induced to discover associations.

For a subset of objects $G_1 \subseteq G$, we define

$$G_1' := \{m \in M \mid \text{for all } g \in G_1,\ gIm\} \tag{1}$$

as an operation to find out the common features $G_1' \subseteq M$ shared by all the objects in G_1. Likewise, for $M_1 \subseteq M$, we define

$$M_1' := \{g \in G \mid \text{for all } m \in M_1,\ gIm\} \tag{2}$$

as another operation to find out all the objects $M_1' \subseteq G$ sharing the common features M_1 at the same time [58].

Their combination could maximize the dependency and similarity in a pattern itemset and make it a stable closure, which is also a cluster.

Application 2: M_1' indicates all the possible activities sharing the common features M_1. In other words, if observations M_1 are partially observed data, M_1' reveals all the possible activities given those observations.

For instance, if actions $\{bc\}$ are detected by the sensor network, according to $\{bc\}' = \{g_1g_4g_5\}$, the most possible ongoing activities are g_1, g_4 and g_5. However, itemset $\{bc, g_1g_4g_5\}$ is not a stable cluster due to $\{g_1g_4g_5\}' = \{bci\}$.

3.3.3 Cluster Representation by Formal Concept

After the definition of similarity metrics, FCA can cluster similar target classes in ontology for knowledge discovery.

A concept is a pair (G_1, M_1) where $G_1' = M_1$, $M_1' = G_1$. G_1 and M_1 are called the *extent* and the *intent* of a concept [58]. Because of $(G_1')' = (M_1)' = G_1$, each concept is a closure under the concept-forming operations. $\mathcal{B}(G,M,I)$ denotes a universe containing all the concepts of a context $\mathcal{K}$.

Concepts are the smallest FCA units having discovered knowledge. Each concept is a cluster corresponding to an

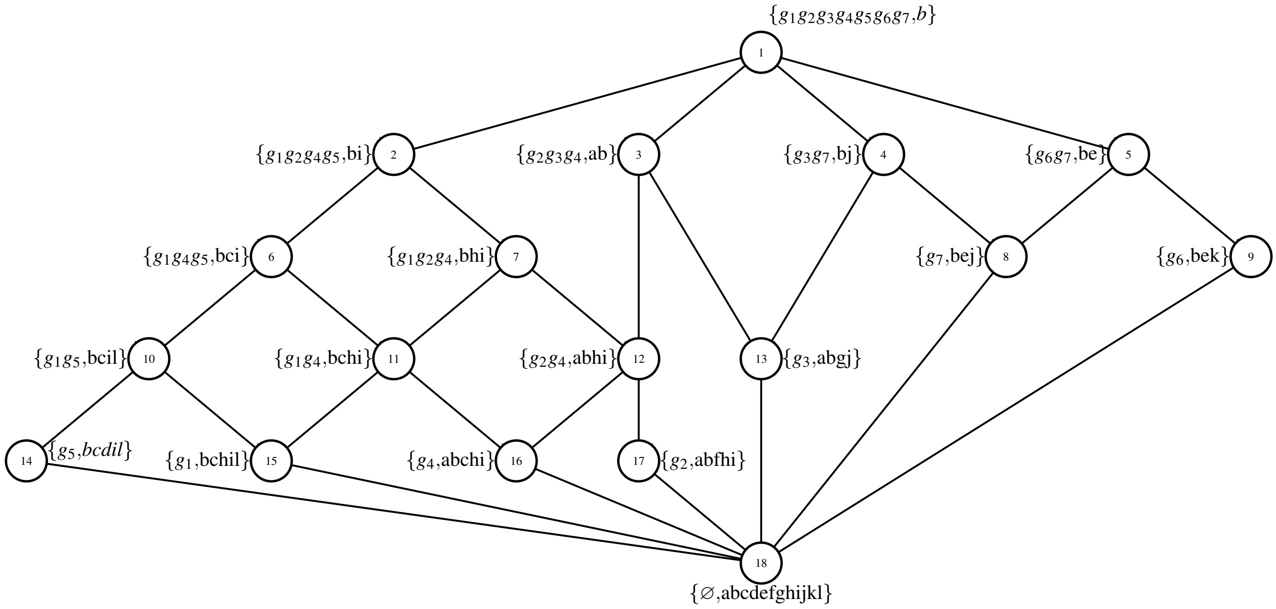

Fig. 7: Hasse diagram of Matrix in Fig. 5

inference. Its intent refers to all the observable features indicating the centroid of cluster, and the extent refers to all the similar objects sharing the intent M_1. The properties of closure ensure that the internal ontological similarity of cluster between the extent and intent is maximized.

Application 3: Each concept clusters similar activities based on the shared sensor events or actions in the intent. Furthermore, if the observed data belong to the intent, the elements in the extent indicate all the possible ongoing activities given those observations.

$$\{\underbrace{g_1g_4g_5}_{\text{predictions}}, \overbrace{bci}^{\text{observations}}\} \tag{3}$$

As shown in Equation (3), no matter in which order the data are captured by a sensor network, if actions $\{bci\}$ are observed, the most probable prediction about the ongoing activity will be chosen among g_1, g_4 and g_5.

3.3.4 Cluster Indexation by Formal Concept Lattice

As of now, similar objects were successfully clustered by different centroids (i.e. common features). Next, FCA will order all the discovered clusters by a structural preorder in order theory. The objective is to index and retrieve all those clusters efficiently for constructing a graph-based knowledge base.

A lattice $(\mathcal{B}, \preceq)$ is an ordered version of $\mathcal{B}(G,M,I)$. All the concepts in $\mathcal{B}$ are ordered by a predefined partially ordered symbol $\preceq$ indicating hierarchical relations between concepts.

Suppose (G_1,M_1) and (G_2,M_2) are two concepts of $\mathcal{B}$, (G_1,M_1) is called the *subconcept* of (G_2,M_2) if either $G_1 \subseteq G_2$ or $M_2 \subseteq M_1$, written as $(G_1,M_1) \preceq (G_2,M_2)$. The symbol $\preceq$ is named as the *hierarchical order*. Meanwhile, (G_2,M_2) is the *super concept* of (G_1,M_1). The construction of lattice is a process that discovers all the concepts of a context $\mathcal{K}$ by the concept-forming operations, and orders them by the order. It is worth pointing out that the subconcept and the super concept of a concept are not unique in $\mathcal{B}$.

Application 4: Suppose concepts, $\{g_1g_5$,bcil$\}$, $\{g_1g_4g_5$,bci$\}$ and $\{g_1g_2g_4g_5$,bi$\}$, were discovered from the matrix in Fig. 5. As shown in Equation (4), the last two concepts are the super concepts of the first one.

$$\{g_1g_5, bcil\} \preceq \{g_1g_4g_5, bci\} \preceq \{g_1g_2g_4g_5, bi\} \tag{4}$$

The relations among inferences are established by the hierarchical order. Moreover, the prediction about the most possible ongoing activities frequently changes according to different observations in different stages.

3.3.5 Knowledge Base Visualization by Hasse Diagram

In mathematics, a Hasse diagram is a graph depicting a finite partially ordered set. In our case, it is a visualization of lattice $(\mathcal{B}, \preceq)$ representing concepts as nodes (see Figure 7).

There are two special nodes in a Hasse diagram: the topmost one $\{G, \varnothing\}$ [2] named *Supremum* and the lowermost one $\{\varnothing, M\}$ named *Infimum*. Nodes are connected with edges

[2] The Supremum $\{G,b\}$ in Fig.7 is a special case because all the activities in G have interactions with action b.

named *Galois connection* which denotes the partial order $\preceq$ between nodes [58]. A Hasse diagram manages a knowledge base through its graph structure. For this reason, in order to manage knowledge information like unstructured behavioral patterns, first of all, the topic-related target classes and the feature variables are extracted from unstructured patterns. And then, the correlations between two sets of variables are maximized by concept-forming operations. After that, similar target classes are clustered by different centroids. At last, all the clusters are indexed by means of the partial order. Once the knowledge base is built, the next step is about to design efficient algorithms to incrementally retrieve specific behavioral patterns from the FCA-based graphical knowledge base.

Application 5: As can be seen from Fig. 7, concepts are organized by different levels. From a higher level to a lower one, the amount of observation increases and the scope of possible activities shrinks. Using successively observed data to incrementally infer ongoing activity is the core idea of the inference engine.

3.4 Inference Process

In the following words, we present an innovative search algorithm to retrieve knowledge in the knowledge base.

3.4.1 Principle of Real-time Knowledge Inferences

As summarized earlier, all the inferences about the possible ongoing activities given partial observations are encapsulated in the nodes of the hierarchical lattice. Thus, as an expert system for behavioral pattern recognition, the FCA-based solution is to use efficient algorithms to retrieve useful patterns in the graphical knowledge base using sequential observations.

Most of the graph search algorithms must traverse the whole graph to locate targets every time. When new observations are captured, they could not recall or continue from the last interrupted position and have to start over again for searching updated α. For these reasons, we propose a new graph search algorithm to locate target concepts incrementally in the FCA-based models. On one hand, the new algorithm improves the search efficiency. On the other hand, it also maintains the consistency of reasoning.

Suppose that a *sequence* α is an ordered list of observed data, denoted as $\alpha = \{\alpha_1 \prec \alpha_2 \prec ... \prec \alpha_m\}$, where $\alpha_j \prec \alpha_{j+1}$ means α_j occurs before α_{j+1}. Besides, we define a *token* to locate the local optimal concept having the smallest intent in size containing the observations in α. The reason behind such location is to avoid the loss of predictions.

The transition of token in the diagram is the incremental inference process caused by the extension of observed data. In a Hasse diagram, higher-level concepts have smaller intents due to less observed data. Thus, the topmost concept containing α is the local optimal one. This hypothesis is described in Equation (5) as follows:

$$\forall M_1 \supseteq \alpha, T_b \subseteq M_1 \Longrightarrow (G_1, M_1) \preceq (T_A, T_b) \text{ and } G_1 \subseteq T_A \quad (5)$$

where T_b is the smallest superset of α and owned by the topmost concept (T_A, T_b). For example, in Figure 7, if $\alpha = \{b \prec i\}$, then node 2 $\{g_1g_2g_4g_5, bi\}$ is the topmost concept we are looking for. If we choose another one (e.g. node 6 $\{g_1g_4g_5, bci\}$) instead of the topmost one, several predictions will be missing (e.g. $\{g_1g_2g_4g_5\} - \{g_1g_4g_5\} = \{g_2\}$) and could never be found in the following searches.

3.4.2 Knowledge Retrieval

Breadth-first search (BFS) is one of the most common graph search algorithms. The main idea is to explore all the neighbor nodes in the same level before moving to the next level. Because of successively extended α, in our case, BFS is more efficient than another one called depth-first search (DFS) and chosen as the base of our new algorithm.

We propose a new half-duplex graph search algorithm (HDGS) to locate each local optimal concept for successively extended α. As can be seen from the name, HDGS consists of two directions search. Firstly, the *top-down search* locates the first discovered concept containing α. And then, starting from this concept, the *bottom up search* turns back along the hierarchical order and looks for the topmost concept containing α. More details about the HDGS algorithm are sketched in Algorithm 1 and 2.

Algorithm 1: Top-down search of HDGS algorithm

```
Data: start position sp, sequence $\alpha$.
Result: first met super concept containing $\alpha$.
begin
    fifo ← node[sp]
    while fifo do
        if fifo[0] not visited then
            mark as visited
            if $\alpha \subseteq$ fifo[0].intent then
                return fifo[0]
            else
                add fifo[0].successors into fifo
        remove fifo[0] from fifo
    end
end
```

In Pseudo-code (2), we need to pay attention to Line (12) which seeks the topmost concept having the minimal cardinality of intent containing α. Due to Equation (5), the local optimal concept has smaller cardinality than the other ones.

Algorithm 2: Bottom up search of HDGS algorithm

```
Data: start position $sp$, set $\alpha$.
Result: topmost super concept containing $\alpha$.
begin
    $fifo \leftarrow node[sp].predecessors$
    $S \leftarrow \varnothing$
    while $fifo$ do
        if $fifo[0]$ not visited then
            mark as visited
            if $\alpha \subseteq fifo[0].intent$ then
                add $fifo[0].predecessors$ into $fifo$
                $S \leftarrow S \cup fifo[0]$
        remove $fifo[0]$ from $fifo$
    end
    return $\arg\min_{s \in S}(\mid s.intents \mid)$
end
```

Table 1: Example of simple ADLs recognition

Round	Observed Data α	Located Topmost Concept	Predictive Activities
1	$\{b\}$	node 1 $\{g_1g_2g_3g_4g_5g_6g_7,\mathbf{b}\}$	$g_1g_2g_3g_4g_5g_6g_7$
2	$\{bc\}$	node 1 $\curvearrowright$ node 6 $\{g_1g_4g_5,\mathbf{bc}\mathrm{i}\}$	$g_1g_4g_5$
3	$\{bcb\}$	node 6 $\{g_1g_4g_5,\mathbf{bc}\mathrm{i}\}$	$g_1g_4g_5$
4	$\{bcbh\}$	node 6 $\curvearrowright$ node 11 $\{g_1g_4,\mathbf{bch}\mathrm{i}\}$	g_1g_4
5	$\{bcbhi\}$	node 11 $\{g_1g_4,\mathbf{bchi}\}$	g_1g_4
6	$\{bcbhih\}$	node 11 $\{g_1g_4,\mathbf{bchi}\}$	g_1g_4
7	$\{bcbhihl\}$	node 11 $\curvearrowright$ node 15 $\{g_1,\mathbf{bchil}\}$	g_1

3.5 Simple ADLs Recognition

The simple ADLs recognition is a process continuing to locate local optimal concepts given successively extended α. For instance, $\alpha = \{b \prec c \prec b \prec h \prec i \prec h \prec l\}$, Table. 1 is the illustration of recognition process. The symbol $\curvearrowright$ indicates the transition of token.

Beyond the issue of locating local optimal concepts, we also need to consider another tough issue about multilevel inheritance. It is a very common situation in AmI context due to diverse lifestyles and personal habits. Besides the flexible execution orders, an activity could also be accomplished by alternative ways like adding or omitting optional events [5, 56]. For instance, *PrepareCoffee*(g_4) and another three derived activities about preparing coffee: *PrepareBlackCoffee*(g_1), *PrepareCoffeeWithoutMilk*(g_2) and *PrepareCoffeeWithoutSugar*(g_3), have the multilevel inheritance relations as $g_1 \subset g_2, g_3 \subset g_4$.

Therefore, the retrieval strategy that we adopted is based on the *greedy* manner. That is, if an activity is recognized, its completeness will also be verified until all the necessary events in the intent have been done. If it belongs to one of the inherited activities, we continue adding observed data into α until token t shifts to the Infimum $\{\varnothing, M\}$, which means all probable activities are recognized in the previous extensions.

3.6 Complex ADLs Recognition

Compared with the single ADLs recognition, the complex one refers to distinguish mixed patterns belonging to different activities.

As mentioned, each node of Hasse diagram is a cluster regrouping ontological-similar objects that shared common features. For any activity, no matter how varied its execution orders are among sensor events, the set of corresponding patterns are normally in the same node in the Hasse diagram due to the FCA-clustering. Furthermore, a behavioral pattern could derive many inherited patterns representing as adjoining nodes by adding or refining optional sensor events. Thus, a node representing similar activities and its leaf nodes representing the inherited nodes define a larger group of clusters having the similarly ontological relations among internal activities.

If an incoming data is excluded by such a cluster, it means that the data has strong ontological differences with internal activities. As a result, the incoming data is classified as an outlier of the current plan which is being executed and has to be put into another one. The new one starts a new search with a new token.

The principle of deciding whether an observed data is necessary to be excluded or not by the current plan is determined by the diagram. Suppose that a node (G_1, M_1) is located by the token, the set of relevant data R_e given a target class g is obtained by Equation (6).

$$R_e = \bigcup_{\forall g \in G_1} g' \tag{6}$$

where g' is the concept-forming operation shown in Equation (1). All the other data, no matter indexed or not by the lattice, will be classified as the outliers of the current plan because the Infimum is immediately located by the token. Once an outlier is detected, a provisional boundary will be marked and a new plan for caching will be created at the same time. The search of the current plan will also rollback from the Infimum to the previous position.

Suppose $\alpha = \{b \prec e \prec b \prec c \prec i \prec b \prec l \prec g \prec k \prec h\}$ indicating interleaved activities *PrepareHotChocolate* (g_1) and *PrepareToast* (g_6). There is also an unreliable data g (*take out spaghetti*). Table 2 depicts the complex ADLs recognition process. The symbol $\circlearrowleft_{Infimum}$ indicates the rollback operation from the Infimum.

Table 2: Example of complex ADLs recognition

Round	Observed Data α	Located Topmost Concept	Predictive Activities
1	$\{b\}$	node 1 $\{g_1g_2g_3g_4g_5g_6g_7,\mathbf{b}\}$	$g_1g_2g_3g_4g_5g_6g_7$
2	$\{be\}$	node 1 $\curvearrowright$ node 5 $\{g_6g_7,\mathbf{be}\}$	g_6g_7
3	$\{beb\}$	node 5 $\{g_6g_7,\mathbf{be}\}$	g_6g_7
4	$\{bebc\}$	node 5 $\circlearrowright_{Infimum}$ $\{g_6g_7,\mathbf{be}\}$ node 6 $\{g_1g_4g_5,\mathbf{bci}\}$	g_6g_7 $g_1g_4g_5$
5	$\{bebci\}$	node 5 $\circlearrowright_{Infimum}$ $\{g_6g_7,\mathbf{be}\}$ node 6 $\{g_1g_4g_5,\mathbf{bci}\}$	g_6g_7 $g_1g_4g_5$
6	$\{bebcib\}$	node 5 $\{g_6g_7,\mathbf{be}\}$ node 6 $\{g_1g_4g_5,\mathbf{bci}\}$	g_6g_7 $g_1g_4g_5$
7	$\{bebcibl\}$	node 5 $\circlearrowright_{Infimum}$ $\{g_6g_7,\mathbf{be}\}$ node 6 $\curvearrowright$ node 10 $\{g_1g_5,\mathbf{bcil}\}$	g_6g_7 g_1g_5
8	$\{bebciblg\}$	node 5 $\circlearrowright_{Infimum}$ $\{g_6g_7,\mathbf{be}\}$ node 10 $\circlearrowright_{Infimum}$ $\{g_1g_5,\mathbf{bcil}\}$ node 13 $\{g_3,\text{a}\mathbf{bgj}\}$	g_6g_7 g_1g_5 g_3
9	$\{bebciblgk\}$	node 5 $\curvearrowright$ node 9 $\{g_6,\mathbf{bek}\}$ node 10 $\circlearrowright_{Infimum}$ $\{g_1g_5,\mathbf{bcil}\}$ node 13 $\circlearrowright_{Infimum}$ $\{g_3,\text{a}\mathbf{bgj}\}$	g_6 g_1g_5 g_3
10	$\{bebciblgkh\}$	node 9 $\{g_6,\mathbf{bek}\}$ node 10 $\curvearrowright$ node 15 $\{g_1,\mathbf{bcilh}\}$ node 13 $\circlearrowright_{Infimum}$ $\{g_3,\text{a}\mathbf{bgj}\}$	g_6 g_1 ~~g_3~~

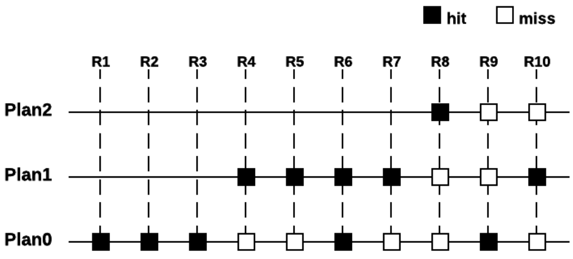

Fig. 8: Interweaving plans

At round 4, when c is observed, $\{bebc\}$ is excluded by the current plan because no subconcept of node 5 contains these observations except the Infimum. Thus, a new plan is created to cache c and launches a concurrent search. At round 8, because g is excluded by all the existing plans. A new concurrent one is created at that moment to cache g. Activities g_6 and g_1 are finally recognized at round 9 and 10, because their sizes of extent are equals to 1 and all the required observations in the intents are observed.

Figure 8 illustrates the interweaving situation. There are totally three plans P_i ($i \in \{0,1,2\}$) in the figure. P_0 is the initial plan. P_1 and P_2 are created when observed data is irrelevant to all the existing plan. Squares indicate two states of observed data: the black ones indicate the observed data is relevant to the patterns in the present P_i (i.e. hit), and the hollow ones indicate the data is irrelevant (i.e. miss). For an incoming data, it could trigger three possible states:

- strictly belongs to one plan: the observed data belongs to a unique plan. For example, $R1$, $R2$, $R3$, $R4$, $R5$, $R7$, $R8$, $R9$ and $R10$ in Fig. 8.
- belongs to more than one plan: it always happens to concurrent ADLs. For example, $R6$ in Fig. 8.
- belongs to none of the existing plans: In sequential ADLs, it is the moment triggering the boundary detection. In interleaved ADLs, the occupant may start to do another activity or an irrelevant action, or the system may receive an unreliable data. For example, $R4$ and $R8$ in Fig. 8.

At the end of the data stream, a completeness check will verify all the existing plans. There are two objectives: first of all, the amount of predictive activities will be checked. The plan having too many predictive activities will be abandoned due to ambiguity. Otherwise, a further check will verify the completeness of each activity calculated by Equation 7.

$$C_i = \frac{|g_i' \cap \alpha|}{|g_i'|} \quad \text{and} \quad g_i \in G \qquad (7)$$

where $|g_i' \cap \alpha|$ indicates the amount of observed data and $|g_i'|$ indicates the required one. An activity having low completeness will be abandoned. In Table 2, activity g_3 was finally abandoned due to low completeness, and the cached data g was identified as an unreliable data.

3.7 Inference Assessment

Because of few observed data, a local optimal concept usually has more than one candidate in its extent, which means that there will be more than one prediction. Without an efficient assessment, redundant predictions will be ambiguous and useless to make decisions for real-time assistance. In this case, we desire to evaluate the relevance of each prediction in a local optimal concept and choose the most relevant one as the *local optimal prediction*.

As mentioned in the previous sections, an ADL g_i could be accomplished by alternative patterns because of different personal preferences. Furthermore, these derived patterns reflect on flexible execution orders, repetitive events, adding or omitting optional events, etc. At the same time, everyone also has his own relatively stable preference to execute an ADL. Namely, for the same occupant executing an ADL, there are only a few deviations among each execution. Based on this hypothesis, we take advantage of historical patterns containing the preferences of occupants to generate a knowledge database called *accumulated matrix*. For each event, we calculate its expectant position appearing in each ADL to establish a series of naive distributions.

To measure the context similarities between historical patterns and current incomplete ongoing ADL, average deviations are calculated using Root-Mean-Square Deviation (RMSD). It makes a quantitative comparison as an assessment to estimate how well the current sequence fits historical data. A lower RMSD score indicates that a prediction is more accurate due to excellent fitting with historical patterns.

We propose our assessment as follows: for each candidate g_i in the extent, under the condition of executing g_i, we calculate the deviation between actual average positions in α and the accumulated ones in the matrix. Thus, the local optimal prediction should be the one with minimal deviation which has the best fitting in comparison with historical data. Obviously, our assessment consists of two modules: accumulation and evaluation.

3.7.1 Accumulation

For each event α_j in a training item α, a complete sequence of events of ADL g_i (i.e. $\alpha_j \in \alpha$, $\alpha \in g_i$), we update the accumulated value of corresponding element (g_i, α_j) in the accumulated matrix by Equation (8):

$$\sigma_{ij} = \sigma'_{ij} + j \tag{8}$$

where j is the position of α_j in α. σ'_{ij} is the previous accumulated value and σ_{ij} is the newly updated one. Equation (9) represents the same accumulation in another global view:

$$\sigma_{ij} = \sum_{k=1}^{N_{ij}} \sigma_{(ij,k)} \tag{9}$$

where N_{ij} represents the occurrences of element (g_i, α_j) existing in the whole training dataset. $\sigma_{(ij,k)}$ is the position of α_j in the k-th training item describing ADL g_i.

3.7.2 Evaluation

When an incoming event α_j was observed, first of all, we calculate its average position $\overline{\varphi_j}$ in current sequence α. It is calculated by Equation (10).

$$\overline{\varphi_j} = \frac{1}{\#\alpha_j} \sum_{k=1}^{|\alpha|} k, \quad \text{if } \alpha_k = \alpha_j \tag{10}$$

where $|\alpha|$ is the size of current sequence α, and $\#\alpha_j$ is the occurrences of α_j in α. The condition $\alpha_k = \alpha_j$ is necessary to integrate $\#\alpha_j$ discrete positions of α_j.

And then, for each candidate g_i, we calculate the deviation of α given g_i. Equation (11) expresses the root-mean-square deviation D_i of current sequence α executing g_i:

$$D_i = \sqrt{\frac{1}{|\alpha|} \sum_{\forall \alpha_j \in \alpha} (\overline{\varphi_j} - \frac{1}{N_{ij}} \sigma_{ij})^2} \tag{11}$$

where σ_{ij}/N_{ij} is the expectant position obtained from accumulated matrix.

Thus, RMSD scores $\{D_1, D_2, ..., D_i\}$ of candidates in the current extent $G_1 = \{g_1, g_2, ..., g_i\}$ were calculated. The element g_i having the minimal RMSD value is the local optimal prediction because of the best fitting with historical patterns.

4 Experiments

The performances of inference engine are tested using two datasets created in two smart environments, LIARA smart home and CASAS testbed. The common classification metrics, F-measure and accuracy [60, 61], are used to evaluate the performance of activity recognition. All the experiments are carried out on the computer with tech specs of Intel Core i7 Processor 2.4GHz and 8GB RAM, under Ubuntu 16.04.

4.1 LIARA Smart Home

LIARA smart home is a prototype of future living environments which concentrates the innovation solutions surrounding smart environments based on the ambient intelligence to provide real-time cognitive assistance for the disabled or elderly people. It consists of various sensors, actuators, passive RFID tags, tablets and wearable devices to monitor indoor environmental changes caused by human behaviors in a non-intrusive way.

Figure. 9 shows the prototypical design of LIARA smart home. Objects in the figure are embedded with measurable electrical components. For instance, infrared, light sensors and RFID antennas have been installed on the walls. The oven in the kitchen zone is monitored and controlled by a built-in micro computer and temperature sensors. A tablet is also embedded on the refrigerator. The water consumption

Fig. 9: Sensor layout of the LIARA smart home.

is measured by water sensors, and the power consumption is recorded by a power analyzer located at the main electrical panel. The open and close states of cabinets could be detected by binary sensors. Pressure mats are placed in the bathroom to trace occupants' movements. Besides, passive RFID tags are attached on all the other daily commodities to track their spatial positions.

Using the infrastructure above, we created a dataset in order to recognize complex activities given atomic actions. The sequences of actions are generated from our previous research: the passive RFID signal parsing [62] can identify the usage of items and the electrical devices identification [63] can identify the usage of appliances. To create training and test data, first of all, one participant performed each activity twenty times, and then, he was allowed to freely perform activities in sequential, interweaving or parallel ways. Due to complicated scenarios and much more interactions with appliances and food ingredients, we choose instrumental kitchen activities as our research objects. Twelve activities are described by sequentially observed actions (see Table 4). As shown in Table. 3, LIARA data has three important fields indicating the date and time of the triggering moment: *timestamps*, recognized *atomic actions* and the corresponding *activity*.

In order to test the performance under large scale data, based on the real data, we generated numerous synthetic data having more varied patterns. Furthermore, datasets were randomly divided into *10* subsets, and such division was repeated *10* times. Each time, one subset was chosen as the test set and the other *9* subsets were put together to form a training set. This approach is called *10-fold cross-validation*.

The objective of 10-fold cross-validation is to evaluate the capacity about generalization, a well-known issue in machine learning [64]. Sometimes, a model could receive excellent evaluations on the data existing in the training set. However, once the test data has not been seen before, the classification result may break down. As an effective approach, cross-validation could indicate the performance of built model when it is asked to make a prediction on the data that is not used to create the model. With its help, each pattern in the dataset was removed at least once from the training sets.

Table 3: Data structure of LIARA training items

Timestamps	Atomic Actions	Activities
1	BoilWater	PrepareCoffee
2	TakeCupFromCupboard	PrepareCoffee
3	TakeOutCoffeePowder	PrepareCoffee
4	PutCoffeePowderIntoCup	PrepareCoffee
5	StoreCoffeePowder	PrepareCoffee
6	PourWaterIntoCup	PrepareCoffee
7	TakeOutSugar	PrepareCoffee
8	AddSugarIntoCup	PrepareCoffee
9	StoreSugar	PrepareCoffee
10	TakeOutMilkFromRefrigerator	PrepareCoffee
11	PourMilkIntoCup	PrepareCoffee
12	StoreMilkInRefrigerator	PrepareCoffee
13	BrewCoffee	PrepareCoffee
14	PutSpoonIntoSink	PrepareCoffee

Table 4: Statistical information and F-measure results of LIARA dataset

Activities	NO. Actions	F-measure
PrepareSandwichWithoutMustard	11	0.947
PrepareCoffeeWithoutSugar	11	0.947
PrepareCereals	8	1.000
PrepareMilkTea	12	1.000
PreparePudding	5	1.000
PrepareToastsEggs	20	1.000
PrepareMilk	5	0.952
PrepareSandwichWithoutButter	9	0.869
PrepareSpaghetti	18	1.000
PrepareCoffee	14	0.976
PrepareSandwich	15	0.902
PrepareCoffeeWithoutMilk	11	0.806
Overall F1 score	-	0.954
Overall accuracy	-	0.985

In Table. 4, statistical information and F-measure results using FCA-based inference engine are given out. Activities without multilevel inheritance relations have better recognition accuracies in the complex mode. This is because activities with multilevel inheritance relations are easier to be affected by unreliable data and recognized as one of their similar derivations.

4.2 CASAS Smart Apartment

CASAS smart apartment is designed and constructed by the Center for Advanced Studies in Adaptive Systems of Washington State University. Its benchmark datasets[3] represent sensor events collected in a smart apartment testbed [19]. In the experiment, we compare our results of *Kyoto-3* in-

[3] available at http://ailab.wsu.edu/casas/datasets/

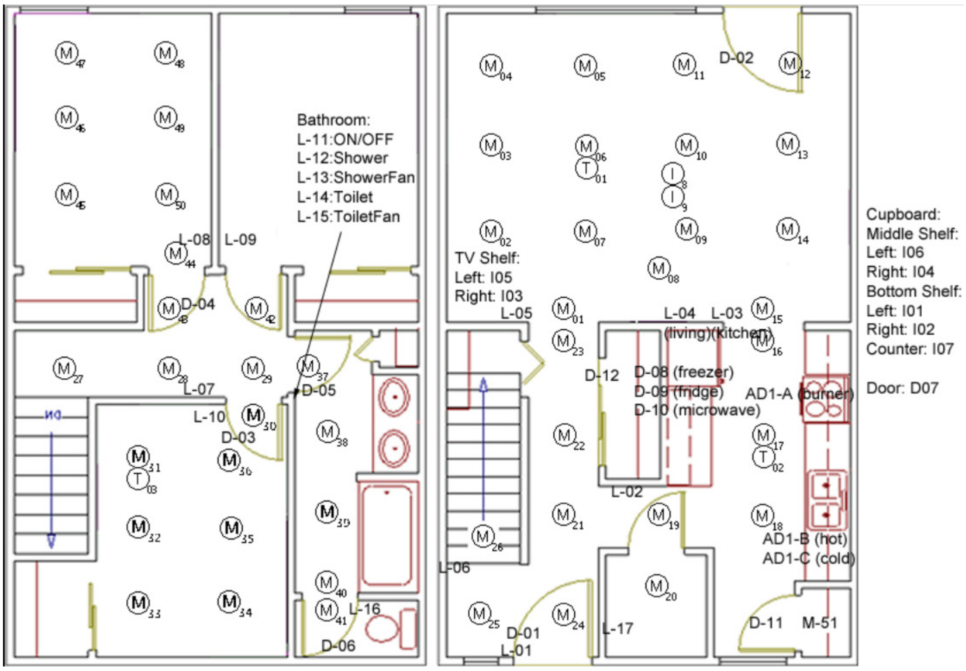

Fig. 10: Sensor layout of the CASAS intelligent apartment.

Table 5: Comparison of accuracies of CASAS dataset

Classes	Naive Bayes [33]	HMM [19]	FCA-based
ac_1	50%	58%	100%
ac_2	62%	78%	100%
ac_3	27%	43%	60%
ac_4	39%	46%	95%
ac_5	78%	80%	95%
ac_6	83%	82%	100%
ac_7	89%	81%	100%
ac_8	57%	67%	100%

terweaved ADL activities dataset with the other methods in the literature. The CASAS dataset contains the patterns of sequential and interleaved activities. As shown in Fig. 10, the whole apartment, including the bedrooms, a bathroom, a kitchen and a living room, is deployed full of heterogeneous sensors to capture various environmental states.

Instead of using passive RFID tags, the CASAS laboratory directly uses motion sensors to track an occupant' scopes of activities. The data in the sequences are the raw sensor events. Besides, the CASAS smart apartment also includes temperature sensors, light controllers and a variety of item sensors detecting the usages and interactions of attached objects by individuals. Moreover, customer-built analog sensors monitor the usage of hot water, cold water and stove burner. The phone usage is captured by Asterisk software and the states of doors and cabinets are captured by contact switch sensors. Pressure sensors monitor the usage of key items such as medicine container, cooking pot and phone book [30].

There are twenty participants performing eight basic and instrumental activities in the apartment. First of all, each activity was performed separately, and then these participants are asked to perform the entire set of eight activities again in any order, interweaving and performing tasks in parallel if desired. Eight activities were involved: *fill medication dispenser* (ac_1), *watch DVD* (ac_2), *water plants* (ac_3), *answer the phone* (ac_4), *prepare birthday card* (ac_5), *prepare soup* (ac_6), *clean* (ac_7), and *choose outfit* (ac_8). Each sensor reading is tagged with the timestamps, sensor id and its value.

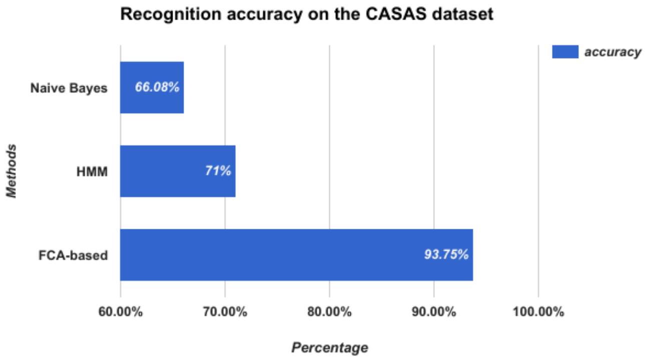

Fig. 11: Recognition accuracy of different methods on the CASAS Kyoto-3 dataset

In Table. 5, we compared the recognition accuracy with different methods [19, 33]. In Fig. 11, our method achieves

Table 6: Comparison of F-measure of CASAS dataset

Classes	HMM [19]	MLN (supervised) [35]	FCA-based
ac_1	0.656	0.803	**1.000**
ac_2	0.862	0.882	**1.000**
ac_3	0.285	0.740	**0.750**
ac_4	0.589	0.688	**0.973**
ac_5	0.828	0.807	**0.974**
ac_6	0.826	0.873	**1.000**
ac_7	0.881	0.781	**1.000**
ac_8	0.673	0.904	**1.000**
avg	0.700	0.810	**0.962**

the highest accuracy (93.75%) among naive Bayes (66.08%) and HMM (71%) [33]. In Table. 6, we compared the performance of our method with another two methods described in [19, 35] by F-measure. Complex behavioral patterns are classified as eight classes (activities). From these comparisons, we could see that our method outperforms in each recognition case.

For the time complexities in both training and test phases, we give out the statistical information in Table. 7. The training phase includes sequential pattern extraction, formal lattice construction and historical data accumulation. While handling with LIARA dataset, we could see that the training and testing times are both very low. Compared with LIARA dataset, CASAS data has much fewer training items, but the training time is much longer than the LIARA one. The reason is that the number of target classes greatly affect the number of clusters. The augmentation of clusters also increases the complexity of searching in the Hasse diagram. For CASAS dataset, every activity was performed by different actors and each activity having the same name must distribute different class labels as the variations of the original activity.

5 Discussion

In this section, we introduce the advantages and disadvantages of our innovative methods from the obtained experimental results and some discussions about reliability.

5.1 Preprocessing

The construction process of formal lattice depends on the correlations between objects and attributes. Normally, each data with categorical or binary value should be treated as an element in the set of attributes. However, sensor events with discrete values could not be well represented by binary relations. To avoid high dimensionality, discrete sensor values should be transformed into categorical ones by disjoint intervals.

Another optional processing is about the feature selection. In the training phase, if some captured data are not relevant to the belonging activities, or some attributes do not enable to efficiently differentiate activities, they should be pruned from the matrix describing correlations.

5.2 Reliability

The discussion about reliability surrounds in several parts. The first one is about the privacy. In the layout of LIARA smart home, the types of sensors used in the infrastructure are the non-intrusive ones, which do not contain private personal information. Unlike vision-based or video-based methods dealing with sequential and continuous images containing a large number of private data, FCA-based inference engine only deals with discrete categorical or numerical values describing the states of environments. Thus, there is no controversy over privacy.

The second one is the low volume requirement about the training data. FCA-based methods do not need a large amount of data for the construction of formal lattice. It could automatically self-adapt various patterns from small indexed patterns. Redundant patterns will not bring significant improvement in accuracy, in some cases, it will lead overfitting and the useless features have to be pruned. Besides, Both LIARA and CASAS datasets adopted the intended strategy to label the experimental training data [65]. In the intended strategy, participants followed predefined instructions to do various activities. Observers could also clarify in case of uncertainty. For the test dataset, both of them let the participant use another keyhole strategy to generate complex behavioral patterns. In the keyhole strategy, observers do not influence participants behaviors and manually annotate the complex sequences.

The third one is about the reusability. The inference engine contains three key components: the formal lattice to index useful itemsets, the accumulated matrix to record historical preferences and weighted arrays as well as matrices to evaluate causality and threat levels of events [5]. Most of them are independent, or loosely coupled with each other. For example, the matrices and arrays are user independent. The formal lattice could be reusable to provide various behavioral patterns of an activity. Furthermore, the accumulated matrix could be replaced without modifying the architecture of the inference engine.

The fourth one is about the flexibility. As the common prerequisite of knowledge-driven methods [35], domain experts are essential to define and model knowledge base including the domain of interest, relations among individuals, classes and properties. In other words, the interventions from domain experts generally run through entire knowledge-based designs.

Table 7: Performance of FCA-based algorithm in different datasets

Datasets	No. Classes	No. Features	No. Nodes	No. Training Items	Training Times	No. Test Items	Test Times
LIARA	12	70	25	25207	0.0062s	2520	0.8093s
CASAS	160	84	5089	160	40.3625s	20	1.6961s

In contrast, the FCA-based inference engine depends on few prerequisite knowledge. From the FCA construction to the final recognition, there is no special requirement of understanding complicated rules or domain knowledge.

The last one is about the robustness. In software engineering, the robustness refers to the ability of a system to handle exceptions or erroneous inputs during execution. For a machine learning algorithm, it refers to the stable performance while handling datasets with noise. As mentioned in [35], when different patterns are partially captured or an occupant exhibited strange behaviors, the performance will normally fluctuate.

Our FCA-based method not only possesses good generation capacity, but also is sensitive to detect various abnormal behavioral patterns. For each unseen pattern which is not in the training dataset, but appears in the test dataset, FCA-based inference engine will analyze its patterns and the similarity with existing cases, and then propose the target class with the most similar patterns. According to the most common abnormal behaviors observed among people suffering cognitive impairment, in [5], we formally defined six cognitive errors and proposed a series of customer-built error detectors. These errors involve the majority of abnormal situations caused by the unreliable data. If the data loss happens due to less reliable sensors, most activities could be normally recognized and identified as unfinished ones. Even in the worse cases, FCA-based methods could evaluate partially observed data, calculate the similarity and classify the instance into the most possible activity cluster. As shown in the examples in Tables 1 and 2, repetitive data and few redundant ones have almost no affection to the final recognition results.

6 Conclusion

In this paper, we introduce a new activity inference engine to analyze sequential and temporal behavioral patterns in order to recognize complex ongoing ADLs in smart environments equipped with non-intrusive sensor networks. Complex ADLs involve the activities performed in the concurrent, sequential or interleaved ways.

Compared with conventional supervised data mining methods, our engine has strong generalization capacity and some reliable characteristics. The results tested in two different datasets demonstrate the excellent performance in behavioral pattern analysis.

In our future research, we intend to concentrate on the activity recognition among multiple occupants. In addition, more complex evaluation in the contextual and temporal parts will replace current RMSD assessment. A series of optimizations about formal lattice construction are also considered in the plan. For example, the incremental updating of new training data, weighted formal context instead of the binary one and a more efficient pruning mechanism.

References

1. Department of Economic and Social Affairs. *World Population Ageing Report*. United Nations, 2015.
2. National Institute on Aging. *Why Population Aging Matters: A Global Perspective*. NIH Publication, 2016.
3. World Health Organization. *Global Health and Aging*. NIH Publication, 2015.
4. United Nations. *Alzheimer's Disease Fact Sheet*. National Institute on Aging, NIH Publication, 2016.
5. Jianguo Hao, Sébastien Gaboury, and Bruno Bouchard. Cognitive errors detection: Mining behavioral data stream of people with cognitive impairment. In *Pervasive Technologies Related to Assistive Environments*, pages 15:1–15:8. ACM, 2016.
6. Diane J Cook and Sajal K Das. How smart are our environments, an updated look at the state of the art. *Pervasive and mobile computing*, 3(2):53–73, 2007.
7. Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2013.
8. Fariba Sadri. Ambient intelligence: A survey. *ACM Computing Surveys (CSUR)*, 43(4):36, 2011.
9. Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013.
10. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.
11. Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. Assessing acceptance of assistive social agent technology by older adults: the almere model. *International journal of social robotics*, 2(4):361–375, 2010.

12. Paolo Remagnino and Gian Luca Foresti. Ambient intelligence: A new multidisciplinary paradigm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 35(1):1–6, 2005.
13. Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009.
14. Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.
15. Liyanage C De Silva, Chamin Morikawa, and Iskandar M Petra. State of the art of smart homes. *Engineering Applications of Artificial Intelligence*, 25(7):1313–1321, 2012.
16. Saisakul Chernbumroong, Shuang Cang, Anthony Atkins, and Hongnian Yu. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, 40(5):1662–1674, 2013.
17. G Michael Youngblood, Diane J Cook, and Lawrence B Holder. Managing adaptive versatile environments. *Pervasive and Mobile Computing*, 1(4):373–403, 2005.
18. MS Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *International Conference on Computer Vision*, pages 1036–1043, 2011.
19. Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. Tracking activities in complex settings using smart environment technologies. *International journal of biosciences, psychiatry, and technology (IJB-SPT)*, 1(1):25, 2009.
20. Charlie Wilson, Tom Hargreaves, and Richard Hauxwell-Baldwin. Smart homes and their users: a systematic analysis and key challenges. *Personal and Ubiquitous Computing*, 19(2):463–476, 2015.
21. Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes - present state and future challenges. *Computer methods and programs in biomedicine*, 91(1):55–81, 2008.
22. Alexandros Andre Chaaraoui, José Ramón Padilla-López, Francisco Javier Ferrández-Pastor, Mario Nieto-Hidalgo, and Francisco Flórez-Revuelta. A vision-based system for intelligent monitoring: Human behaviour analysis and privacy by context. *Sensors*, 14(5):8895–8925, 2014.
23. Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.
24. Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.
25. Bishop and Christopher M. *Pattern recognition and machine learning*. Springer, 2006.
26. Mohamad Al-Wattar, Rinat Khusainov, Djamel Azzi, and John Chiverton. Activity recognition from video data using spatial and temporal features. In *International Conference on Intelligent Environments*, pages 250–253, 2016.
27. Andrea Cavallaro. Privacy in video surveillance [in the spotlight]. *IEEE Signal Processing Magazine*, 2(24):168–166, 2007.
28. Marja Ruotsalainen, Timo Ala-Kleemola, and Ari Visa. Gais: A method for detecting interleaved sequential patterns from imperfect data. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 530–534. IEEE, 2007.
29. Tao Gu, Zhanqing Wu, Xianping Tao, and H. K. Pung. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *IEEE International Conference on Pervasive Computing and Communications*, pages 1–9, 2009.
30. Parisa Rashidi, Diane J Cook, Lawrence B Holder, and Maureen Schmitter-Edgecombe. Discovering activities to recognize and track in a smart environment. *IEEE transactions on knowledge and data engineering*, 23(4):527–539, 2011.
31. Joseph Modayil, Tongxin Bai, and Henry Kautz. Improving the recognition of interleaved activities. In *UBICOMP 2008: Ubiquitous Computing, International Conference, UBICOMP 2008, Seoul, Korea, September 21-24, 2008, Proceedings*, pages 40–43, 2008.
32. Derek Hao Hu and Qiang Yang. Cigar: concurrent and interleaving goal and activity recognition. In *AAAI Conference on Artificial Intelligence*, pages 1363–1368, 2008.
33. Geetika Singla and Diane J. Cook. Interleaved activity recognition for smart home residents. pages 145–152, 2009.
34. Sylvain Hallé, Sébastien Gaboury, and Bruno Bouchard. Towards user activity recognition through energy usage analysis and complex event processing. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 3. ACM, 2016.
35. Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1–12. ACM, 2016.
36. Patrice Roy, Bruno Bouchard, Abdenour Bouzouane, and Sylvain Giroux. A hybrid plan recognition

model for Alzheimer's patients: interleaved-erroneous dilemma. *Web Intelligence and Agent Systems: An International Journal*, 7(4):375–397, 2009.

37. George Okeyo, Liming Chen, and Hui Wang. Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*, 39:29–43, 2014.
38. Saguna, Arkady Zaslavsky, and Dipanjan Chakraborty. Towards a robust concurrent and interleaved activity recognition of mobile users. In *IEEE International Conference on Mobile Data Management*, pages 297–298, 2011.
39. Bruno Bouchard, Sylvain Giroux, and Abdenour Bouzouane. A keyhole plan recognition model for alzheimer's patients: first results. *Applied Artificial Intelligence*, 21(7):623–658, 2007.
40. Juan Ye and Simon Dobson. Exploring semantics in activity recognition using context lattices. *Journal of Ambient Intelligence and Smart Environments*, 2(4):389–407, 2010.
41. Juan Ye, Graeme Stevenson, and Simon Dobson. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47–70, 2015.
42. Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Mimmo Parente. Time aware knowledge extraction for microblog summarization on twitter. *Information Fusion*, 28:60–74, 2016.
43. Vincenzo Loia, Giuseppe Fenza, Carmen De Maio, and Saverio Salerno. Hybrid methodologies to foster ontology-based knowledge management platform. In *Intelligent Agent (IA), 2013 IEEE Symposium on*, pages 36–43. IEEE, 2013.
44. Carmen De Maio, Giuseppe Fenza, Mariacristina Gallo, Vincenzo Loia, and Sabrina Senatore. Formal and relational concept analysis for fuzzy-based automatic semantic annotation. *Applied intelligence*, 40(1):154–177, 2014.
45. Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore. Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. *Information Processing & Management*, 48(3):399–418, 2012.
46. Jianguo Hao, Abdenour Bouzouane, Bruno Bouchard, and Sébastien Gaboury. Real-time activity prediction and recognition in smart homes by formal concept analysis. In *12th International Conference on Intelligent Environments*, pages 103–110. IEEE, 2016.
47. Jianguo Hao, Abdenour Bouzouane, and Sébastien Gaboury. Activity inference engine for real-time cognitive assistance in smart environments. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–20, 2017.
48. Barry Reisberg, Sanford Finkel, John Overall, Norbert Schmidt-Gollas, Siegfried Kanowski, Hartmut Lehfeld, Franz Hulla, Steven G Sclan, Hans-Ulrich Wilms, Kurt Heininger, et al. The Alzheimer's disease activities of daily living international scale (ADL-IS). *International Psychogeriatrics*, 13(2):163–181, 2001.
49. MP Lawton and Elmne M Brody. Assessment of older people: Self-maintaining and instrumental activities of daily living. *Nursing Research*, 19(3):278, 1970.
50. Xiaoyong Chai and Qiang Yang. Multiple-goal recognition from low-level signals. In *National Conference on Artificial Intelligence*, pages 3–8, 2005.
51. Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Multioccupant activity recognition in pervasive smart home environments. *Acm Computing Surveys*, 48(3):1–36, 2015.
52. Eunju Kim, Sumi Helal, and Diane Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9(1):48–53, 2010.
53. Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.
54. Vincent Rialle, Catherine Ollivet, Carole Guigui, and Christian Hervé. What do family caregivers of Alzheimer's disease patients desire in smart home technologies? - Contrasted results of a wide survey. *Methods of Information in Medicine*, 47(1):63–69, 2008.
55. Robert E Wray and John E Laird. Variability in human behavior modeling for military simulations. 2010.
56. Tim LM Van Kasteren, Gwenn Englebienne, and Ben JA Kröse. Hierarchical activity recognition using automatically clustered actions. In *International Joint Conference on Ambient Intelligence*, pages 82–91. Springer, 2011.
57. Been-Chian Chien and Rong-Sing Huang. Activity recognition using discriminant sequence patterns in smart environments. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 329–341. Springer, 2014.
58. Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 1999.
59. Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. Chapman and Hall/CRC, 2013.
60. Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
61. Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.
62. Dany Fortin-Simard, Jean-Sébastien Bilodeau, Kevin Bouchard, Sébastien Gaboury, Bruno Bouchard, and Abdenour Bouzouane. Exploiting passive RFID technology for activity recognition in smart homes. *IEEE*

Intelligent Systems, 30(4):7–15, 2015.

63. Corinne Belley, Sébastien Gaboury, Bruno Bouchard, and Abdenour Bouzouane. Nonintrusive system for assistance and guidance in smart homes based on electrical devices identification. *Expert Systems with Applications*, 42(19):6552–6577, 2015.
64. Charu C Aggarwal. *Data classification: algorithms and applications*. CRC Press, 2014.
65. Patrice C Roy, Bruno Bouchard, Abdenour Bouzouane, and Sylvain Giroux. Ambient activity recognition in smart environments for cognitive assistance. *International Journal of Robotics Applications and Technologies (IJRAT)*, 1(1):29–56, 2013.