Recognizing Multi-Resident Activities in Non-intrusive Sensor-Based Smart Homes by Formal Concept Analysis

Jianguo Hao*, Abdenour Bouzouane, Sébastien Gaboury

LIARA Laboratory, Université du Québec à Chicoutimi
555 Boulevard de l'Université, Chicoutimi, G7H 2B1, Canada

Abstract

Activity recognition is one of the most important prerequisites for smart home applications. It is a challenging topic due to the high requirements for reliable data acquisition and efficient data analysis. Besides, the heterogeneous layouts of smart homes, the number of residents and varied human behavioral patterns also aggravate the complexity of recognition. Therefore, most human activity recognition systems are based on an unrealistic assumption that there is only one resident performing activities. In this paper, we investigate the issue of multi-resident activity recognition and propose a knowledge-driven solution on the basis of formal concept analysis (FCA) to identify human activities from non-intrusive sensor data. We extract the ontological correlations among sequential behavioral patterns. At the same time, these correlations are well organized in a graphical knowledge base, without intervention from domain experts. We propose an incremental lattice search strategy in order to retrieve the best inference given a few sensor events. Compared with other conventional probabilistic methods, our solution outperforms on the CASAS multi-resident benchmark dataset. Furthermore, we open up a promising solution of sequential pattern mining to discover the ontological features of temporal and sequential sensor data.

Keywords: Multi-Resident Activity Recognition, Formal Concept Analysis, Sequential Pattern Mining, Smart Homes, Ambient Intelligence

*Corresponding author

Email addresses: `jianguo.hao1@uqac.ca` (Jianguo Hao*), `abdenour.bouzouane@uqac.ca` (Abdenour Bouzouane), `sebastien.gaboury@uqac.ca` (Sébastien Gaboury)

1. Introduction

According to recent reports of United Nations [1, 2], the huge improvement in health and standard of living, the longevity as well as low birth rates have greatly affected the population proportion of elderly people. As a result, our world is entering into an aging society and the population aging has actually affected our lives in many ways [3]. Elderly people usually require more home care because of their aging-related physical or cognitive impairment. The progressive deterioration in cognitive skills makes them difficult to live independently. The expenditure for long-term caregivers gradually increases the financial burden on families [2]. However, most of them are willing to live as long as possible in their own homes [4, 5]. As a promising solution for the elder-care applications, smart homes are designated to provide adaptive and personalized service for the residents of the home environments [6].

With the help of ambient intelligent (AmI), Internet of things (IoT) and wireless sensor networks (WSNs) technologies [7, 8], the development of smart homes has been greatly promoted. First, AmI highlights the importance of recognizing ambient, contextual and situational information in smart homes. It advocates the use of pervasive computing [9] and context-aware [10] technologies for human-centric interactions. Second, WSNs and IoT integrate various devices and sensors into a unified network to facilitate the real-time exchange of information [11]. In addition, they make small, low-cost and non-intrusive sensors with low-power consumption widely deployed in home automation. In the future, sensing platforms will also intelligently reduce the data storage and processing requirements, such as remote configuration [12]. Thus, smart environments can be used for security monitoring, home automation assistance, situational awareness, energy saving and other fields.

As one of the most important prerequisites for ambient assisted living applications, identifying ongoing human activities is essential for the design and development of customized context-aware systems in order to provide appropriate assistance at the right time [13]. The systems should accumulate data for behavioral analysis to understand what activities are being carried out and the subsequent essential behaviors required to complete these activities. Finally, they should choose appropriate feedback to benefit the residents in the smart environment [14]. However, recognizing human activities in a sensor-based smart home is a difficult and challenging task for the scientific community because of unreliable data, varied behavioral patterns, and interweaving human-object interactions.

Moreover, the complexity of activity recognition increases when there are multiple residents in a smart environment [15]. Multiple inference rules must be applied to the same sensors at the same time in the same place. In fact, most living environments have more than one resident. For example, family members get together to prepare dinner, or to do housework at the same time. Multi-resident activities can be carried out in an individual, parallel or cooperative manner. Because of the social characteristics of human beings, activities can be coordinated by multiple residents. In these cases, each sensor reading may involve more than one resident.

Compared with the single-resident activity recognition, recognizing activities in the multi-resident scenario is equally important. Older people usually live with other family members like their spouse and children. However, these family members are not always available to provide timely help. Based on this assumption, ambient living assistance to monitor the multi-resident activities is still necessary, because it will greatly reduce the burden of family care. Moreover, due to obvious differences in behavioral patterns, the inferences of single-resident activity recognition cannot be directly applied to the multi-resident one.

In our previous studies, we first proposed the prototype of our inference system [16] and then successfully recognized the complex activities performed in a concurrent, sequential or interleaved way [17]. We also showed its great potential in anomaly detection [18]. As an extension, in this paper, we focus on the issue of recognizing multi-resident activities in a non-intrusive sensor-based smart home. The main contributions of the paper are:

- Further extend and improve the application of Formal Concept Analysis (FCA) in multi-resident activity recognition.

- Propose a promising sequential pattern mining solution to discover the ontological features of temporal and sequential data.

- Represent and maintain domain knowledge through the concept lattice of FCA, a graphical knowledge base that is independent of specific sensor layouts and activities to recognize.

- Retrieve the best inferences of multi-resident activities in the graphical knowledge base through a new lattice search strategy.

The remainder of this article is organized as follows. Section 2 surveys the recent research progress about multi-resident activity recognition in non-intrusive sensor-based smart environments. Section 3 presents the preliminaries about recognizing multi-resident activities in smart homes. Section 4 introduces the basics of formal concept analysis, and the knowledge-driven model based on this theory. Section 5 shows the results under different measurement criteria and compares them with other classic models. The discussions are investigated in the same section. We conclude in Section 6.

2. Related Work

According to the different ways of data collection, the infrastructure designs of a smart home can be divided into two types: intrusive and non-intrusive ones [19]. However, in this article, the related work and the solutions proposed later only discuss how to analyze and recognize multi-resident activities in non-intrusive sensor-based smart homes.

For the intrusive design, many vision-based devices such as RGB cameras, depth cameras, or vision-based social robots are deployed in the living environment to capture real-time dynamic information in smart homes [20, 21, 22, 23, 24]. The captured information is saved in the form of an image sequence, and each image is a set of homogeneous pixel values. Although vision-based solutions provide more detailed and rich context information about human behaviors [25], they also collect more private and sensitive information at the same time. Moreover, the characteristic of large-scale pixel values determine that the captured image sequences have to be processed by big data techniques, such as image processing, machine learning and pattern recognition [26]. Their high computational costs are also a major drawback.

For the non-intrusive one, besides various home appliances providing identifiable electrical load signature [27], WSNs establish an object network consisting of electronic components such as sensors, actuators or RFID tags to monitor environmental conditions. These components are usually embedded or attached to household furniture, doors, windows and other daily necessities. In addition, in recent studies, more and more wearable and mobile devices have been used to collect basic human motions or vital signs [23, 28, 29]. Fig. 11 depicts the general architecture of a typical sensor-based smart home. Heterogeneous non-intrusive sensors are deployed throughout an apartment to monitor and capture human behaviors, as well as the usage of home appliances.

In the literature, different solutions are proposed to solve the problem of multi-resident activity recognition based on the sensor-based infrastructure design. They can be categorized as data-driven and knowledge-driven models. However, both of them regard graphical models as the first choice to describe the association among activities and to provide a dynamic description of state transitions.

2.1. Data-driven Models

Compared with knowledge-driven models, data-driven ones place more emphasis on using large-scale data to drive internal reasoning [30]. Some mainstream solutions are the models based on the statistical and probabilistic theories, such as hidden Markov models (HMMs), conditional random fields (CRFs) and their variants. They identify all relevant variables in the smart environment and build dynamic probabilistic models that take into account the regularity of probability distribution and the state transition probabilities.

2.1.1. Probabilistic and Statistical Models for Classification

Using historical behaviors and profiles of residents, Crandall and Cook [31] combine an HMM with a Naive Bayesian Classifier (NBC) to identify residents. The system maps sensor events to the residents who triggered them, and then predicts residents' desires and further interacts with them. In [32], authors present a Bayesian network-based probabilistic generative framework to characterize the structural variabilities of complex activities.

Chiang et al. [33] adopt two graphical models, parallel HMM (PHMM) and coupled HMM (CHMM), to identify activities in a multi-resident environment. Besides, they also propose a new dynamic Bayesian network extending CHMM. To model activity patterns, domain knowledge has been added and sensor data has been categorized in the preprocessing. Benmansour et al. [34] develop an HMM-based combined label (CL-HMM) and a linked HMM (LHMM) to compare their performances against the PHMM and CHMM methods. Besides, Wang et al. [35] study a temporal probabilistic model called Factorial Conditional Random Field (FCRF) to model interacting processes in a sensor-based, multi-user scenario.

In [36], Chiang et al. propose a feature-based knowledge transfer framework to extract and transfer knowledge between two different smart environments. They first use a PCA-like method to reformulate input feature sets, and then measure the divergence among the features by Jensen-Shannon divergence. After that, a graph matching algorithm is used to derive the best feature mapping between training and testing datasets. Liu et al. [37] propose another two-stage approach to firstly cluster the training data by K-means using temporal features like start time, end time and approximate duration, and secondly to recognize the activities in each cluster.

In fact, reliable transition probabilities and emission matrices depend on large amounts of training data having stable probability distributions. The probabilities should be calculated from a dataset which probability distributions are quite close to the reality. Generally, data-driven models stress on discovering probabilistic or statistical regular patterns over training data. Thus, reliable probability distributions and statistical stability are the most important factors for the final results. However, small-scale training data could not ensure the distributions of training data are infinitely close to the reality. As a consequence, results of probabilistic models will be sensitive to unbalanced distributions.

2.1.2. Models using Association Rules

Chen and Tong explore a two-stage activity recognition method in [38]. It is an extension of the typical HMM and CRF. It uses association rules to learn combined training sequences at first stage, and then maps test sequences to multi-resident activities at the second stage.

Prossegger and Bouchachia [39] propose an application of incremental decision trees to classify activities in a multi-resident context. Their model allows leaf nodes to be multi-labeled for representing single or multiple classes and incrementally accommodates new instances as well as new activities.

As reported by Hsu et al. [40], CRFs are applied with different strategies to preprocess the dataset. Their work investigates the importance of data association in multi-resident activity recognition. At the same time, it also emphasizes the significance of considering the noise in the dataset and the prior knowledge about the environment in the preprocessing.

2.1.3. Deep Learning

Fang and Hu [41] propose a deep learning algorithm to recognize human activities. They adopt the deep belief networks (DBNs) built by restricted Boltzmann machine in the research. They also compare their results with HMM and NBC. Applying artificial neural network (ANN), Oniga and Suto [42] analyze the signals acquired from acceleration sensors. Zhang et al. [43] combine HMM and DNN models to recognize activities.

Moreover, for a part of methods like deep learning algorithm, there is no efficient mechanism to organize discovered knowledge. As black-boxes, if the results are not good in some cases, it is hard to explain the reasons and find out the solutions.

For the data-driven approaches, they try to use mathematical theories to establish probabilistic or statistical models based on the analysis of historical data. However, due to the sensitivity of noisy data, they typically have high requirements for data quality and volume to generate a stable and reusable model. Data scarcity may cause underfitting. Additional operations, such as data cleansing, may be applied before processing. Moreover, most of them have insufficient extensibility. If new training data greatly affects the probability distribution or statistical stability of previous training dataset, the entire model needs to be retrained.

2.2. Knowledge-Driven Models

Compared with data-driven approaches, knowledge-driven models are easier to be understood and interpreted by researchers and domain experts in knowledge representation. Their classification results are also easier to explain. When their performance is unsatisfactory, it is easier to find the reason for optimization. Instead of retraining models to find the regular patterns by probability and statistical theories, knowledge-driven models can be easily extended by adding homogeneous new domain knowledge.

Ye and Stevenson present a knowledge-driven approach combining ontologies with semantic matching techniques to recognize daily human activities [13]. The proposed approach works well for the activities having explicit semantics, but it is limited in distinguishing the ones having ambiguous semantic features. Their successive research [44] continues to recognize multi-user concurrent activities from an unsegmented continuous sensor sequence. Combining ontological reasoning with statistical methods, the boundaries of different activities are automatically detected by dividing a continuous sensor sequence into partitions.

Alam et al. [45] investigate the challenges of improving the recognition of complex activities in multi-resident smart homes. They propose a loosely-coupled hierarchical dynamic Bayesian network to identify coarse-grained activities using fine-grained atomic actions and sensor data. Because of the prohibitive computation, they have to discover the key spatio-temporal constraints in the activity contexts across users and learned association rules on the basis of Apriori algorithm to prune the state space of the Bayesian network. However, the context correlations and constraints among activities cannot be generated automatically. These constraints well defined the conflicts for extra and inter-user activities in spatial and temporal correlations.

Explicit semantics are essential for most of knowledge-driven models. The models usually depend on prior knowledge defined by domain experts or an open ontology to infer results. Thus, their maintenance and extension are difficult for the persons who are not familiar with specific domain knowledge. Moreover, their customization usually requires significant artificial costs. Sometimes, they can distinguish activities with great semantic gaps among sensor events, but cannot well recognize two concurrent activities with similar semantic features [44].

3. Preliminaries about Non-intrusive Smart Homes

In this section, we introduce the preliminaries related to the issues of sensor-based multi-resident activity recognition. They include the different features of data streams, multi-level data granularity and the formal definition of multi-resident activity recognition.

3.1. Data Features

As an important factor of data mining, data features are crucial for efficient data analysis [46]. It directly determines which methods are more suitable for preprocessing, feature selection and knowledge discovery [47]. Unlike the vision-based solutions which generate derived features from pixel values of original image to detect desired portions or shapes [48, 49, 50], the sensor-based ones focus more on the natural characteristics of sensor events, such as spatio-temporal features, probabilistic and contextual information.

As a result of continuous samplings and information exchange, ubiquitous electronic components constantly generate a large number of data to describe environmental changes in smart homes. The changes involve human motions, environmental conditions [51] (including locations, movements and temperature, etc), consumption [27] (energy or resources) and human-to-environment interactions [35].

There are several distinct features of the data generated in smart homes. First of all, large volumes of data provide an opportunity to discover rules and correlations from seemingly chaotic data through mathematical, especially probability and statistical theories. Then, each captured data has a timestamp that records the exact moment when a sensor event was captured or triggered. Next, based on the timestamps, all the data captured in a time interval can be sorted into a sequence. The order of the captured data always contains rich contextual relations, as some specific data is often restricted by several causal constraints. In addition, because of the heterogeneous sensors, their data types can be discrete, continuous, nominal (categorical), binary, ordinal or numeric [47]. Finally, people can carry out activities one after another in reality. Therefore, the data is usually continuous and there is no clear boundary to determine the beginning and end of an activity. Moreover, multi-resident activities are more difficult to recognize, because the data belonging to different residents can be recorded at the same time.

3.2. Data Granularity in Smart Environments

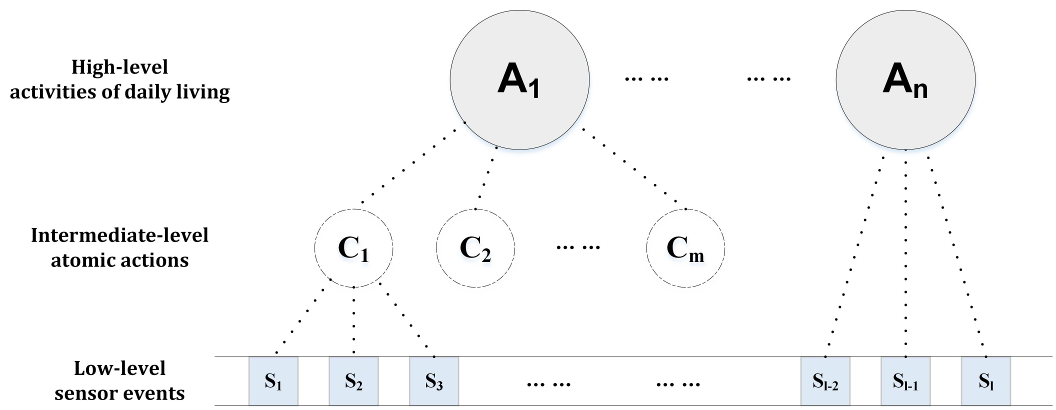

Figure 1: Data granularity in smart homes

The data produced by sensor-based smart homes can be divided into three layers of granularity. Each layer represents a type of behavioral element. There are two primary many-to-many mappings in the sensor-based activity recognition. The first one is from low-level sensor events to high-level activities (e.g. $S_l \Rightarrow A_n$). The second one uses intermediate-level atomic actions to recognize high-level activities (e.g. $C_m \Rightarrow A_n$). Fig. 1 represents these mappings. Atomic actions are the smallest human behaviors that cannot be further subdivided, such as walking, sitting, standing up, taking, bringing, opening, closing, stirring, etc. They either describe the instantaneous state of human behavior, or the current interaction with other objects. Fine-grained elements are located at relatively lower layers (e.g. S_l or C_m for A_n). Each coarse-grained element consists of one or more fine-grained elements. For example, an activity "prepare dinner (A_1)" consists of several atomic actions, such as "take out something from a refrigerator (C_1)" and "preheat an oven (C_2)", in other words, $C_1, C_2 \Rightarrow A_1$. In addition, both C_1 and C_2 can also be detected and represented by one or more sensor events (S_l), written as $S_l \Rightarrow C_1, C_2$.

3.3. Multiple Resident Activity Recognition

Before exploring and recognizing activities performed by multiple residents, we should first analyze their regular patterns. The multiple resident activities can be jointly performed in two manners: parallel and cooperative.

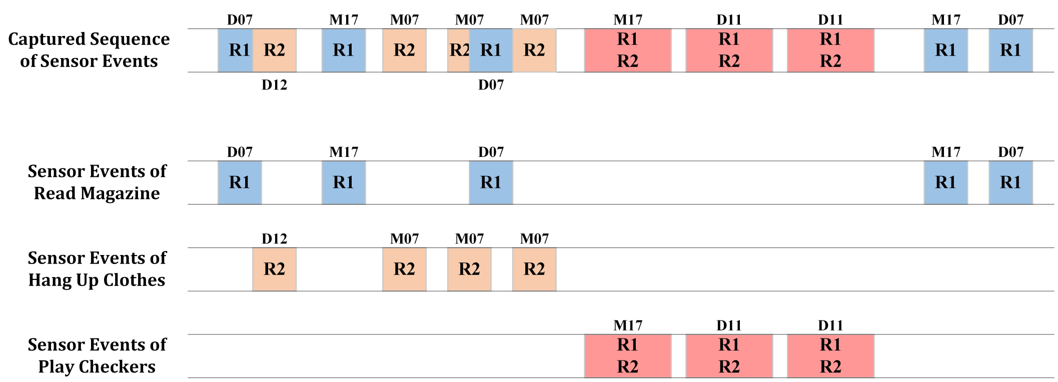

Figure 2: Regular behavioral patterns of multi-resident activities in smart homes

Fig. 2 illustrates the different patterns of multi-resident activities that occur in a smart home. The transverse axis is a time axis in order to illustrate the sequential relations between the captured pattern and the original ones. Ideally, the topmost pattern is captured and observed by the smart home. It contains multiple behavioral patterns describing parallel or cooperative activities. It should be decomposed as the three following patterns classified by activities. The squares of different colors represent different activities of different residents (e.g. two residents R1 and R2). A sensor event may be triggered by either a resident (e.g. D07 was triggered by R1 and D12 triggered by R2.), or multiple ones (e.g. M17 was triggered by both R1 and R2).

In the parallel manner, different activities are carried out independently by different residents at the same time. Two or more behavioral patterns are independent of each other. Since there is no causal constraint between different activities, their behavioral data will be interweaved. In addition, almost all sensor events are triggered by only one resident (see the patterns of reading magazine and hanging up clothes in Fig. 2). For cooperative activities, an activity is carried out in cooperation by a group of people [51, 52]. When an activity is either interactive or cooperative, a resident can ask others to participate in order to achieve a common goal. Due to the interaction and cooperation, most sensor events are triggered by multiple residents at the same time, it is difficult to determine exactly who triggered which sensor event (see the pattern of playing checkers in Fig. 2).

4. Formal Concept Analysis

Formal concept analysis (FCA) is a field of applied mathematics based on the concept and conceptual hierarchy. It activates the mathematical thinking for conceptual data analysis and knowledge discovery [53]. With its help, heterogeneous correlations existing between target classes of interest and observable features can be unified as homogeneous binary relations. FCA firstly clusters similar target classes sharing the same ontological features, then encapsulates them as inferences, and finally orders them for fast retrieval. Given the observations of different stages, our FCA-based model can make incremental inferences about possible ongoing activities done by different residents.

Because of the excellent performance of knowledge representation and extraction in large volumes of unstructured data [54], FCA is widely used in the fields like knowledge discovery, ontology learning [55], information retrieval and recommender system [56] to extract useful information and to construct a knowledge graph for data organization and visualization [57].

4.1. Knowledge Management by FCA

In this subsection, we describe the components of FCA and their roles in knowledge representation and management. At the same time, we also explain how to use FCA theory to solve the problem of multi-resident activity recognition. Fig. 4 outlines the overview process of modeling and recognition. After observing a sequence of sensor events, the modeling (or training) phase is marked by a red trajectory, and the recognition phase is marked by the black one.

Simplified CASAS Activities [51]		m_1: D07	m_2: D11	m_3: D12	m_4: D13	m_5: D14	m_6: I04	m_7: M04	m_8: M06	m_9: M07	m_{10}: M09	m_{11}: M13	m_{12}: M17	m_{13}: M23
Fill medication dispenser	g_1	×					×						×	
Hang up clothes	g_2			×						×	×			×
Move furniture	g_3							×						×
Read magazine	g_4								×	×	×			
Water plants	g_5								×	×				
Sweep floor	g_6			×					×	×	×	×	×	×
Play checkers	g_7		×										×	
Prepare dinner	g_8										×	×		
Set table	g_9			×							×			×
Read magazine	g_{10}	×											×	
Pay bills	g_{11}					×							×	
Pack picnic food	g_{12}								×	×				
Pack picnic food	$g_{12'}$								×	×				
Retrieve dishes	g_{13}				×				×	×	×	×		
Retrieve dishes	$g_{13'}$								×	×	×	×		
Retrieve dishes	$g_{13''}$								×	×	×	×		
Pack picnic supplies	g_{14}	×									×		×	
Pack and bring supplies	g_{15}			×		×				×	×		×	×

Figure 3: Matrix representing the correlations between activities g_i and sensors m_j.

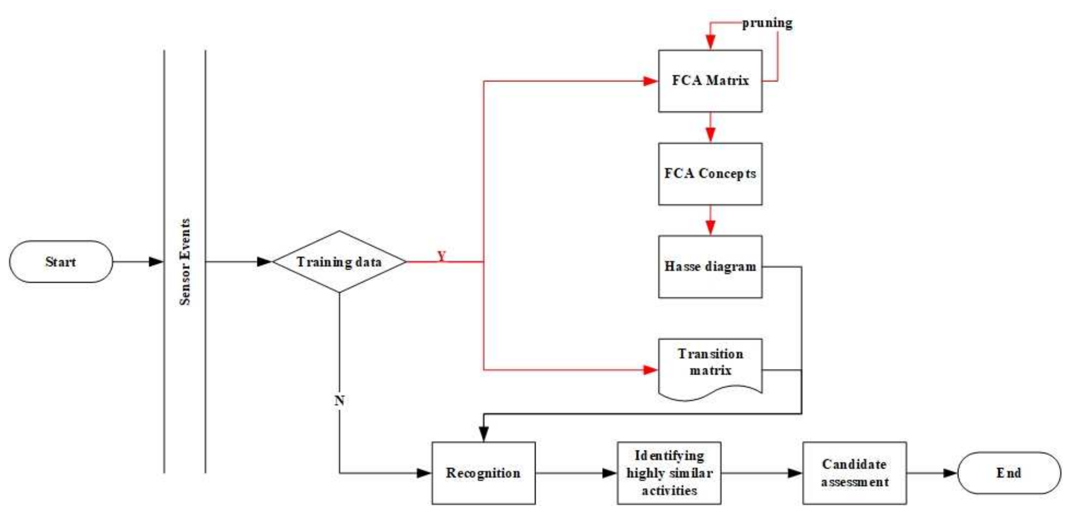

Figure 4: Overview process of FCA modeling and activity recognition

Fig. 5 outlines the procedure concerning FCA-based modeling, from knowledge extraction to visualization. First, the correlations between activities and sensor events are abstracted into a formal context. Second, an optional feature selection is executed to prune redundant and irrelevant features. Third, knowledge is discovered by two concept-forming operations and encapsulated as inferences. Finally, a graphical knowledge base is constructed by sorting and indexing those inferences and can be visualized through a Hasse diagram.

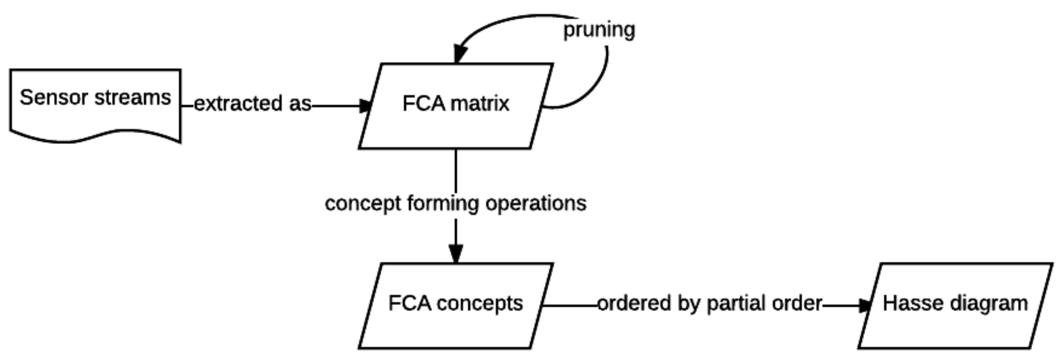

Figure 5: Procedure of Knowledge Representation and Management by FCA

4.1.1. Knowledge Storage by Formal Context

Due to various data features and the multi-level granularity, unifying and managing heterogeneous data is a thorny issue. To solve this problem and then analyze the temporal and sequential patterns, first of all, the correlations between target classes and observable features are mapped to a specific data structure named formal context.

Formal context $\mathbb{K}(G, M, I)$ is a mathematical abstraction of reality scenes. The triplet structure $\mathbb{K}$ consists of two disjoint sets G and M, and their Cartesian product set I. The elements of G are called *objects*, which represent coarse-grained target classes. The ones of M are called *attributes*, which represent fine-grained observable features. If $g \in G$ is correlated with $m \in M$, the correlation can be written as gIm [53]. Context $\mathbb{K}$ can be represented and visualized by a $|G| \times |M|$ matrix. With the help of formal context, FCA can further discovery the dependencies between G and M.

Application 1. In the case of activity recognition, we treat the activities to recognize as target classes G and observable sensor events as feature variables

M. For instance, we introduce a simplified version of a benchmark dataset called CASAS [51] as a concrete example. Sequential and temporal data is extracted and stored in a 17×13 binary matrix (see Fig. 3). It is worth mentioning that g_{13}, $g_{13'}$ and $g_{13''}$ are three different patterns to accomplish the same activity "Retrieve dishes". In particular, with the same sensor events, $g_{13''}$ is redundant for $g_{13'}$, and will be reduced later. There are 13 non-intrusive sensors that passively capture human behaviors in a smart apartment. To extract correlations, if a sensor event m_j appears in the data stream describing an activity g_i, it means $g_i I m_j$, and the element at row g_i and column m_j of the matrix will be filled by a cross.

4.1.2. Similarity Maximization by Concept-Forming Operations

In data mining, especially data clustering technique, similarity metrics are essential to generate clusters [58]. To exploit useful information from an FCA matrix and cluster similar target classes sharing the same feature variables, FCA defines its own metrics to maximize similarity. Items in the same cluster have high similarity because they share some of the same ontological features.

A pair of closure operations, so-called the *concept-forming operators*, is introduced. As mentioned above, the target classes are the activities to recognize, and the features are the observable sensor events captured by various non-intrusive sensors.

For a subset $G_1 \subseteq G$, 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. Conversely, 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 [53].

Using the two operators at the same time, FCA could generate stable closures, named class-feature pairs, to cluster correlated classes and features for maximizing their dependency and the similarity. In our case, the operator (1) can find out the common sensor events shared by a set of activities, and the operator (2) can reveal which activities have the given set of observations (sensor events).

Application 2. For instance, if sensor events $\{m_3, m_{10}\}$ are observed, according to $\{m_3 m_{10}\}' = \{g_2 g_6 g_9 g_{15}\}$, the most possible ongoing activities are g_2, g_6, g_9 and g_{15}. However, such a class-feature pair is not stable due to $\{g_2 g_6 g_9 g_{15}\}' = \{m_3 m_{10} m_{13}\}$. The stable one is $\{g_2 g_6 g_9 g_{15}, m_3 m_{10} m_{13}\}$, called a formal concept.

4.1.3. Cluster Representation by Formal Concept

To infer ongoing activities from given observable features, FCA uses the discovered knowledge encapsulated in those class-feature pairs. Moreover, to ensure the reliability of inferences, FCA only uses the stable pairs that simultaneously satisfy the two concept-forming operations. The satisfied pairs are the special clusters called formal concepts.

A formal concept $c := (G_1, M_1)$ is a closure that is formed under the constraints of the two concept-forming operations (1)(2), where $(G_1')' = (M_1)' = G_1$. G_1 is called the *extent* of concept c, written as $\mathrm{ext}(c)$. Likewise, M_1 is called the *intent* of c, written as $\mathrm{int}(c)$ [53], which is also the centroid of the cluster [58, 59]. The universe containing all the concepts of a context $\mathbb{K}$ is represented by $\mathfrak{B}(G, M, I)$.

For FCA, concepts are the smallest units with reliable discovered knowledge. Thus, they are also treated as reliable inferences. For any concept c, its extent $\mathrm{ext}(c)$ indicates the predicted ongoing activities if the features in the intent $\mathrm{int}(c)$ are observed. The properties of closure ensure the maximization of ontological similarity and dependency in a concept.

Application 3. A concept c clusters and encapsulates similar activities into $\mathrm{ext}(c)$ on the basis of the common features in $\mathrm{int}(c)$. Furthermore, if a sequence of sensor events $\alpha \subset \mathrm{int}(c)$ is observed, the elements in $\mathrm{ext}(c)$ indicate the recognized ongoing activities given α.

$$\{ \quad \underbrace{g_2 g_6 g_9 g_{15}}_{\text{recognized ongoing activities}} \quad , \quad \overbrace{m_3 m_{10} m_{13}}^{\text{current observed data}} \quad \}$$

In the above example, $\mathrm{ext}(c) = \{g_2 g_6 g_9 g_{15}\}$ and $\mathrm{int}(c) = \{m_3 m_{10} m_{13}\}$. As described in Section 4.1.2, the sensor events in $\mathrm{int}(c)$ exist in all the patterns of activities in $\mathrm{ext}(c)$. Therefore, if the sensor events currently observed are m_3, m_{10} and m_{13}, the scope of possible ongoing activities should be g_2, g_6, g_9 or g_{15}.

4.1.4. Cluster Indexation by Concept Lattice

After generating the concepts that cluster similar classes by different centroids (i.e. feature variables), FCA provides a solution to automatically index all the discovered concepts. The purpose is to effectively manage and construct a graphical knowledge base to quickly retrieve inferences.

Concept lattice $\underline{\mathfrak{B}}$ is an ordered version of $\mathfrak{B}(G, M, I)$. All the concepts in $\underline{\mathfrak{B}}$ are ordered by a predefined partial order $\preceq$ indicating the hierarchical relations between two concepts [53].

Suppose (G_1, M_1) and (G_2, M_2) are two concepts, (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 a partial order in mathematics named *hierarchical order*. Meanwhile, (G_2, M_2) is the *superconcept* of (G_1, M_1). It is worth pointing out that the subconcept and the superconcept of a concept are not unique in $\mathfrak{B}(G, M, I)$ due to the transitive relation.

The lattice construction is a data mining process that first discovers all the concepts of a context $\mathbb{K}$, and then orders them simultaneously. In recent decades, many researchers have made great efforts in order to construct a concept lattice efficiently [60]. Our model is built on the basis of the construction algorithm proposed by Lindig [61].

Application 4. Three concepts, $\{g_6g_8g_{13}g_{13'}, m_{10}m_{11}\}$, $\{g_6g_{13}g_{13'}, m_8m_9m_{10}m_{11}\}$ and $\{g_{13}, m_4m_8m_9m_{10}m_{11}\}$, are mined from the matrix in Fig. 3. As shown in Equation (3), the last two concepts are the superconcepts of the first one.

$$\begin{aligned} \{g_{13}, m_4m_8m_9m_{10}m_{11}\} &\preceq \{g_6g_{13}g_{13'}, m_8m_9m_{10}m_{11}\} \\ &\preceq \{g_6g_8g_{13}g_{13'}, m_{10}m_{11}\} \end{aligned} \tag{3}$$

The relations between two concepts with different centroids are sorted and linked by the hierarchical order. Thus, a lattice $\underline{\mathfrak{B}}$ constructed by any lattice construction algorithm can be visualized as a graphical model.

4.1.5. Knowledge Visualization by Hasse Diagram

In mathematics, a finite partially ordered set can be depicted by a Hasse diagram. In our case, a lattice $\underline{\mathfrak{B}}$ can be visualized as the one shown in Fig. 6. Each node refers to a discovered concept, and partial orders are represented by edges, which are also named Galois connections [53].

There are two special nodes in a Hasse diagram: the topmost one $\{G, \varnothing\}$ named *Supremum* and the lowermost one $\{\varnothing, M\}$ named *Infimum*. They represent the initial and final states of an activity recognition process, respectively. Once a Hasse diagram is built, the next step is to use efficient algorithms to retrieve the knowledge encapsulated inside the concepts from its graphical structure.

Application 5. As we can see from Fig. 6, concepts are organized by different levels. From the top to bottom in a Hasse diagram, the range of possible activities is narrowed when more sensor events are observed.

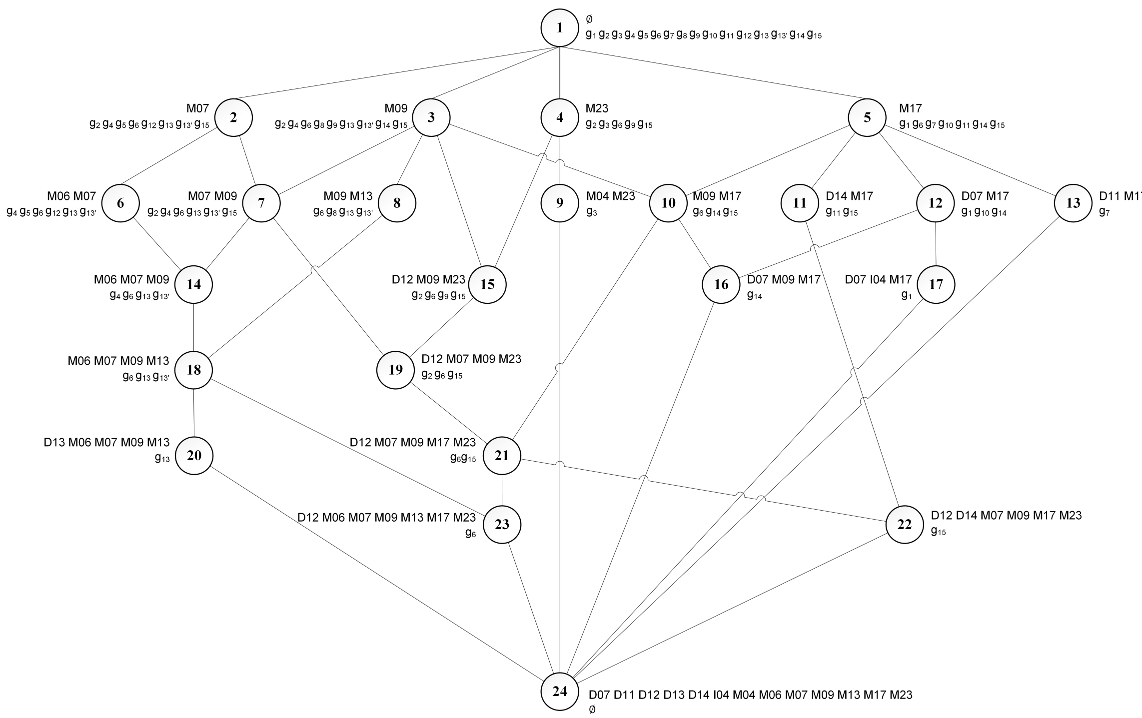

Figure 6: Hasse diagram of Matrix in Fig. 3

4.1.6. Pruning

To enhance the generalization of algorithm and improve the efficiency of lattice construction, in the feature selection, we propose two optional pruning processes to filter the useless attributes of a context. The first one is called the global pruning. The attributes that have extremely high or low occurrences should be removed from the context to avoid overfitting. This is because the attributes with extremely high occurrences have very limited ability to differentiate different activities. Similarly, the ones with extremely low occurrences are usually identified as noisy or meaningless data.

The second one is called the local pruning. Training data can be first grouped by activities according to their labels, and then a pruning operation is used to filter the correlations (i.e. crosses in the matrix) with extremely low occurrences in a group. In our previous research [18], attributes were classified into two types: essential and optional. Essential attributes mean that they are indispensable for an activity, in other words, they appear in all the patterns describing the same activity. Optional attributes usually represent personal preferences and they do not appear in each pattern. Therefore, in a group that contains all the patterns describing the same activity, the correlations with low occurrences will be treated as noisy data.

4.2. Knowledge Retrieval

Once unstructured knowledge is extracted by FCA and encapsulated in a Hasse diagram, efficient knowledge retrieval is crucial to fast query appropriate inferences based on the observed data. Consequently, we propose an FCA diagram search algorithm based on the breadth-first search (BFS). As an incremental search algorithm, when the new data updates the observed data, the current search will continue the previous one by loading the last located concept (i.e. node), and then will update the search result of this round for the next search. The new algorithm is called the *Half-Duplex Search (HDS) algorithm*, which consists of two parts. The top-down part aims to quickly find the first concept containing the observed data. Searching along the backward partial orders, the bottom-up part is designed to ensure that the newly located concept containing the observed data is the topmost one, and also the optimal solution.

As a pruned version of the Hasse diagram shown in Fig. 6, Fig. 7 depicts a simplified retrieval process, where the number in a node is the id of that node. Consider a successively observed sequence of sensor events $\alpha = \{M_9 \prec M_6\}$, node n_3 having M_9 is firstly located. When M_6 is observed, HDS starts a continuous search at the previous node n_3.

To locate n_{14}, the topmost concept having α, there are two alternative paths: $1 \to 3 \to 7 \to 14$ and $1 \to 3 \to 8 \to 18 \to 14$. This is because both n_7 and n_8 at the same level in the diagram have the same priority for BFS. If n_8 is chosen, the first located node having α is n_{18}. Evidently, it is not the topmost one because $n_{18} \preceq n_{14}$ and $\mathrm{ext}(n_{18}) \subset \mathrm{ext}(n_{14})$. Thus, if n_{18} is chosen as the search result, it will make some relevant inferences missing

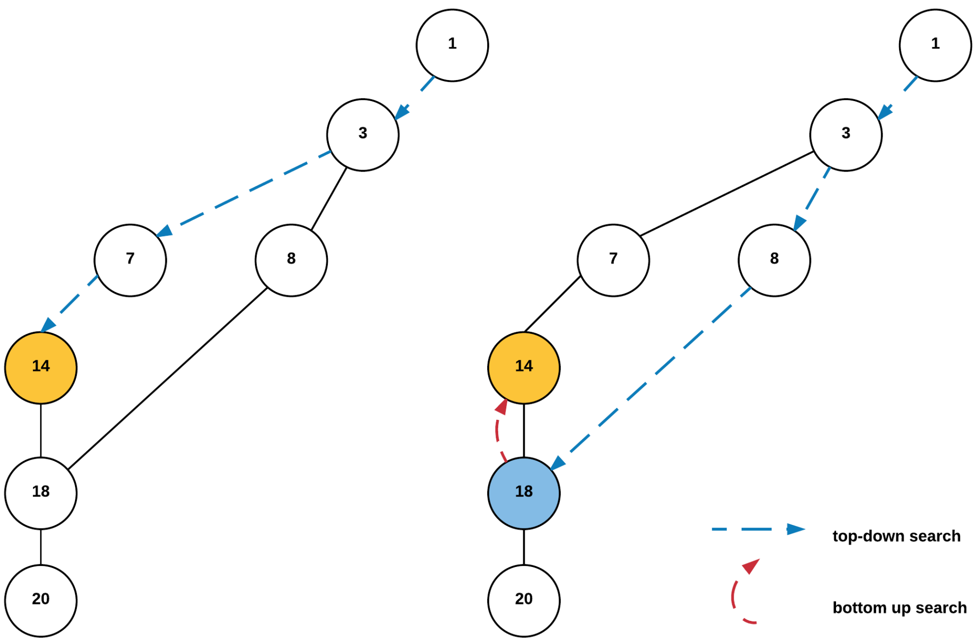

Figure 7: Knowledge retrieval using half-duplex diagram search

(e.g. the missing inferences equals $\text{ext}(n_{14}) - \text{ext}(n_{18})$). That is why we use the bottom-up algorithm to locate the topmost concept.

Algorithm 1: Top-down search of HDS algorithm

```
   Data: previous position p, sequence α.
   Result: first met concept containing α.
1  begin
2      fifo ← node[p]
3      while fifo do
4          if fifo[0] not visited then
5              mark as visited
6              if α ⊆ fifo[0].intent then
7                  return fifo[0]
8              else
9                  add fifo[0].successors into fifo
10         remove fifo[0] from fifo
11     end
12 end
```

Algorithms 1 and 2 are the pseudo-codes of top-down and bottom-up algorithms. In Algorithm 1, the first concept that contains the observed data α is located at Line 6 to 7. After that, Algorithm 2 incrementally searches the topmost concept from the last located node of Algorithm 1 along the reverse partial orders. A temporary set S is created to cache all the found concepts containing α. At the end (see Line 12), the topmost concept is determined by calculating the cardinalities of intents of the concepts in S. The concept having the minimal cardinality of intent is the topmost one due to the hierarchical order.

4.2.1. Behavioral Pattern Recognition

As shown in Section 3.3, multi-resident activities are classified as two categories: parallel and cooperative. Therefore, their behavioral patterns can also be divided into two types.

For multi-resident activities, behavioral data belonging to different residents or activities are often interweaved in their executions. This proposition is based on the analysis of the behavioral patterns of these two categories of activities. For parallel activities, two or more behavioral patterns are inde-

Algorithm 2: Bottom-up search of HDS algorithm

Data: located position p, sequence α.

Result: topmost concept containing α.

```
begin
    fifo ← node[p].predecessors
    S ← ∅
    while fifo do
        if fifo[0] not visited then
            mark as visited
            if α ⊆ fifo[0].intent then
                add fifo[0].predecessors into fifo
                S ← S ∪ fifo[0]
        remove fifo[0] from fifo
    end
    return arg min_{s∈S}(| s.intents |)
end
```

pendent of each other. Since there is no causal constraint between different activities, their behavioral data will be interweaved. In addition, almost all sensor events are triggered by only one resident (see the patterns of reading magazine and hanging up clothes in Fig. 2). For cooperative activities, due to the interaction and cooperation of residents, most sensor events are triggered by multiple residents at the same time, it is difficult to determine exactly who triggered which sensor event (see the pattern of play checkers in Fig. 2).

In order to simulate the interweaving situation, we create several temporary caches to simulate the long-term intentions of residents (i.e. the activities they are willing to do). As shown in Fig. 8, each cache stores the search result of last knowledge retrieval in Hasse diagram. It indicates the inference about all possible ongoing activities given partially observed sensor events. The system continuously loads subsequently observed sensor events. If a newly captured sensor event makes the new retrieval return the Infimum as the search result, it means that this sensor event is very different from the previously observed data in the ontology. It will be rejected by the current cache (i.e. the current intention) and the cache itself will rollback. The system will perform a roll polling operation to check if any existing cache can accept it. If all existing caches have triggered the rollback operation, the system will create a new cache to store this sensor event. In other words, a new parallel or cooperative activity may be in progress. At the beginning, there is only one primary cache for each resident. As time passes, residents start to interact with the other residents or carry out parallel activities, and more and more caches indicating different inferences are added into the polling.

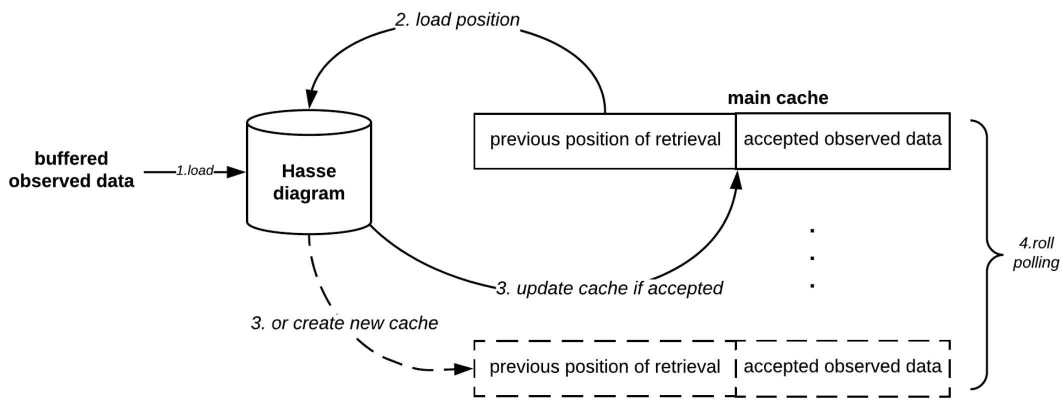

Figure 8: Recognition process using Hasse diagram

Once a cache has enough observed sensor events about an activity, the extent of the concept located by the cache determine the final recognition result.

Activities will be considered as recognized when there is only one object in the extent of the final located concept, such as n_{13}, n_{16} and n_{20}, or an object have never shown in its successive concepts, like g_4 in n_{14} could not be found in its subconcept n_{18}.

Suppose α = {M09≺M06≺M17≺D13≺D07≺M13≺M07} is a sequence indicating multi-resident activities g_{13} and g_{14}. Table. 1 illustrates the recognition process. The symbol $\curvearrowright$ represents a transition of inference and $\circlearrowleft_{Infimum}$ represents a rollback operation from the Infimum. At round 2, the bottom-up search ensures that node 14 is located, not node 18. At round 3, when M17 is observed, {M09M23M17} is excluded by previously located node 14 because there is no subconcept containing it except the Infimum. Thus, after the roll polling, a new cache is created to store M17. At round 8, when there is no more observable sensor event, the missing data M09 in the second cache will be automatically completed by the previous one observed at round 1.

Table 1: Example of multi-resident activity recognition

Round	**Observed** **Data** α	**Located** **Topmost Concept**	**Predictive** **Activities**
1	{M09}	node 3 $\{g_2g_4g_6g_8g_9g_{13}g_{13'}g_{14}g_{15},\mathbf{M09}\}$	$g_2g_4g_6g_8$ $g_9g_{13}g_{13'}g_{14}g_{15}$
2	{M09M06}	node 3 $\curvearrowright$ node 14 $\{g_4g_6g_{13}g_{13'},\mathbf{M06}\mathrm{M07}\mathbf{M09}\}$	$g_4g_6g_{13}g_{13'}$
3	{M09M06M17}	node 14 $\circlearrowright_{Infimum}$ $\{g_4g_6g_{13}g_{13'},\mathbf{M06}\mathrm{M07}\mathbf{M09}\}$ node 5 $\{g_1g_6g_7g_{10}g_{11}g_{14}g_{15},\mathbf{M17}\}$	$g_4g_6g_{13}g_{13'}$ $g_1g_6g_7g_{10}g_{11}g_{14}g_{15}$
4	{M09M06M17D13}	node 14 $\curvearrowright$ node 20 $\{g_{13},\mathbf{D13M06}\mathrm{M07}\mathbf{M09}\mathrm{M13}\}$ node 5 $\circlearrowright_{Infimum}$ $\{g_1g_6g_7g_{10}g_{11}g_{14}g_{15},\mathbf{M17}\}$	g_{13} $g_1g_6g_7g_{10}g_{11}g_{14}g_{15}$
5	{M09M06M17D13 D07}	node 20 $\circlearrowright_{Infimum}$ $\{g_{13},\mathbf{D13M06}\mathrm{M07}\mathbf{M09}\mathrm{M13}\}$ node 5 $\curvearrowright$ node 12 $\{g_1g_{10}g_{14},\mathbf{D07M17}\}$	g_{13} $g_1g_{10}g_{14}$
6	{M09M06M17D13 D07M13}	node 20 $\circlearrowright$ $\{g_{13},\mathbf{D13M06}\mathrm{M07}\mathbf{M09M13}\}$ node 12 $\circlearrowright_{Infimum}$ $\{g_1g_{10}g_{14},\mathbf{D07M17}\}$	g_{13} $g_1g_{10}g_{14}$
7	{M09M06M17D13 D07M13M07}	node 20 $\circlearrowright$ $\{g_{13},\mathbf{D13M06M07M09M13}\}$ node 12 $\circlearrowright_{Infimum}$ $\{g_1g_{10}g_{14},\mathbf{D07M17}\}$	g_{13} $g_1g_{10}g_{14}$
8	{**M09**M06M17D13 D07M13M07}	node 20 $\circlearrowright$ $\{g_{13},\mathbf{D13M06M07M09M13}\}$ node 12 $\curvearrowright$ node 16 $\{g_{14},\mathbf{D07M09M17}\}$	g_{13} g_{14}

4.2.2. Transition Matrix

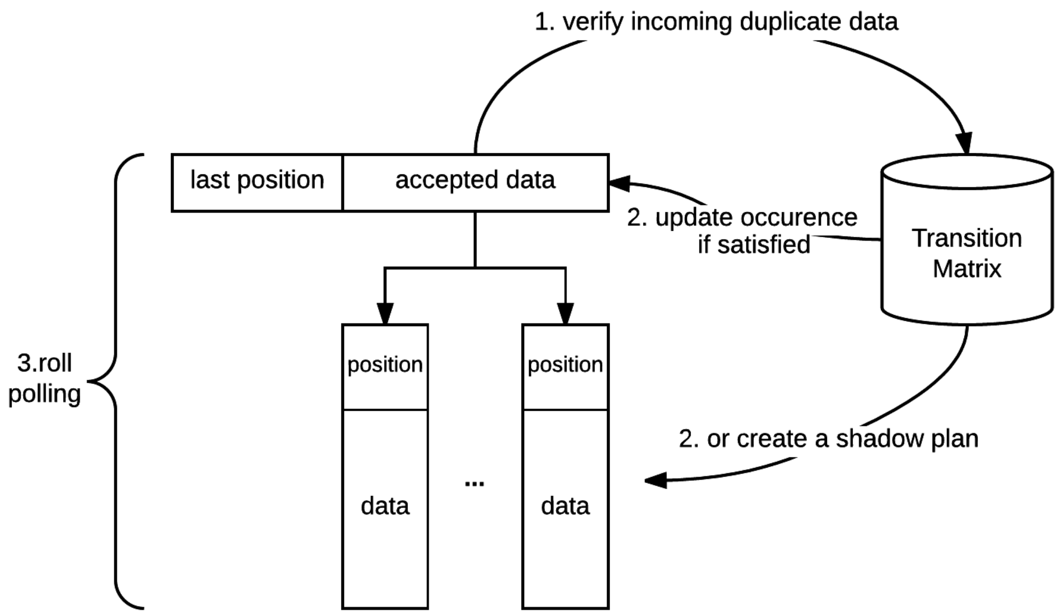

Figure 9: Identifying highly similar activities by transition matrix

Besides the FCA-based graphical model, for $|G|$ indexed activities, we define a transition matrix T_i for each of them to record the context information among sensor data (see Fig. 10). The objective is to distinguish similar or multi-level inheritance patterns. For instance, g_1 and g_2 are two highly similar activities, and the sensor events of g_1 are the subset of the ones of g_2. If they are performed by two residents at the same time, it is hard to correctly identify the real ongoing activities in the duplicate data without considering context information. Fortunately, transition matrices provide a feasible solution because even two similar patterns having exactly the same set of sensor data, the transition states among sensor data will be different.

Each T_i is a $N \times N$ square matrix where $N = |M| + 2$ and $|M|$ is the cardinality of indexed sensor events. Its columns or rows indicate an array $\{start, m_1, ..., m_j, ..., m_{|M|}, end\}$ where *start* and *end* are the boundary labels appearing in the training data.

For example, in the training phase, if a sequence describing activity g_5 is $\{start, m_8, m_9, m_9, end\}$, the elements $a_{0,8}, a_{8,9}, a_{9,9}$ and $a_{9,N-1}$ in the matrix T_5 should be updated.

$$T_1 = \begin{pmatrix} a_{00} & 2 & \cdots & 4 \\ 0 & a_{11} & \cdots & 5 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 5 & \cdots & a_{N-1,N-1} \end{pmatrix} \quad ... \quad ... \quad T_{|G|} = \begin{pmatrix} 0 & 0 & \cdots & 20 \\ 7 & 6 & \cdots & 11 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \cdots & 0 \end{pmatrix}$$

Figure 10: Transition matrices of different activities

In fact, duplicate data indicating repeated sensor events comes from frequent sampling or repeated triggering. In the recognition phase, when a new sensor event is repetitive, it will be only checked by the transition matrix. This is because duplicate sensor data will always be accepted by the caches containing it.

For example, because of few sensors deployed in an apartment, g_4, g_5 are two totally different activities, but they have similar sensor data. $g'_5 = \{m_8, m_9\}$ and $g'_4 = \{m_8, m_9, m_{10}\}$, so $g'_5 \subset g'_4$. As shown in Fig. 9, suppose the observed data are $\{m_9 \prec m_8 \prec m_{10} \prec m_8 \prec m_9\}$. Duplicated data m_8, m_9 will be detected after being observed (see step 1 in the figure). Because of no clear boundary, we could not simply justify that the duplicated m_8 belong to g_4, so we check the transition matrices to verify the transition $a_{10,8}$ in T_4. A cache will be created to store the duplicated data (see step 2) if and only if a_{ij} is lower than a threshold for any pattern of g_4. A roll polling operation (see step 3) will check each cache when a new duplicated data is observed.

4.3. Candidate Assessment

Because of few observed data, a concept usually has more than one element in its extent, which means that there are several candidates (possible ongoing activities) according to the observed data. Redundant candidates are ambiguous and useless to make decisions for real-time assistance. In this case, we desire to evaluate the relevance of each candidate in a concept and choose the most relevant one as the *local optimal prediction*. The relevance is defined as the similarity between existing learned patterns and the pattern to recognize.

As mentioned in the previous sections, an activity g_i could be accomplished by alternative patterns because of different personal preferences.

Furthermore, these derived patterns have flexible execution orders, repetitive events and optional data. At the same time, each resident may have a relatively stable preference to execute an activity. Namely, for the same resident executing an activity, there are only a few deviations among each execution. Based on this hypothesis, we take advantage of historical patterns containing the preferences of residents to generate a knowledge database called *accumulated matrix*. For each sensor data, we calculate its expectant position appearing in each activity to establish a series of naive distributions.

To measure the contextual similarities between historical patterns and the one of the current ongoing activity, average deviations are calculated using Root-Mean-Square Deviation (RMSD). It makes a quantitative comparison to estimate how well the current behavioral pattern fits accumulated historical data. A lower RMSD score indicates that the prediction is more accurate due to the adaptation to the 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 adaptation in comparison with historical data. Obviously, our assessment consists of two modules: accumulation and evaluation.

4.3.1. Accumulation

For each sensor data α_j in a training item α, which is a complete sequence of sensor data of activity 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 (4):

$$\sigma_{ij} = \sigma'_{ij} + j \tag{4}$$

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. The number of accumulated values σ_{ij} is the sum of positions of a sensor event α_j that appears in each pattern describing activity g_i. If a pattern is stored in an array, the position of a sensor event can be defined as its index value in the array. We accumulate such a value in order to calculate the average positions of sensor values and to calculate the standard deviation to measure the confidence of each average position. Equation (5) represents the same accumulation in another global view:

$$\sigma_{ij} = \sum_{k=1}^{N_{ij}} \sigma_{(ij,k)} \tag{5}$$

where N_{ij} represents the occurrences of the sensor data (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 activity g_i.

4.3.2. 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 (6).

$$\overline{\varphi_j} = \frac{1}{\#\alpha_j} \sum_{k=1}^{|\alpha|} k[\alpha_k = \alpha_j] \tag{6}$$

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

And then, for each candidate g_i, we calculate the deviation of α given g_i. Equation (7) expresses the root-mean-square deviation D_i of current sequence α executing g_i:

$$D_i = \sqrt{\frac{1}{|\alpha|} \sum_{\forall \alpha_j \in \alpha} (\overline{\varphi_j} - \frac{1}{N_{ij}} \sigma_{ij})^2} \tag{7}$$

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 adaptation to historical patterns.

5. Experiments

In this section, we use a benchmark dataset to evaluate the performance of our models. To compare the results with other models under the same measures, the following experiments are evaluated by both leave-one-out (LOOCV) and 3-fold cross-validations [62].

5.1. Dataset

The benchmark dataset adopted in the experiments is the CASAS Kyoto-4 multi-resident dataset [1]. It contains sensor events collected from a smart

[1]http://ailab.wsu.edu/casas/datasets

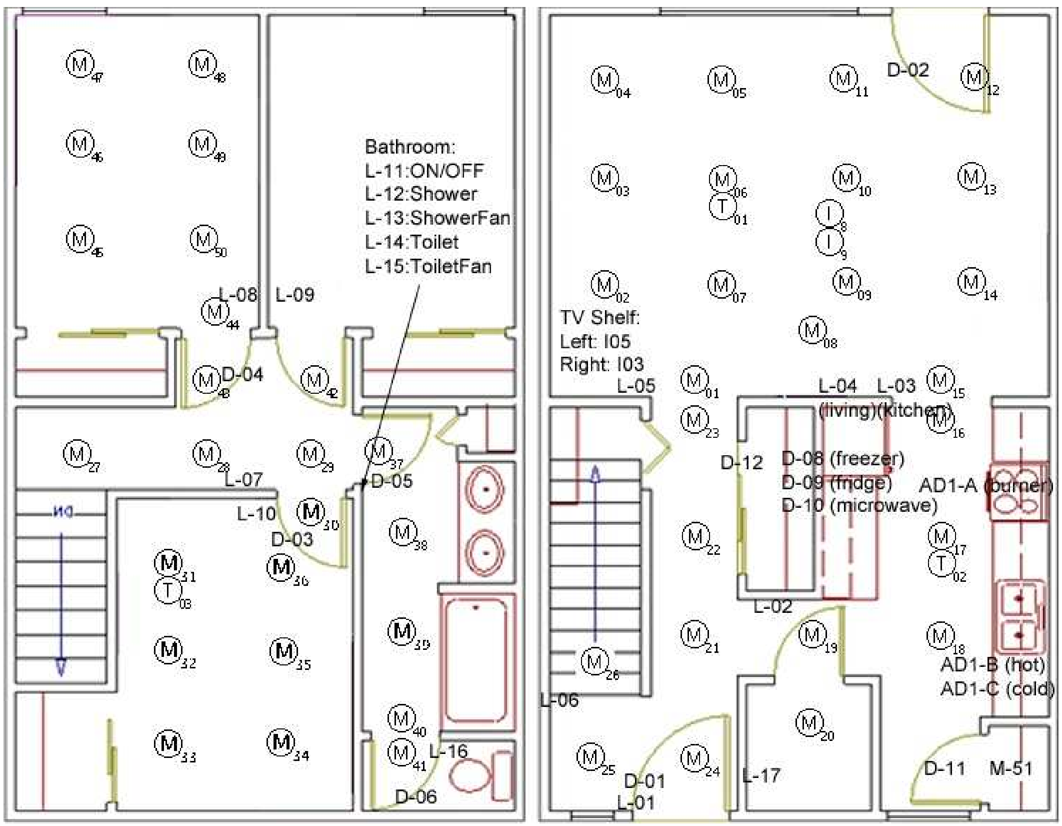

Figure 11: Non-intrusive sensor layout of the CASAS smart apartment

apartment testbed. To generate Kyoto4 dataset, researchers from CASAS laboratory recruited 40 volunteer participants to perform 15 activities in their smart apartment [51]. Each time, the multi-resident environment was occupied by two volunteers at the same time to perform assigned tasks concurrently. Collected sensor events were manually labeled with the activity ID it belongs to, and the ID of the resident who triggered it.

Fig. 11 shows the non-intrusive sensor layout of the CASAS smart apartment. Heterogeneous sensors are ubiquitously deployed in the house. Different types of sensor information have different distinguishable ability to classify different activities. In the CASAS datasets, researchers rely upon motion (M_i) and temperature sensors, analog sensors (D_i), as well as item sensors (I_i) to recognize activities that are being performed [51, 63, 64]. Motion sensors can provide active spatial information about human activities, and item sensors can record the interactions between residents and specific objects of interest such as refrigerators, keys and medicine containers [64]. Analog sensors can monitor the use of water and stove burner, as well as the open/close operations of doors, windows and cabinets. However, most of them cannot provide decisive information to distinguish who (or which activity) produced the sensor events [14].

Activity ID	**Activity**	**Type**	**Performers**
1	Fill medication dispenser	Individual	R1
2	Hang up clothes	Individual	R2
3	Move furniture	Cooperative	R1, R2
4	Read magazine	Individual	R2
5	Water plants	Individual	R1
6	Sweep floor	Individual	R2
7	Play checkers	Cooperative	R1, R2
8	Prepare dinner	Individual	R1
9	Set table	Individual	R2
10	Read magazine	Individual	R1
11	Pay bills	Cooperative	R1, R2
12	Pack picnic food	Individual	R1
13	Retrieve dishes	Cooperative	R1,R2
14	Pack picnic supplies	Cooperative	R2
15	Pack and bring supplies	Individual	R1

Table 2: Independent and cooperative activities in the CASAS dataset

As shown in Table 2, “R1” and “R2” refer to two different residents. Sometimes, two residents performed activities together called “joint activities" or in the same space. For joint activities, residents cooperate to jointly accomplish the task. The remaining independent activities are performed independently and in parallel [51]. The average activity times and number of sensor events generated for each activity are shown in Table. 3 [51].

Activity ID	R1 Time	R1 Events	Activity ID	R2 Time	R2 Events
1	3.0	47	2	1.5	55
3	0.7	33	3	0.5	23
5	2.5	61	4	1.0	18
7	3.5	38	6	2.0	72
8	1.5	41	7	2.0	25
10	4.5	64	9	1.0	32
12, 15	1.5	37	11	5.0	65
-	N/A	N/A	13, 14	3.0	38

Table 3: Average time (in minutes) and number of sensor events generated for each activity [51]

5.2. Modeling

As shown in Fig. 5, in the training phase, FCA extracts correlations from sequences of activities and then saves them into an FCA matrix. The first column indicates "*activity with the id of training item*" and the remaining ones indicate "*sensor events*". If a training item (i.e. a sequence of sensor events describing an activity) contains certain sensor events, it is affirmed that there is a binary relations among them. Correlations are represented as crosses in the matrix. Before constructing the Hasse diagram from training data, the pruning operation is executed for filtering redundant attributes.

Hidden ontological correlations are discovered by the matrix. Once different items describing the same activity have been put together, most of their internal attributes have clustered due to their similarity in the ontology. This is because an activity is associated with a specific location and constant interactive items. For example, activities involving preparing coffees will always interact with coffee cups. Another example is that the activity of preparing dinners always involves some fixed positions in the kitchen. Thus, the related correlations in the FCA matrix will cluster together and generate a group as a node in a Hasse diagram.

5.3. Results and Discussion

Cooperative activities could also be called joint activities if and only if at the same time, both of resident perform the same cooperative activity. The

	NBC [64]	HMM [64]	CRF [40]	TSM-HMM [38]	TSM-CRF [38]	FCA
Accuracy	63.27	60.90	58.41	75.77	75.38	**94.26**

Table 4: Comparison of recognition accuracies

cooperative could be regarded as well recognized when both of recognitions are correct. We compare our results with other references using the same dataset [33, 34, 38, 40, 51, 64, 65].

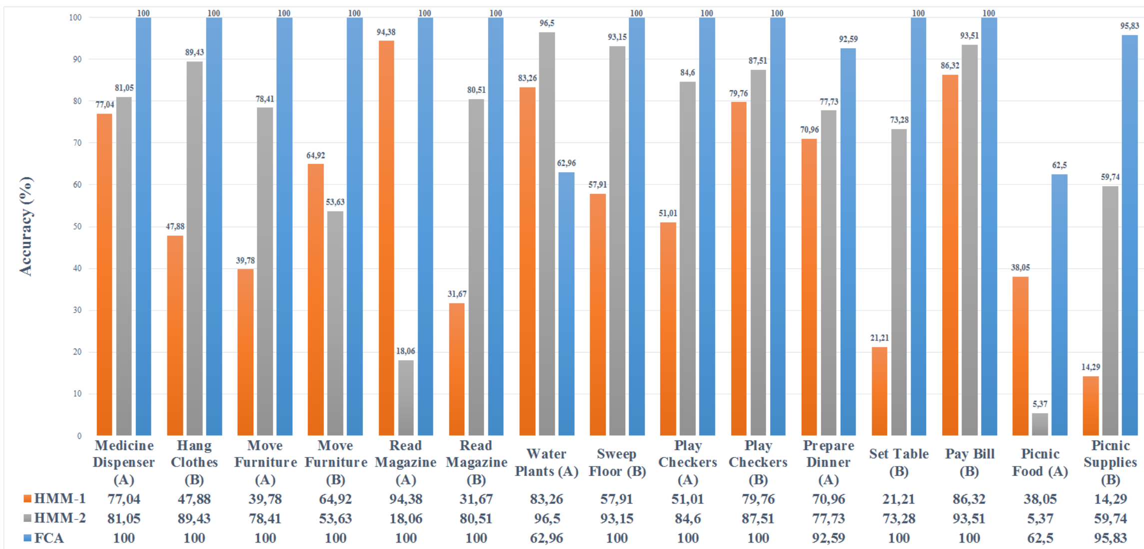

Figure 12: Performance of recognizing each multi-resident activity

First of all, we compare each activity recognition result with [51] and show the results in Fig. 12. Our results also surpass the results shown in Fig. 9 of [38]. The results are based on the same 3-fold cross-validation. As described in [51], HMM-1 is a single HMM model implemented for both residents. For HMM-2, an HMM model is built for each resident. In the results, we could see that most of the recognition are excellent except two activities: water plants (activity 5) and picnic food (activity 12). The reason has been indicated in [51] that the activities with insufficient sensor events will be difficult to differentiate from other activities. In the view of FCA models, the distinguishable ability of a sensor is negatively correlated with the number of shared activities. We also compare our results with other classical algorithms, including naive Bayes classifier (NBC), hidden Markov model (HMM), conditional random field (CRF) and their variants. The brief

Approach	Residents	Accuracy	Individual	Cooperative	Average	Precision	Recall	F-measure
CL-HMM [34]	R1	91.33±8.15	91.11±8.41	92.76±21.87	91.78±11.68	92.25±6.99	92.54±6.59	92.38±6.71
	R2	91.61±7.87	92.37±6.64	91.22±11.07	91.8±6.96	91.12±7.43	91.7±7.99	91.35±7.5
	Average	91.47±7.5	91.74±6.07	92.33±11.24	91.91±7.3	91.68±6.1	92.12±6.42	91.89±6.17
LHMM [34]	R1	92.36±8.48	93.86±7.89	65.19±43.57	81.4±21.32	93.25±7.46	91.93±7.56	92.48±6.98
	R2	94.17±5.05	90.8±7.52	96.42±5.48	93.61±5.12	93.9±5.44	93.43±6.4	93.61±5.63
	Average	93.27±6.21	92.33±6.95	82.77±21.3	87.53±11.22	93.58±5.41	92.68±6.18	93.1±5.62
FCA	R1	97.25±7.94	97.25±7.94	96.26±10.17	96.75±0.49	98.90±5.49	98.35±6.04	98.42±4.60
	R2	94.71±8.61	90.38±15.6	99.03±4.81	94.70±4.32	97.05±6.42	97.53±6.15	97.07±4.85
	Average	**95.98±1.27**	**93.81±3.43**	**97.26±2.11**	**95.53±1.73**	**97.97±0.93**	**97.94±0.41**	**97.75±0.68**

Table 5: Comparison of results categorized by different activity types and residents

Methods	Accuracy	Precision	Recall	F-measure
FCA†	92.86±12.54	97.62±2.06	95.24±11.66	95.10± 9.32
LHMM† [34]	88.23±10.23	81.46±10.27	79.43± 10.35	80.3±9.84
TSM-HMM∗ [38]	97.40	80.03	81.92	40.48
TSM-CRF∗ [38]	97.25	80.05	79.91	39.99
CHMM+Interaction vertices† [33]	78.26	-	-	-
Random Forest‡ [65]	88.60	-	-	-
SVM‡ [65]	83.70	-	-	-
Naive Bayes‡ [65]	81.20	-	-	-

Table 6: Comparison of results of joint activities

results are summarized in Table. 4.

After that, we compare our results of independent parallel activity recognition with another reference [34] (see Table. 5). In this comparison, we use leave-one-out method to evaluate the performance. The results are classified by different residents and the types of activities. According to the results under different metrics, we could find that our FCA-based method outperforms the other HMM-based methods. In the part of recognizing joint activities, the FCA-based method also has excellent performance (see Table. 6[2]). It has obtained the best results in terms of precision, recall and F-measure.

The proposed FCA-based model has better capacity than the previous version [17] while identifying similar activities. This is because the newly added transition matrices can be useful when two patterns are highly similar. On the premise of keeping the context information, the FCA-based model with the transition matrices reduces the influence of imbalanced distributions of training data and enforce the impact of internal regulars of patterns. Even two patterns consist of the same sensors events, their sequential contexts would be different. It means that for a sensor event in two highly

[2]the methods marked by † use the leave-one-out cross-validation, the one marked by * uses the 5-fold cross-validation, and the ones marked by ‡ use the 10-fold cross-validation.

similar patterns, its previous and successive sensor events will not always be the same ones. Compared with two HMM methods in [51], the overall performance of activity recognition has increased 37.02% and 22.76%. In the LOOV experiments, our methods improve 4.51% and 2.71% accuracies.

Besides, the FCA-based model simulates the real scenarios that include the interweaving patterns. There is no explicit segmentation to reveal the beginning and end of a sequence indicating an activity. To determine a sensor data belongs to which patterns, the conventional HMM methods use a series of probabilities such as joint and transition probabilities to judge the affiliations of a sensor data. If a posteriori probability is lower than a threshold, then the systems will judge that it belongs to another pattern. In our method, we do not directly use probability to evaluate the confidential degrees, however, we make the decision from the semantic parts. If a sensor data has great semantic gaps with the others, then it will be judged as one part of another pattern.

Comparing with the HMM methods, the FCA-based models can give a scope of possible ongoing activities and refine the results by the RMSD assessment. However, it works well only for the independent activities performed in parallel. This is because one person's activities will be affected by another one, especially for the cooperative activities. Thus, the RMSD assessment has to wait for enough data to infer the most reliable recognition in the case of cooperative activity recognition.

The FCA-based models have considered the robustness problem about handling noisy sensor data. For each unseen pattern that is not in the training dataset, but in the test dataset, the models will compare its similarity with learned patterns and propose the most similar activity cluster as the recognition result. Similar to handle with noisy sensor data, in [66], we defined six common anomalies existing in the behavioral patterns of people with Alzheimer's disease and proposed their solutions. Likewise, in the worst case, unreliable sensor data will be evaluated and classified into a similar activity cluster.

6. Conclusion

In this paper, we address the problem of multi-resident activity recognition in non-intrusive sensor-based smart homes. We use a knowledge-driven sequential pattern mining solution based on formal concept analysis to discover knowledge from sequential and temporal data. A graphical knowledge base is automatically constructed by the formal lattice. Using a lattice search strategy, we can automatically and incrementally infer the most possible ongoing activities given a part of observed data. The incremental knowledge retrieval makes the static formal lattice containing ontological knowledge become dynamic. The combination of the graphical knowledge base and the transition information make the FCA-based model reduce the dependency of stable data distribution in the training data. The experimental results show that the recognition accuracy outperforms than some traditional statistical or probabilistic models. In the future, we will explore a better measure for evaluating possible candidates than the RMSD one. Moreover, the temporal relations such as time interval between two sensor events will also be considered in the future improvement.

References

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] 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.

[5] Kirsten KB Peetoom, Monique AS Lexis, Manuela Joore, Carmen D Dirksen, and Luc P De Witte. Literature review on monitoring technologies and their outcomes in independently living elderly people. *Disability and Rehabilitation: Assistive Technology*, 10(4):271–294, 2015.

[6] Xing Zhang, Haipeng Wang, and Zhiwen Yu. Toward a smart home environment for elder people based on situation analysis. In *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on*, pages 7–12. IEEE, 2010.

[7] 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.

[8] Jessamyn Dahmen, Brian L Thomas, Diane J Cook, and Xiaobo Wang. Activity learning as a foundation for security monitoring in smart homes. *Sensors*, 17(4):737, 2017.

[9] Adam Greenfield. *Everyware: The dawning age of ubiquitous computing*. New Riders, 2010.

[10] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[11] Hemant Ghayvat, Subhas Mukhopadhyay, Xiang Gui, and Nagender Suryadevara. WSN and IOT-based smart homes and their extension to smart buildings. *Sensors*, 15(5):10350–10379, 2015.

[12] Charith Perera, Dumidu S Talagala, Chi Harold Liu, and Julio C Estrella. Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in iot clouds. *IEEE Transactions on Computational Social Systems*, 2(4):171–181, 2015.

[13] Juan Ye and Graeme Stevenson. Semantics-driven multi-user concurrent activity recognition. In *International Joint Conference on Ambient Intelligence*, pages 204–219. Springer, 2013.

[14] 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.

[15] Terence KL Hui, R Simon Sherratt, and Daniel Díaz Sánchez. Major requirements for building smart homes in smart cities based on internet of things technologies. *Future Generation Computer Systems*, 2016.

[16] 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.

[17] Jianguo Hao, Abdenour Bouzouane, and Sébastien Gaboury. Complex behavioral pattern mining in non-intrusive sensor-based smart homes using an intelligent activity inference engine. *Journal of Reliable Intelligent Environments*, pages 1–20, 2017.

[18] 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.

[19] 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.

[20] Fabien Cardinaux, Deepayan Bhowmik, Charith Abhayaratne, and Mark S Hawley. Video based technology for ambient assisted living: A review of the literature. *Journal of Ambient Intelligence and Smart Environments*, 3(3):253–269, 2011.

[21] Alexandros André Chaaraoui, Pau Climent-Pérez, and Francisco Flórez-Revuelta. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, 39(12):10873–10888, 2012.

[22] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim. A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments. *Sensors*, 14(7):11735–11759, 2014.

[23] Allen Y Yang, Sameer Iyengar, Philip Kuryloski, and Roozbeh Jafari. Distributed segmentation and classification of human actions using a wearable motion sensor network. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[24] Yan Song, Jinhui Tang, Fan Liu, and Shuicheng Yan. Body surface context: A new robust feature for action recognition from depth videos. *IEEE transactions on circuits and systems for video technology*, 24(6):952–964, 2014.

[25] Qin Ni, Ana Belén García Hernando, and Iván Pau de la Cruz. The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors*, 15(5):11312–11362, 2015.

[26] 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.

[27] Corinne Belley, Sebastien 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.

[28] 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.

[29] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[30] David Barber. *Bayesian reasoning and machine learning.* Cambridge University Press, 2012.

[31] Aaron S Crandall and Diane J Cook. Using a hidden markov model for resident identification. In *Proceedings of the 2010 Sixth International Conference on Intelligent Environments*, pages 74–79. IEEE Computer Society, 2010.

[32] Li Liu, Shu Wang, Guoxin Su, Zi-Gang Huang, and Ming Liu. Towards complex activity recognition using a bayesian network-based probabilistic generative framework. *Pattern Recognition*, 68:295–309, 2017.

[33] Yi-Ting Chiang, Kuo-Chung Hsu, Ching-Hu Lu, Li-Chen Fu, and Jane Yung-Jen Hsu. Interaction models for multiple-resident activity recognition in a smart home. In *International Conference on Intelligent Robots and Systems*, pages 3753–3758. IEEE, 2010.

[34] Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Modeling interaction in multi-resident activities. *Neurocomputing*, 230:133–142, 2016.

[35] Liang Wang, Tao Gu, Xianping Tao, Hanhua Chen, and Jian Lu. Recognizing multi-user activities using wearable sensors in a smart home. *Pervasive and Mobile Computing*, 7(3):287–298, 2011.

[36] Yi-Ting Chiang, Ching-Hu Lu, and Jane Yung-Jen Hsu. A feature-based knowledge transfer framework for cross-environment activity recognition toward smart home applications. *IEEE Transactions on Human-Machine Systems*, 47(3):310–322, 2017.

[37] Yaqing Liu, Dantong Ouyang, Yong Liu, and Rong Chen. A novel approach based on time cluster for activity recognition of daily living in smart homes. *Symmetry*, 9(10):212, 2017.

[38] Rong Chen and Yu Tong. A two-stage method for solving multi-resident activity recognition in smart environments. *Entropy*, 16(4):2184–2203, 2014.

[39] Markus Prossegger and Abdelhamid Bouchachia. Multi-resident activity recognition using incremental decision trees. In *Adaptive and Intelligent Systems*, pages 182–191. Springer, 2014.

[40] Kuo-Chung Hsu, Yi-Ting Chiang, Gu-Yang Lin, Ching-Hu Lu, Jane Yung-Jen Hsu, and Li-Chen Fu. Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 417–426. Springer, 2010.

[41] Hongqing Fang and Chen Hu. Recognizing human activity in smart home using deep learning algorithm. In *Control Conference (CCC), 2014 33rd Chinese*, pages 4716–4720. IEEE, 2014.

[42] Stefan Oniga and Jozsef Suto. Human activity recognition using neural networks. In *Control Conference (ICCC), 2014 15th International Carpathian*, pages 403–406. IEEE, 2014.

[43] Licheng Zhang, Xihong Wu, and Dingsheng Luo. Human activity recognition with HMM-DNN model. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*, pages 192–197. IEEE, 2015.

[44] Juan Ye, Graeme Stevenson, and Simon Dobson. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47–70, 2015.

[45] Mohammad Arif Ul Alam, Nirmalya Roy, Archan Misra, and Joseph Taylor. Cace: Exploiting behavioral interactions for improved activity recognition in multi-inhabitant smart homes. In *International Conference on Distributed Computing Systems*, pages 539–548. IEEE, 2016.

[46] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[47] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[48] Salah Althloothi, Mohammad H Mahoor, Xiao Zhang, and Richard M Voyles. Human activity recognition using multi-features and multiple kernel learning. *Pattern recognition*, 47(5):1800–1812, 2014.

[49] Ahmad Jalal, Md Zia Uddin, and T-S Kim. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics*, 58(3), 2012.

[50] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern recognition*, 61:295–308, 2017.

[51] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing*, 1(1):57–63, 2010.

[52] Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Multioccupant activity recognition in pervasive smart home environments. *ACM Computing Surveys*, 48(3):34, 2016.

[53] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 1999.

[54] 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.

[55] 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.

[56] 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.

[57] Jonas Poelmans, Dmitry I Ignatov, Sergei O Kuznetsov, and Guido Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert systems with applications*, 40(16):6538–6560, 2013.

[58] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

[59] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[60] Sergei O Kuznetsov and Sergei A Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):189–216, 2002.

[61] Christian Lindig. Fast concept analysis. In *International Conference on Conceptual Structures*, pages 152–161, 2000.

[62] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[63] Diane J Cook and Maureen Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5):480, 2009.

[64] Diane J Cook. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 27(1):32–38, 2012.

[65] Raihani Mohamed, Thinagaran Perumal, Md Sulaiman, Norwati Mustapha, MN Zainudin, et al. Multi label classification on multi resident in smart home using classifier chains. *Advanced Science Letters*, 24(2):1316–1319, 2018.

[66] Jianguo Hao, Sébastien Gaboury, and Bruno Bouchard. Cognitive errors detection: Mining behavioral data stream of people with cognitive impairment. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 15. ACM, 2016.