J Ambient Intell Human Comput (2018) 9:679–698
DOI 10.1007/s12652-017-0467-7

ORIGINAL RESEARCH

Activity inference engine for real-time cognitive assistance in smart environments

Jianguo Hao[1] · Abdenour Bouzouane[1] · Bruno Bouchard[1] · Sébastien Gaboury[1]

Received: 31 October 2016 / Accepted: 14 February 2017 / Published online: 17 March 2017

Abstract Recent research in ambient intelligence allows wireless sensor networks to perceive environmental states and their changes in smart environments. An intelligent living environment could not only provide better interactions with its ambiance, inside electrical devices and everyday objects, but also offer smart services, even smart assistance to disabled or elderly people when necessary. This paper proposes a new inference engine based on the formal concept analysis to achieve activity prediction and recognition, even abnormal behavioral pattern detection for ambient-assisted living. According to occupants' historical data, we explore useful frequent patterns to guide future prediction, recognition and detection tasks. Like the way of human reasoning, the engine could incrementally infer the most probable activity according to successive observations. Furthermore, we propose a hierarchical clustering approach to merge activities according to their semantic similarities. As an optimized knowledge discovery approach in hierarchical ambient intelligence environments, it could optimize the prediction accuracies at the earliest stages when only a few observations are available.

✉ Jianguo Hao
jianguo.hao1@uqac.ca

Abdenour Bouzouane
abdenour.bouzouane@uqac.ca

Bruno Bouchard
bruno.bouchard.phd@ieee.org

Sébastien Gaboury
sebastien.gaboury@uqac.ca

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

Keywords Ambient intelligence · Activity prediction · Activity recognition · Anomaly detection · Data mining · Formal concept analysis

1 Introduction

With faster development of information and communication technologies, ambient intelligence (AmI) has become a popular field of research in recent years (Ramos et al. 2008). It refers to a digital living environment that proactively supports the occupants inhabiting it and lets them enjoy intelligent user experiences (Sadri 2011). As an interdisciplinary domain, AmI incorporates multiple cutting-edge technologies such as artificial intelligence (AI), things of Internet (IoT) and human–computer interaction (HCI), etc. (Remagnino and Foresti 2005; Cook et al. 2009). Summing up the characteristics of AmI, its advantages are such as these: first of all, it is aware of environmental changes. Then, with the help of computational units, it could rapidly respond various requirements in a short time. Last, it could provide better personal interactive experiences concerning context awareness.

AmI utilizes IoT to build a network of objects embedded with measurable components like sensors, RFID tags and readers, or actuators to collect and exchange data in real-time (Gubbi et al. 2013). These objects include home appliances, household furniture and the other daily commodities. In recent years, considerable attention has also been paid to wearable devices, to accurately collect user's behavioral information or vital signs (Acampora et al. 2013). These ubiquitous electronics could make it possible to achieve real-time monitoring and avoid risks at the earliest stages. Besides, due to heterogeneous components, AmI continuously produces large-scale unintelligible data.

Output data involve environmental changes (positions, movements, temperature and pressure, etc.), and consumption (energy or resources) (Cook et al. 2009). They are usually temporal and sequential, even unstructured and chaotic. As a consequence, machine learning and data mining, two subfields of artificial intelligence, are widely used to automatically interpret, infer and understand the current situation in order to respond real-time requirements of occupants (Ramos et al. 2008; Sadri 2011). Moreover, AmI could seamlessly integrate touchscreen, speech processing, assisted social robots or any other advanced HCI technologies with daily activities (Boudreault et al. 2016; Broekens et al. 2009). Ideally, AmI is sensitive to the needs of occupants. Current situations will be analyzed and appropriate feedbacks or interventions will be given out.

Considering the advantages above, our living environments could be rebuilt as "smart homes" (De Silva et al. 2012) under the standards of AmI to offer a wide range of care and assistance services for people who cannot live independently (Cook and Das 2007), especially for patients with Alzheimer's disease.

In the last few decades, population aging has become a worldwide crucial issue (United Nations 2013, 2016). As a common feature associated with elderly people, cognitive impairment has attracted increasing attention from scientific communities. Severe deterioration in cognitive skills will induce memory difficulties and could not be able to well schedule and undertake activities of daily living (ADLs), which involve basic self-care tasks (Barberger-Gateau and Fabrigoule 1997; Suryadevara et al. 2013). In addition, people with cognitive impairment will cause more abnormal behaviors while performing diverse activities and need more care in daily life.

As a promising solution, smart homes attempt to make disabled or elderly people live on their own with less nursing care by providing appropriate assistance while carrying out activities. To achieve this goal, as one of the most important prerequisites, smart homes have to recognize ongoing activities and to understand the occupants' real needs behind them. Therefore, understanding an occupant's true intention has significant effects in ensuring high-quality services for real-time assistance. Hence, for the intention understanding, activity recognition is the minimum requirement, and prediction is the ultimate objective.

In some cases, especially in the context of assisted living environment, recognizing a completely finished activity may not be helpful because none of assistance could be provided during its execution. Compared with the recognition task, prediction is required to infer the ongoing activity using only a partial observed data. The performance of smart homes could be improved if it enables to predict an ongoing activity as early as possible according to cumulative observed data.

The objective is to prefer recognizing unfinished activities rather than classify the completed ones (Ryoo 2011). Activity prediction is necessary to help occupants prevent dangerous activities before they occur. The real-time data analysis could be necessary to determine whether an abnormal behavior has taken place and if any preventive intervention is required. Once current behavioral pattern tends to be abnormal, we wish that AmI system will decide whether starts preventive interventions or notices the family members, neighbors or caregiver. Thus, in this paper, we propose a real-time inference engine that could infer ongoing activities using partially observed data and detect abnormal behavioral patterns. The modeling of inference engine is an attempt to learn sequential patterns from activities to construct a case-based pattern reasoning model. It takes advantage of context-aware rules to prompt individuals for well scheduling and carrying out complex activities.

The remainder of this paper is organized as follows. Section 2 outlines the related works about AmI. Section 3 introduces the knowledge representation of formal concept analysis. The proposed inference engine is also presented in that section. Section 4 proposes an extra ontological clustering approach to enhance the interpretation of data. Several error detection agents are presented in Sect. 5. Section 6 presents the experimental results. Section 7 summarizes the results and discusses their performances. Finally, the conclusion is reported in Sect. 8.

2 Related works

Because of large-scale generated data, AI has become an efficient solution to handle with various AmI problems. Most of raw data containing valuable knowledge provide regular patterns or useful cases, but it is usually hard to directly use them to solve practical problems (Jonyer et al. 2001). Each indexed pattern in the knowledge base is an assertion about the real case. Thus, it is essential to determine an efficient representative form to index, organize and retrieve the knowledge. With the help of diverse AI techniques like knowledge representation, machine learning and data mining, useful information could be discovered from continuous data streams. Based on varied data features, different types of data streams should be interpreted and analyzed by suitable approaches. For example, image processing and pattern recognition are effective to deal with image sequences captured by vision-based facilities (Chen et al. 2012). For numerical or the other sequences captured by sensor-based facilities, they are usually processed by traditional data mining and machine learning methods (Alsheikh et al. 2014).

In this paper, we discuss three popular topics existing in the AmI research community: activity prediction, recognition and anomaly detection. They concern feature extraction of sequences, likelihood evaluation, and classification. In the following words, we present the current state of arts about the solutions of three research topics in different categories.

2.1 Sequence classification

According to different temporal orders, the sensing data from heterogeneous sensors form a multi-dimensional data stream. Reasonably, activity recognition could also be treated as the discrete sequence data classification problem (Aggarwal 2014). Chien and Huang (2014) used the discriminant sequence patterns and activity transition patterns to recognize and predict activities.

2.2 Case-based classification

Case-based classification, or case-based reasoning, is a method that summarizes and reuses old similar experiences to understand and solve new situations (Kolodner 1992). It is also a unified approach to knowledge representation, classification, and learning. As a common strategy of human, the reasoning by reusing past cases is frequently applied and supported by the results from cognitive psychological research (Aamodt and Plaza 1994). It usually integrates cases as distributed subunits within an indexable knowledge structure for the later matching with similar cases.

2.3 Rule-based classification

Compared with learning-based approaches, rule-based ones are more readable and comprehensive. Bao and Intille (2004) tested various machine learning approaches to recognize activities from user-annotated acceleration data, and concluded that C4.5 decision tree received the highest recognition accuracy. Chen et al. (2010) proposed a heterogeneous feature selection approach using J48 decision tree to create a classification model.

2.4 Probabilistic classification

As classical probabilistic machine learning methods, hidden Markov model (HMM), conditional random fields (CRF) and their derivations are widely used in the ambient intelligence applications (Kim et al. 2010; Aggarwal and Ryoo 2011).

Van Kasteren et al. (2011) proposed a two-layer hierarchical model using the hierarchical hidden Markov model to cluster sensor data into clusters of actions, and then use them to recognize activities. Nazerfard and Cook (2013, 2015) proposed a two-step prediction model based on Bayesian Network. In the beginning phase, the model predicts the features about the following sensor stream. After that, the next activity with the start time is predicted by using continuous normal distribution. Besides the approaches mentioned above, some researchers also adopt classical machine learning techniques to solve AmI problems (Alsheikh et al. 2014; Minor et al. 2015).

2.5 Ontology-based classification

Ontology is formal, explicit representation of domain knowledge that consists of concept, concept properties, and relationships between concepts (Boujemaa et al. 2008). An ontology is created by linking individual concepts with mutual relations. Thus, the semantic gap between low and high level concepts could be bridged according to the predefined relations (Aggarwal 2014).

Yau and Liu (2006) proposed an OWL-based situation ontology to hierarchically model situations to facilitate sharing and reusing of situation knowledge and logic inferences. Ye and Dobson (2010) defined a data structure called context lattice to infer human activities. It could express the semantic knowledge about the relationships of low-level sensor data and high-level scenarios. Suryadevara et al. (2013) introduced a lexicographic tree to generate frequent pattern sets in terms of duration. With this semantic model, the next action is predicted from its past training data.

2.6 Comparison

Unlike the classification task using feature vectors, sequences or data streams do not have explicit features. A sequence or pattern is an ordered list of events representing as symbolic and numerical values, vectors or their combination (Xing et al. 2010). Each previous category has its advantages to solve some specific data types and cases. In the following words, we compare the categories with different criteria.

The first criteria is about missing values. Events in the AmI context are successively observed, so a feature vector captured at a certain time usually has many missing values. Thus, besides incremental methods, the other ones like rule-based classification is more difficult to achieve real-time activity prediction.

The second one is related to the generalization. For case-based and ontology-based classifications, they depend too much on historical cases or predefined semantic relations between target variables of interest and their features. Thus, they could not well handle with some extreme cases which are not considered before. On the contrary, probabilistic classification is purely based on

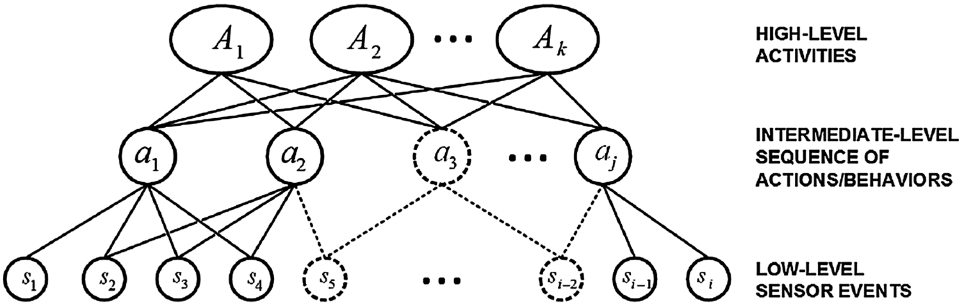


Fig. 1 Multilevel structure in smart environments

the occurrence probability of an assertion in reality to determine the degree of certainty, so it has better generalization capacity.

The third one concerns anomaly detection. To detect an abnormal behavioral pattern, we should be familiar with its particular characteristics. For instance, an abnormal one could be caused by irregular behavioral features like wrong component events (missing, redundant or irrelevant) or abnormal execution (disordered or distraction). Because of dynamic forms, it is hard to ensure that all possible abnormal situations are considered and covered in the training datasets. Moreover, labeling abnormal items are also prohibitively expensive (Chandola et al. 2009). Thus, for frequency-based probabilistic classification and the other four similarity-based classifications, without explicit knowledge representation, methods in one single category could not deal with all abnormal situations and they are easy to suffer from high false alarm rates (Hao et al. 2016b).

Our proposition is quite familiar with the case-based reasoning that organizes the past behavioral patterns for identifying current pattern in the background of AmI. Compared to the hierarchical index structures used in the case-based reasoning (Bareiss 1989; Porter et al. 1990), our inference engine adopts a more readable and comprehensive model to organize and infer the knowledge in real-time.

Some of our previous works are the prerequisites of our new proposition. A logic framework based on four-level lattice was also defined for plan recognition (Bouchard et al. 2007), and it prompts us to solve the activity recognition problem with more mature lattice-based models. Furthermore, a real-time knowledge-driven incremental approach based on FCA was proposed to predict and recognize ongoing activities (Hao et al. 2016a). An assessment based on the root-mean-square-deviation (RMSD) which was used to evaluate the relevance of each intermediate prediction was also proposed at the same time. Most common behavioral errors among people with cognitive impairment are defined and formulated in Hao et al. (2016b). Customer-built detectors were proposed to identify abnormal patterns in the continuous data.

3 Inference engine based on FCA

In this section, we first briefly introduce the basic notions about formal concept analysis, then explain each component and their roles in the knowledge representation. Afterwards, according to the context of AmI problems, we demonstrate the feasibility of representing AmI knowledge bases in the form of FCA, and finally present the details about how to model our inference engine.

3.1 Formal concept analysis

Formal concept analysis (FCA) is a mathematical theory that derives concept hierarchies from a given dataset. Its core idea is to cluster similar individuals of interest sharing same features. Correlations of these 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] Because of the excellent performance of knowledge representation and extraction in large volume of unstructured data (De Maio et al. 2016, 2017), FCA is widely used in various domains like knowledge discovery, ontology learning (Loia et al. 2013), semantic annotation (De Maio et al. 2014), information retrieval and recommender system (De Maio et al. 2012) to extract useful information and to construct a knowledge graph or graphical ontology for data organization, visualization and mining (Poelmans et al. 2013).

FCA requires to choose suitable target variables of interest and their features for modeling. There is some reasonable granularity in smart environments, which could be chosen as modeling objects (Hao et al. 2016a). Figure 1 depicts the common multilevel granularity to solve activity-centered problems (Van Kasteren et al. 2011; Chien and Huang 2014). Three levels of granularity represent different kinds of data. Fine-grained discrete components are located at relatively lower levels and coarse-grained ones

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

	m_a	m_b	m_c	m_d	m_e	m_f
g_1	×	×				
g_2	×	×	×			
g_3	×			×		
g_4	×	×	×	×		×
g_5	×					
g_6					×	×

Fig. 2 Matrix representing a context $\mathcal{K}$

are positioned at higher levels. Each coarse-grained supercomponent is composed of one or more fine-grained subcomponents. A long-term activity is composed of more than one short-term action or sensor event. However, an action or a sensor event is too short in response time to provide appropriate assistance during its execution. Thus, we choose activities as target variables of interest and observed data as features in this paper.

In the following words, we describe each FCA component and demonstrate how to use FCA for knowledge representation, modeling and incremental inference.

3.1.1 Formal context (context)

A context $\mathcal{K}$ is the mathematical abstraction of a real scene. It is represented as a triplet $\mathcal{K}(G, M, I)$ which consists of two disjoint sets G and M. Set I represents their Cartesian products as binary relations, which defines the issues to be solved. The elements of G representing higher-level larger discrete components are called the *objects*, and the elements of M representing lower-level smaller subcomponents are called the *attributes*. Attributes are the descriptors of objects. If object g is related with attribute m, we write gIm (Ganter and Wille 1999).

As a knowledge storage way, a context encodes unstructured or heterogeneous information in a machine-recognizable form. The triple $\mathcal{K}(G, M, I)$ could be saved as a $|G| \times |M|$ matrix. An example is given in Fig. 2. If $g_i I m_j$ is true, there is a cross in the element of row g_i and column m_j.

Example As shown in Fig. 1, we could represent the set of coarse-grained activities as G and the set of fine-grained observed data as M. Affiliated relations I could be interpreted as "is measured/described by". If an activity $g_i \in G$ is composed of $B \subset M$, then crosses should fill up the elements (g_i, m_j), $\forall m_j \in B$ in the matrix.

3.1.2 Concept-forming operations

A pair of closure operations, so-called the *concept-forming operators* is induced to discover dependent associations by observed variables. They are defined as a similarity metric to generate clusters.

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

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

to find out all the associated attributes $A' \subseteq M$ relating the same observed objects A. Likewise, for $B \subseteq M$, we define

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

to find out all the associated objects $B' \subseteq G$ relating the same observed attributes B (Ganter and Wille 1999).

The combination of two concept-forming operations could maximize the dependencies between the two sets of variables and generate a stable closure.

Example In Fig. 2, to find out the commonly shared attributes between g_3 and g_4, we have $\{34\}' = \{ad\}$. Likewise, the similar activities having features m_a, m_b, m_c are g_1, g_2, g_4 due to $\{abc\}' = \{124\}$.

3.1.3 Formal concept (concept)

A concept is a pair (A, B) where $A \subseteq G$, $B \subseteq M$, $A' = B$, $B' = A$. Sets A and B are called the *extent* and *intent* of a concept (Ganter and Wille 1999). Because of $(A')' = (B)' = A$, concepts are also closures under the concept-forming operations. $\mathcal{B}(G, M, I)$ denotes a universe containing all discovered concepts.

Each concept corresponds to an activity inference given observations. The intent refers to a superset of observed features (e.g. actions or sensor data), and the extent refers to probable target variables of interest (e.g. activities) given the intent. The property of closure ensures the internal semantic correlations between the extent and intent are maximized.

Example (124, ab) is a concept of context $\mathcal{K}$ in Fig. 2. We could assert that possible activities must be among g_1, g_2 and g_4 given m_a, m_b because only these three activities commonly possess these observed features.

3.1.4 Concept lattice (lattice)

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

Suppose that (A_1, B_1) and (A_2, B_2) are two concepts in $\mathcal{B}$, (A_1, B_1) is called the *subconcept* of (A_2, B_2) if either $A_1 \subseteq A_2$ or $B_2 \subseteq B_1$, written as $(A_1, B_1) \preceq (A_2, B_2)$. Meanwhile, (A_2, B_2) is the *superconcept* of (A_1, B_1). The symbol $\preceq$ is named as *hierarchical order*. The construction of

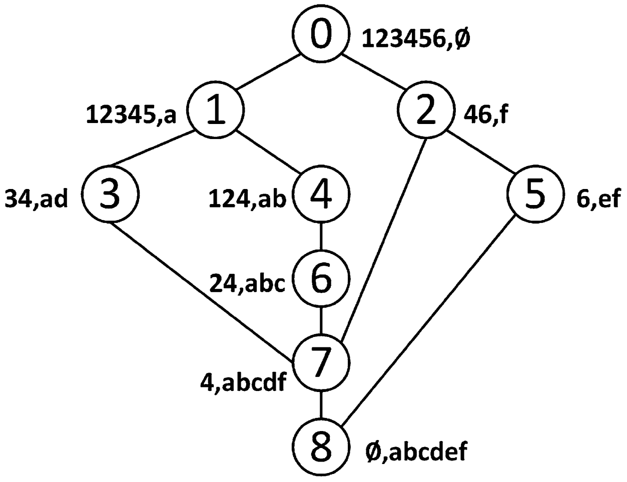


Fig. 3 Hasse diagram generated from the matrix in Fig. 2

lattice is a process that enumerates all the concepts by the concept-forming operations, and orders them by the order.

Example (124, *ab*) and (24, *abc*) are two concepts. Because of $\{ab\} \subset \{abc\}$, (24, *abc*) is one of the subconcepts of (124, *ab*), written as $(24, abc) \preceq (124, ab)$.

3.1.5 Hasse diagram

In mathematics, Hasse diagram is a graph depicting a finite partially ordered set. In our case, it is a visualization of lattice $(\mathcal{B}, \preceq)$ that represents component concepts as nodes (see Fig. 3).

There are two special nodes in a Hasse diagram: the topmost node $\{G, \emptyset\}$ named *Supremum* and the lowermost node $\{\emptyset, M\}$ named *Infimum*. Nodes are connected with edges named *Galois connection* which denotes the partial order $\preceq$ among nodes (Ganter and Wille 1999). As a graphical knowledge base filling up activity inferences, a Hasse diagram could organize and manage its nodes through its graph structure. For this reason, we try to design efficient algorithms to incrementally retrieve matching inferences given real-time observed data stream.

3.2 Real-time activity prediction and recognition

As mentioned in Sect. 3.1.3, each extent indicates the probable activities depending on the observed subset of the corresponding intent. Meanwhile, any intent denotes a superset of observed data in some moments. Like the concepts (124, *ab*) and (24, *abc*), if an intent is extended by new observed data (i.e. m_c), then the scope of probable activities will be reduced (from g_1, g_2, g_4 to g_2, g_4). This is the principle of our incremental activity inference engine.

In a Hasse diagram, concepts are ordered from top to down, in other words, from superconcepts to subconcepts. The Supremum $\{G, \emptyset\}$ is the initial state of the inference process because of no observed data. A token t is defined to locate the concept having the smallest intent in size containing all the observed data. In the process of activity recognition and prediction, with more and more observed data being added, token t will shift from top to down for locating the new matching concept. At the same time, the scope of probable activities becomes smaller and more accurate. When the extension of observed data suspends, current concept located by token t is the inference indicating the most probable predictions about the ongoing activity.

However, locating the concept having the smallest intent in size containing all the observed data in Hasse diagram is similar to a graph searching process. Thus, we transform the issue about activity prediction and recognition into a graph searching problem.

Suppose that α is an ordered list of observed data, denoted as $\alpha = \{\alpha_1 \prec \alpha_2 \prec \cdots \prec \alpha_m\}$, where $\alpha_j \prec \alpha_{j+1}$ means α_j occurs before α_{j+1}. In this paper, sequence α is also defined as an extensible cache successively loading the observed data. In this case, conventional graph traversal strategies could not perfectly satisfy our requirements about incrementally searching inferences. Most of them traverse the whole graph to locate their target, and could not recall or continue from their previous interrupted position in the next searches. When a new observed data is loaded into α, they have to restart for searching updated α in the graph and abandon all the previous searching results.

For these reasons, we propose a new incremental graph searching algorithm for locating target concepts in a Hasse diagram. Each new search could continue from the last interrupted position. Its advantages are not only reflected in the efficiency, but also on the consistency of inference.

The reason to locate the concept having the smallest intent in size containing all the observed data is that there is usually more than one satisfied concept containing α in a Hasse diagram. However, only the topmost superconcept could ensure the consistency of incremental inferences.

For example, in Fig. 3, if $\alpha = \{b\}$, then $n_4(124, ab)$ (i.e. the node 4) is the topmost superconcept we are looking for. If we choose another concept (e.g. $n_6(24, abc)$) instead of the topmost one, several possible activities will be lost (e.g. $\{124\} - \{24\} = \{1\}$) and could never be found in the next inferences (i.e. if $\{1\}$ is missed in n_4, it could never be found in n_6 and its subconcepts).

3.2.1 Graph searching algorithm

Breadth-first search (BFS) is one of the most common graph traversal algorithms. Its main idea is to explore all the neighbor nodes in the same level before moving to the next level. Because of successively expanded α, BFS is more efficient than another algorithm called depth-first search (DFS). However, just like the other algorithms, BFS itself could not well locate each topmost superconcept having a given α. For example, in Fig. 3, if $\alpha = \{abc\}$, then n_7 is always the first discovered concept having α because the path $n_0 \rightarrow n_2 \rightarrow n_7$ is shorter than $n_0 \rightarrow n_1 \rightarrow n_4 \rightarrow n_6$, but $n_7 \preceq n_6$.

Thus, on the basis of BFS, we propose a new half-duplex graph searching algorithm (HDGS) to locate each topmost superconcept. As can be seen from the name, HDGS consists of two searches. Firstly, the *top-down search* locates the first discovered concept containing α. And then, the *down-top search* turns back along the hierarchical order and looks for the topmost superconcept. More details about the HDGS algorithm are sketched in Pseudo-codes (1) and (2).

Algorithm 1: Top-down search of HDGS algorithm

```
Data: start position $sp$, sequence $\alpha$.
Result: first discovered concept containing $\alpha$.
begin
    fifo ← node[sp]
    while fifo do
        if fifo[0] not visited then
            mark it 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
```

Algorithm 2: Down-top search of HDGS algorithm

```
   Data: first located position $sp$, set $\alpha$.
   Result: topmost superconcept containing $\alpha$.
1  begin
2      fifo ← node[sp].predecessors
3      $S \leftarrow \varnothing$
4      while fifo do
5          if fifo[0] not visited then
6              mark it as visited
7              if $\alpha \subseteq$ fifo[0].intent then
8                  add fifo[0].predecessors into fifo
9                  $S \leftarrow S \cup$ fifo[0]
10         remove fifo[0] from fifo
11     end
12     return $\arg\min_{s \in S}(|\, s.intents \,|)$
13 end
```

In Algorithm 2, we need to pay attention to Line (12) which seeks the topmost superconcept having the minimal cardinality of intent containing α. Dues to the hierarchical order defined in Sect. 3.1.4, we could see that the intent of the topmost superconcept has smaller cardinality than the others containing α.

3.2.2 *Retrieval strategy*

Beyond the issue of locating the topmost superconcept given α, we also need to consider another tough issue about multilevel inheritance. It is a very common situation existing in the AmI context due to diverse lifestyles and personal habits. An activity could be performed by alternative ways like adding or omitting optional actions or sensor events (Van Kasteren et al. 2011). Thus, there are multilevel inheritance relations among these derived activities. For instance, *PrepareCoffee*(A_4) and another three derived activities about preparing coffee: *PrepareBlackCoffee*(A_1), *PrepareCoffeeWithoutSugar*(A_2) and *PrepareCoffeeWithoutMilk*(A_3), have the multilevel inheritance relations as $A_1 \subset A_2 \subset A_3 \subset A_4$.

Therefore, the retrieval strategy that we adopted is based on the *greedy* way. That is, if an activity is recognized, its completeness will also be verified until all the necessary actions or sensor 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, which means all probable activities including the ones with multilevel inheritance are recognized in previous extensions.

3.3 Prediction assessment

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

As mentioned in the previous sections, an activity g_i could be performed by alternative patterns because of different personal preferences. Furthermore, these derived patterns reflect on flexible execution orders, repetitive data, adding or omitting optional data, etc. At the same time, each person also has his/her own relatively stable preference to execute the same activity. Namely, for the same occupant 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 an occupant to generate a knowledge database called *accumulated matrix*. For each action or sensor event, we calculate its expectant position appearing in each activity to establish a series of naive distributions.

Table 1 Comparison between ontology and formal concept analysis

Ontology	Formal concept analysis	Reality
Individuals	Objects	Entities of interest
Classes	Concepts	Semantic definitions
Attributes	Attributes	Descriptors and properties of different natures

To measure the similarities between historical patterns and current incomplete, ongoing activity, average deviations are calculated using root-mean-square deviation (RMSD). It makes a quantitative comparison as an assessment to estimate how well the current pattern fits historical data. A lower RMSD score of a possible activity indicates that the activity is more probable to be done due to excellently fitting with historical patterns.

We propose our assessment as follows: for each possible activity 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 most relevant prediction should be the activity with the minimal deviation which has the best fitting in comparison with historical data. Obviously, our assessment consists of two modules: accumulation and candidate evaluation.

3.3.1 Accumulation

For each α_j in a training item α describing activity g_i (i.e. $\alpha_j \in \alpha$), we update the accumulated positional value of the corresponding element (g_i, α_j) in the accumulated matrix by Eq. (3):

$$\sigma_{ij} = \sigma'_{ij} + j \tag{3}$$

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 (4) represents the same accumulation in the global view:

$$\sigma_{ij} = \sum_{k=1}^{N_{ij}} \sigma_{(ij,k)} \tag{4}$$

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 activity g_i.

3.3.2 Candidate evaluation

When an incoming action or sensor event α_j was observed, first of all, we calculate its average position $\overline{\varphi_j}$ in current sequence α. It is calculated by Eq. (5).

$$\overline{\varphi_j} = \frac{1}{\#\alpha_j} \sum_{k=1}^{|\alpha|} k, \quad \text{if } \alpha_k = \alpha_j \tag{5}$$

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 activity g_i, we calculate the deviation of α given g_i. Equation (6) expresses the root-mean-square deviation D_i of current sequence α executing g_i:

$$D_i = \sqrt{\frac{1}{|\alpha|} \sum_{\forall \alpha_j \in \alpha} \left(\overline{\varphi_j} - \frac{1}{N_{ij}} \sigma_{ij} \right)^2} \tag{6}$$

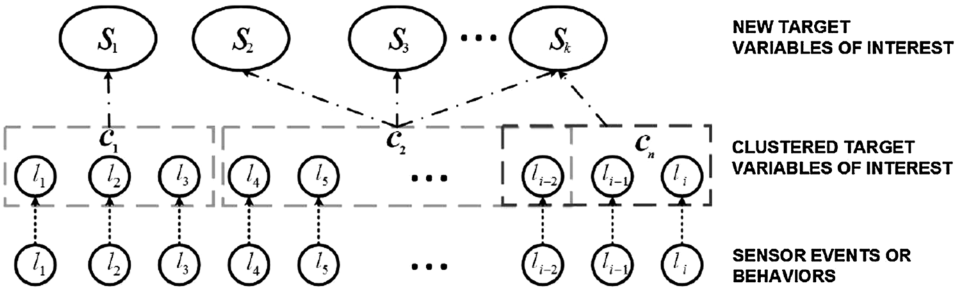
where σ_{ij}/N_{ij} is the expectant position obtained from accumulated matrix.

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

4 Ontological clustering

At the beginning of an activity execution, predictions are less accurate than the other stages for the reason of few observed data. Moreover, some semantically similar activities with almost the same subsequences, especially the ones having multilevel inheritance relations, could confuse the candidate predictions in the early stages.

The purpose of this section is to automatically create an alternative level on the basis of the architecture in Fig. 1



S_1
S_2
S_3
S_k
NEW TARGET VARIABLES OF INTEREST
C_1
C_2
C_n
l_1 l_2 l_3 l_4 l_5 l_{i-2} l_{i-1} l_i
CLUSTERED TARGET VARIABLES OF INTEREST
l_1 l_2 l_3 l_4 l_5 l_{i-2} l_{i-1} l_i
SENSOR EVENTS OR BEHAVIORS

Fig. 4 Alternative level created by ontological clustering

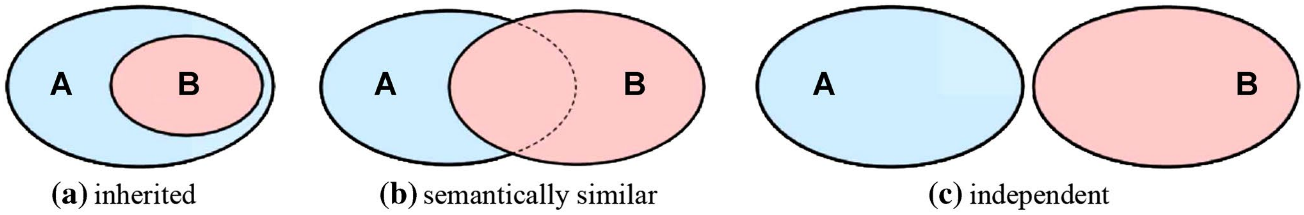


Fig. 5 Semantic relations between two activities

for integrating similar target variables of interest, reducing semantic gaps between two-level data, and enhancing data interpretation. Figure 4 illustrates such a structure: the intermediate level is an alternative abstraction of clustered target variables of interest.

In Sect. 3.2, we concluded that the fewer data about the ongoing activity were observed, the more ambiguous candidates there are in the located concept. Instead of forcibly predicting activities by few observed data in the early stages, approximate predictions about the ongoing activity will be more useful in our case.

For example, if there are three observed actions, BoilWater, TakeOutSpoon and TakeOutMilk, it is difficult to precisely predict which one is being done, maybe PrepareCoffee or PrepareMilkTea. However, we could at least determine that the ongoing activity is related to preparing something to drink. Thus, the system would pay more attention to the cognitive assistance and preventive interventions about preparing something to drink, rather than the ones about preparing something to eat.

As a potential solution, our objective is to cluster target variables of interest according to their semantic similarities. Each new cluster is a semantic definition that could be renamed on the basis of their common semantic features. Formica (2006) has demonstrated that there are some shared characters between ontology and FCA theories (see Table 1). Consequently, we propose an ontological clustering based on FCA to ameliorate our predictions in the early stages.

4.1 Ontological similarity metric

To generate ontological clusters, firstly, we need to define a metric to evaluate semantic similarities among target variables of interest. As described in Fig. 5, there are three possible semantic relations between two objects, which are related to the amount of shared features.

Suppose that A and B are two objects. The first relation is called *inherited*. It is true if and only if an object is a subset of another one. In Fig. 5a, A contains all the features of B, referred as $B \subset A$, called A is inherited from B. This relation is very common in reality due to multilevel inheritance relations caused by diverse living habits and personal preferences. For instance, PrepareCoffeeWithSugar (A_0) is inherited from PrepareBlackCoffee (A_1) because of $A_1 \subset A_0$.

The second one is called *semantically similar*. It is true if and only if two objects have partial common parts among their features. In Fig. 5b, A and B have some partial intersection, referred as $A \cap B \neq \emptyset$. No matter how few the common features are, two semantically similar objects have always semantic similarity.

The third one is called *independent*, which means that two objects are mutually independent. In Fig. 5c, A has no common feature shared with B, referred as $A \cap B = \emptyset$.

Due to the limitation of share features, some newly clustered target variables of interest could not be easily renamed, but it will not hinder their generation. The construction of ontological clusters is the process enumerating the objects mutually having inherited or semantically similar relations.

4.2 Ontological clustering by FCA

There are a wide variety of methods that could be used to address the clustering problems. The objective is to maximize the similarity of objects in a cluster and simultaneously maximize the dissimilarity among clusters. Distance-based and density-based algorithms are the two most common categories, especially the distance-based one. The former is desirable because of the simplicity and ease of implementation in a wide variety of scenarios (Aggarwal and Reddy 2013).

Each clustered target variable has inherited or semantically similar relations with others. Like classical distance-based clustering algorithms (Leskovec et al. 2014), in the final clusters, ontological clustering is also required to find out the clustroids which are the closest on average to the other objects in their clusters. In practice, these clustroids are the common shared features of those objects. However, there are also some special differences. One of them is that objects from different clusters are relatively dissimilar, which means there are overlaps among clusters of target variables.

Our ontological clustering further discovers the target variables of interest having inherited or semantically similar relations on the basis of the current Hasse diagram. The process of ontological clustering based on the FCA could be summarized as follows:

1. Select relevant features (attributes) and prune the noisy or irrelevant ones (Aggarwal and Reddy 2013).
2. Initially define each indexed target variable of interest as an independent cluster by itself.
3. Define a metric to measure similarity.
4. According to the predefined minimal threshold of ontological similarity, repeatedly merge two nearest clusters into one (see Algorithm 3).

Algorithm 3: ontological clustering algorithm

```
   Data: start position sp, Hasse diagram diag, threshold $t_1$.
   Result: topmost superconcept containing $\alpha$.
1  begin
2      fifo ← diag[sp].successors
3      S ← ∅
4      while fifo do
5          if fifo[0] not visited then
6              mark as visited
7          if fifo[0].extent.len < fifo[0].children.extents.len then
8              cluster ← fifo[0].extent
9          similar ← True
10         foreach o in fifo[0].extent do
11             $n_0$ ← fifo[0].intent.len
12             N ← o′ .len
13             if $n_0/N < t_1$ then
14                 similar ← False
15         end
16         if similar then
17             cluster ← fifo[0].extent
18         remove fifo[0] from fifo
19         clusters.add(cluster)
20     end
21     return clusters
22 end
```

In our clustering algorithm, objects g_i in a cluster $A \subset G$ share the same attributes (clustroid). In other words, all the objects sharing the same clustroid should be merged in a cluster. The cardinality of clustroid should be greater than the predefined threshold t_0 (see Eq. (7)).

$$\left|\bigcap_{i=1}^{n} g_i'\right| > t_0, \quad g_i \in A \subset G \tag{7}$$

where g_i' are the attributes of g_i obtained by the concept-forming operation defined in Sect. 3.1.2.

Furthermore, the merger based on a fixed threshold is not sufficient due to various cardinalities of clustroids in different clusters. Thus, the percentage threshold should be better to evaluate the ontological similarities in different clusters. On the basis of Eq. (7), we propose another metric as:

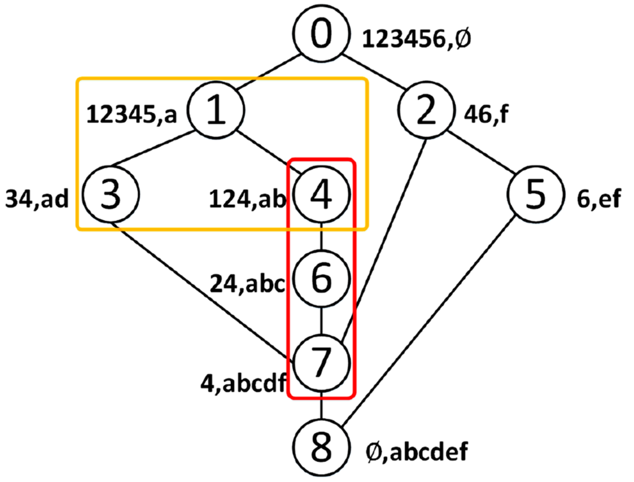


Fig. 6 Clusters in a Hasse diagram

$$\frac{\left|\bigcap_{i=1}^{n} g_i'\right|}{\max\left|g_i'\right|} > t_1, \quad g_i \in A \subset G \tag{8}$$

where the numerator is the commonly shared attributes among internal objects, which is also the clustroid of a cluster. The denominator is the cardinality of the maximal set of observed attributes among sequences describing g_i.

In fact, Eq. (7) is as same as the definition of the concept-forming operation (1). As a consequence, every concept in a Hasse diagram is an ontological cluster with a dynamic threshold.

If the process of ontological clustering is based on the semantic relations described in Fig. 5, to repeatedly merge two nearest clusters into one, there will be two mechanisms to generate clusters. The process is to traverse the whole Hasse diagram to find out all the concepts having corresponding semantic relations.

The first one is to discover inherited relations shown in Fig. 5a. The main character is that some objects in the extent of one concept can not be found in the extents of its subconcepts. It refers to Lines 7–9 and 16–19 in Algorithm 3.

Example In Fig. 6, the red rectangle including nodes 4, 6, and 7 highlights the inherited relation. Object g_1 in node 4 disappears in the extents of the sub nodes 6 and 7. This is because the disappeared objects are the superclasses having less attributes than the subclasses in the sub nodes.

The second one is based on the semantically similar relation in Fig. 5b. If one node has more than one branch, it means that the objects in its extent are the clustroids and current concept is an ontological cluster.

Nevertheless, it is necessary to use the threshold defined in Eq. (8) to control the merging of clusters. It refers to Lines 10–19 in Algorithm 3.

Example In Fig. 6, the yellow rectangle including nodes 1, 3 and 4 highlights the semantically similar relation. Objects in nodes 3 and 4 commonly having an attribute *a*. If the cardinality of the intent in node 1 is bigger than the predefined threshold, the following sub nodes should be merged.

With the help of ontological clustering, the prediction accuracies at the early stages will be improved. When observed data are few and limited, the inference engine will predict the ontological superclass instead of directly predicting an activity. For example, *PrepareCoffee* will be no longer directly predicted, the inference trace will be *PrepareDrinks→PrepareBlackCoffee→PrepareCoffee-WithoutSugar→PrepareCoffee*.

5 Anomaly detection

Due to physical or cognitive impairment, disabled or elderly people, like patients with Alzheimer's disease or traumatic brain injury, have difficulties in performing self-care tasks on their own. The sporadic memory loss has frequently occurred when performing an activity (Roy et al. 2011). Thus, they tend to produce more abnormal behaviors than healthy people. The solutions could be classified as anomaly detection of sequential patterns (Chandola et al. 2009). In this section, we summarize common abnormal behavioral patterns and discuss how to detect corresponding cognitive errors based on their behavioral features.

5.1 Problem settings

Because of varied living habits or other external factors, an activity could be described using diverse behavioral patterns having different optional features. Even if having the same sets of features, two patterns could be different due to various execution orders, repetitive actions or sensor events. Thus, any activity could possess N_i derived patterns having j different sets of features ($N_i \gg j$). Before introducing our propositions, we formally define a sequence of actions or sensor events captured by the sensor network in smart environments.

In order to specify each abnormal behavioral pattern, we define that a sequence α_j describing an activity g_i should be a union (not a set) of:

- essential set $E_i = \bigcap_{i=1}^{N_i} \alpha_i$, which contains all essential actions or events existing in all N_i patterns describing g_i. For example, *boil water* and *pour water into a teacup* are two essential actions for *PrepareMilkTea*, because they exist in any sequence α_i describing the process of making a cup of milk tea, no matter who does it.
- optional set $O_i = \bigcup_{i=1}^{N_i} \alpha_i - \bigcap_{i=1}^{N_i} \alpha_i$, which indicates all the optional actions or events of g_i. For example, *add sugar* could be somebody's personal taste when drinking milk tea, but it does not exist in all the patterns describing *PrepareMilkTea*, so it is a typical optional one.
- possible irrelevant set I_i that $I_i \cap \bigcup_{i=1}^{N_i} \alpha_i = \emptyset$. For example, *take out pasta from cabinet* is an irrelevant action for *PrepareMilkTea* and it should not exist in any normal pattern describing g_i.
- possible redundant set R_i that $R_i \subseteq \bigcup_{i=1}^{N_i} \alpha_i$, which contains any indexed action or event existing in entire N_i patterns of g_i.

The generic symbolic representation of sequence α_j is given out in the form of a triplet (see Eq. (9)):

$$\alpha_j = (\{E_i \cup O_i' \cup I_i' \cup R_i'\}, \preceq_j, C_i) \tag{9}$$

where $O_i' \subseteq O_i$, $I_i' \subseteq I_i$, $R_i' \subseteq R_i$. In particular, $\preceq_j$ defines a permutation of the union (i.e. a possible execution order). C_i is a set of causal constraints restricting the permutation $\preceq_j$. Thus, we could assert that α_j is a normal pattern without errors if and only if set E_i is complete, sets I_i' and R_i' are empty, and $\preceq_j$ satisfies all the constraints in C_i.

From the definition above, we could find that each component in Eq. (9) plays a key role in the abnormal behavioral patterns. In the following words, we present how to detect anomalies using our inference engine.

5.2 Abnormal patterns

By observing and tracking the daily lives of people who cannot live independently, first of all, we define abnormal features appearing in their behavioral patterns. Then, on the basis of pattern analysis, we give out costumer-built solutions to address corresponding cognitive errors.

5.2.1 Initialization

Initialization error is related to the short-term memory loss. The typical behavioral feature is about doing nothing at the beginning phase while performing an activity. The simplest solution is to set a temporal threshold to detect whether an occupant starts to do something at the early stages. In this paper, initialization error will not be considered in the following parts of evaluation and discussion.

5.2.2 *Omission of essential data*

An omission is a failure to do something that ought to be done, but was forgotten, according to the initial planning. It is very common in daily life, even to healthy persons. Sometimes, there are only limited impacts for performing an activity. For example, there is no big deal if somebody forgets to do some actions or to trigger some sensor events in the optional set O such as personal preferences. However, in most of the time, the omission of essential data will disrupt the integrity of the implementation (e.g. forget to add some ingredients while cooking) and affect the quality of accomplishment. In some extreme cases, it will lead serious or fatal consequences (e.g. forget to turn off the oven after using it).

As we analyzed above, the optional elements in set O are less important than the ones in set E, and bring fewer troubles while being forgotten. Thanks to the set-based dual structure of concepts, it is easy to check the level of completion using set theory: if the universal actions or sensor events of an activity g_i is denoted as g_i', the forgotten ones could be calculated as the relative complement $S^C = g_i' - S$, where S is currently observed data.

Example Suppose that $\alpha = \{a \prec c \prec b \prec f\}$ is successively observed. Considering Fig. 3, node 7 is finally located after the extensions. To check the completion of g_4, we compare α with $g_4' = \{abcdf\}$, and the relative complement $g_4' - \alpha = \{d\}$ is not equal to an empty set, so element d is omitted while executing g_4.

5.2.3 *Irrational repetition*

Redundancy in data streams could be caused by miscellaneous reasons: periodic sampling, noisy data, rational or irrational repetition etc. In most cases, rational repetition is harmless, even necessary to accomplish an activity. For example, we need to regularly check the degree of cooking or intermittently stir food ingredients while preparing a meal. In the other extreme cases, irrational repetition will lead to potential threats like excessive consumption (condiments or medications).

The simplest solution is to check whether an incoming observed data exists or not in the current sequence α. To distinguish rational and irrational repetitions, we predefine a weighted matrix to measure the harm level of each feature being repetitive for each indexed activities. As a result, the sensibility of harmful repetitions could be reinforced and the false-positive alerts warning the harmless ones could be weakened.

5.2.4 *Mixture of irrelevant data*

People with cognitive impairment often forget current plan or confuse with another one, and then add irrelevant data into current ongoing activity. From Eq. (9), we could see that irrelevant data set I_i of activity g_i has no intersection with the relevant one $E_i \cup O_i$. In other words, an extension a_j is compatible with current plan if and only if $a_j \in E_i \cup O_i$. Thus, elements in I_i will be excluded.

After a new extension, if updated α can no longer match any concept except the Infimum, it means that one or more irrelevant observed data have mixed into the current sequence, notably the last incoming one should be suspected.

Example Considering Fig. 3, suppose that sequence α is successively extended by $\{a \prec c \prec e \prec d \prec b \prec f\}$. Node 6 is located after the first two extensions $\alpha \leftarrow ac$. In the third round, $\alpha \leftarrow e$, updated $\alpha = \{ace\}$ is incompatible with current plan because there is no subconcept (A, B) having $\alpha \subseteq B$ except the Infimum. As a consequence, last incoming e will be treated as an irrelevant data which has to be removed from current plan and put it aside, into a new cache indicating another plan. At the end of the extensions, node 7 is located and the irrelevant data e is identified.

We summarize the logic above and represent it as pseudocode in Algorithm 4. Cache P_0 always denotes the initial plan of an occupant. A new incoming observed data a is loaded for extension at step 3. Steps 4–7 is to check whether there exists one or more caches in P_i that are compatible with current observed data. If incoming data a is irrelevant to all existing caches (step 9), then create a new cache to save it (steps 10–11). After extensions, we choose the longest cache, P_0 in most of time, as the normal sequence performing g_i (step 12), and the elements in the other caches will be treated as irrelevant data or sensor events.

Algorithm 4: detect irrelevant data

```
     Data: sequence $\alpha$, lattice $\mathcal{L}$, caches $P_i$.
     Result: set of irrelevant data $\mathcal{I}$.
 1   begin
 2       while $\alpha$ do
 3           $a \leftarrow \alpha.popleft$
 4           foreach $P_i$ do
 5               if $\exists(A,B) \in \mathcal{L},\ P_i \cup a \subseteq B$ then
 6                   $P_i \leftarrow P_i \cup a$
 7           end
 8       end
 9       if $\nexists(A,B) \in \mathcal{L},\ P_i \cup a \subseteq B$ then
10           $P_{i+1} \leftarrow a$
11           $P_i \leftarrow P_i + P_{i+1}$
12       $P_M \leftarrow max(size(P_i))$
13   end
```

Fig. 7 Causal matrix of $|M|$ indexed features

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1M} \\ c_{21} & c_{22} & \cdots & c_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \cdots & c_{MM} \end{bmatrix}$$

5.2.5 *Causal conflict*

Suppose that $\alpha_i \prec \alpha_{i+m}$ successively appear in the sequence $\alpha = \{\alpha_0 \prec \cdots \prec \alpha_i \prec \cdots \prec \alpha_{i+m} \prec \cdots \prec \alpha_n\}$. If causal constraint set C has limited that α_{i+m} must occur before α_i, represented as $\alpha_{i+m} \preceq \alpha_i$, then there is a causal conflict in the sequence.

In this paper, we automatically generate causal constraints and verify the causalities among elements in a sequence α. For any α in the training set, first of all, we generate the causal pairs by scanning α and accumulate them into a square matrix (see Fig. 7). If α_i appears before α_j, then $\alpha_i \prec \alpha_j$ and the occurrence at c_{ij} will add 1.

And then, in the constraint generation phase, if $c_{ij} \times c_{ji} \neq 0$ and $c_{ij} \gg c_{ji}$, then $\alpha_i \prec \alpha_j$ is a relative causal constraint, and vice versa. If $c_{ij} \times c_{ji} = 0$ and $c_{ij} \neq 0$, then $\alpha_i \prec \alpha_j$ is an absolute one, and vice versa. At last, in the test phase, if $\alpha_i \prec \alpha_j$ is against the predefined causalities, there is a causal conflict in the ongoing execution.

5.2.6 *Cognitive distraction*

Cognitive distraction is similar to the mixture of irrelevant data. Compared to the initial plan, both of them have the same character that mixed irrelevant data into their patterns, but cognitive distraction has created a transformation of quantitative into qualitative changes. Cognitive distraction could be classified as a collective anomaly (Chandola et al. 2009). At the early stages, observed data belong to the real expected long-term plan. However, at a specific singular point, observed data start to unconsciously distract from the initial plan and turn to another unwilling one.

Figure 8 is an example of cognitive distraction. Plan 0 is used to indicate the initial plan of an occupant. Plan 1 and 2 denote the distracted traces. A black point indicates that the newly observed data at this step is accepted by the current plan. Meanwhile, a white one indicates a rejection.

Example As shown in Fig. 8, a cognitive distraction happened in the fourth extension and T_1 indicates the singular position. Observed data a_4 has not been accepted by Plan 1 due to its irrelevance. Once an action or sensor event is not accepted by all the existing caches, it will be put into a new one. Moreover, if an action or sensor event is compatible with more than one cache, it must be distributed into each compatible cache. At the end of the extensions, we choose the longest cache as the normal pattern. If the chosen cache is not Plan 0, we could assert that the occupant has distracted from the real objective.

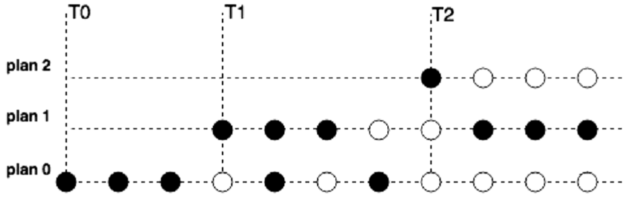


Fig. 8 Cognitive distraction happening at T_1

6 Experiments

In this section, we present our experimental results of the activity inference engine in the parts of the prediction, recognition and anomaly detection. It is worth mentioning that all the experiments are carried out on the computer with tech specs of Intel Core i7 Processor 2.4GHz and 8GB RAM, under Ubuntu 14.04.

6.1 Experimental datasets

As the extension of our previous work (Hao et al. 2016a, b), the experiments are based on the same labeled datasets. The first three ones named *RDATA*, *DDATA* and *EDATA* are created by our LIARA laboratory, which describe the correlations between actions and ADLs. Moreover, *DDATA* is the synthetic dataset created from *RDATA*. The patterns in *RDATA* and *DDATA* are normal discrete sequences without abnormal behaviors. In *EDATA*, all kinds of abnormal patterns described in Sect. 5 except initialization are included. Twelve ADLs (see Table 2) are described by sequentially observed actions (e.g. Table 3). Due to more complicated scenarios and environmental interactions, we choose kitchen activities as our research objects.

As another two benchmark datasets, *Kyoto-1* and *Kyoto-2*,[2] are constructed by the CASAS laboratory of Washington State University (Cook and Schmitter-Edgecombe 2009). Both of them describe the same correlations between sensor events and activities, but the latter also contains abnormal patterns in the data stream. In the anomaly detection task, we use the former dataset to train our inference engine and test the detection efficiency in the latter one.

[2] Available at http://casas.wsu.edu/datasets/.

Table 2 Statistical information about RDATA

Activities	No. actions
PrepareCoffee	14
PrepareCoffeeWithoutSugar	11
PrepareCoffeeWithoutMilk	11
PrepareMilk	5
PrepareSpaghetti	18
PrepareSandwich	15
PrepareSandwichWithoutMustard	11
PrepareSandwichWithoutButter	9
PrepareCereal	8
PreparingToastsAndEggs	20
PreparePudding	5
PrepareMilkTea	12

Table 3 Data structure of training items

TI	Atomic actions	Activities
1	BoilWater	PrepareCoffee
2	TakeCupFromCupboard	PrepareCoffee
3	TakeOutCoffee	PrepareCoffee
4	PutCoffeeIntoCup	PrepareCoffee
5	StoreCoffee	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

6.2 Evaluations

The following evaluations focus on four aspects: modeling, prediction and recognition accuracies and detection efficiency. All these criteria are separately discussed according to the different data features of LIARA and CASAS.

Furthermore, all the datasets were randomly divided into 10 subsets, and such division was repeated 10 times. Each time, one of the subsets 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 (Aggarwal 2014). A model sometimes could receive excellent evaluations on the data existing in the training set, but once the test data has not been seen before, the classification result will be broken down. As an evaluation method, 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. As a consequence, each pattern in the dataset was removed at least once from training sets.

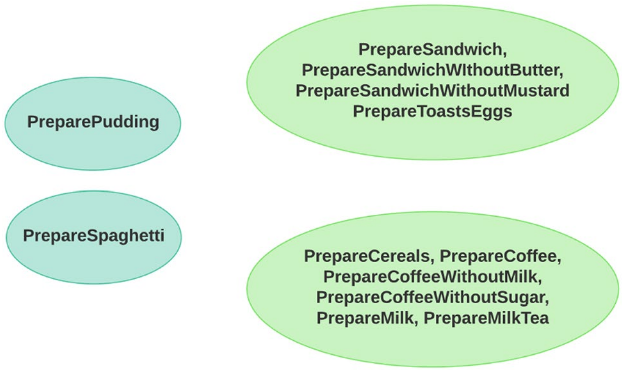

PreparePudding
PrepareSpaghetti
PrepareSandwich, PrepareSandwichWIthoutButter, PrepareSandwichWithoutMustard PrepareToastsEggs
PrepareCereals, PrepareCoffee, PrepareCoffeeWithoutMilk, PrepareCoffeeWithoutSugar, PrepareMilk, PrepareMilkTea

Fig. 9 Ontological clusters of LIARA dataset

6.2.1 *LIARA datasets*

The actions steam came from our previous works (Fortin-Simard et al. 2015; Belley et al. 2015; Halle et al. 2016). The majority of actions, relating the usages of electrical appliances and interactions with daily commodities, were obtained by signal analysis (passive RFID, electrical, etc.). A few actions are also identified by heterogeneous sensors (e.g. binary sensor detecting the open/close states of a cabinet, water sensor measuring the use of water, burner sensor controlling the use of the burner, etc.).

After the ontological clustering, twelve activities are classified into four clusters (see Fig. 9). Two clusters indicating "PrepareSomethingToDrink" and "PrepareSomethingToEat" are generated. Another two small clusters indicate two individual activities because of less similarity with the others.

The results of prediction and recognition are illustrated in Table 4, Figs. 10 and 11. As shown in Table 4, our inference engine has perfect recognition rates.

6.2.2 *CASAS datasets*

Different with LIARA datasets, CASAS ones focus on the recognition using low level sensor events. Heterogeneous sensors, including motion sensors, item sensors, phone usage sensors, water and burner sensors, are distributed around the target space as shown in Fig. 12.

The mapping from low level sensor events to high level activities is indeed more flexible in modeling by bypassing the intermediate level actions. However, sensor event

Table 4 Accuracy of activity recognition

Dataset	No. items	Without clustering (%)	With clustering (%)
RDATA	240	100	100
DDATA	96,972	100	100
Kyoto-1	120	86.7	86.7

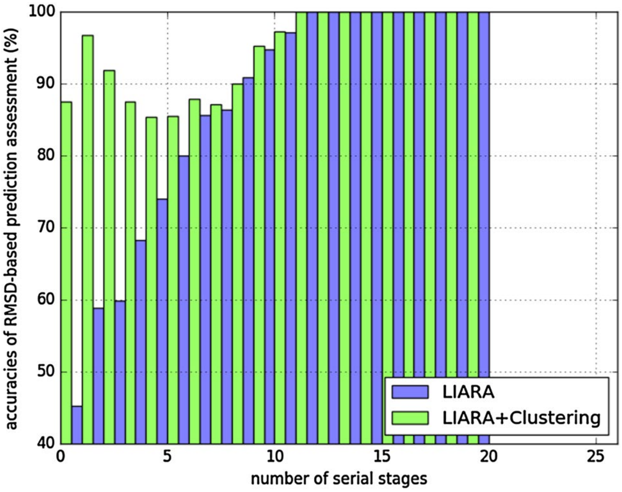


Fig. 10 Comparison of LIARA recognition results

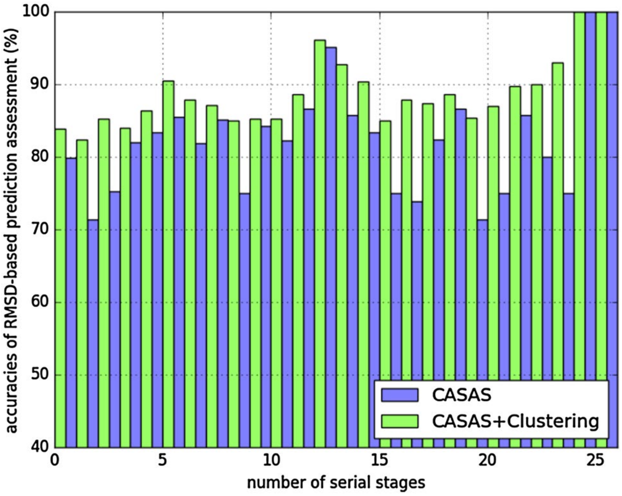


Fig. 11 Comparison of CASAS recognition results

streams are sometimes not intelligible due to the large semantic gap with complex activities Without extra interpretation, it is difficult to understand the meanings behind observed sensor events and corresponding inferences.

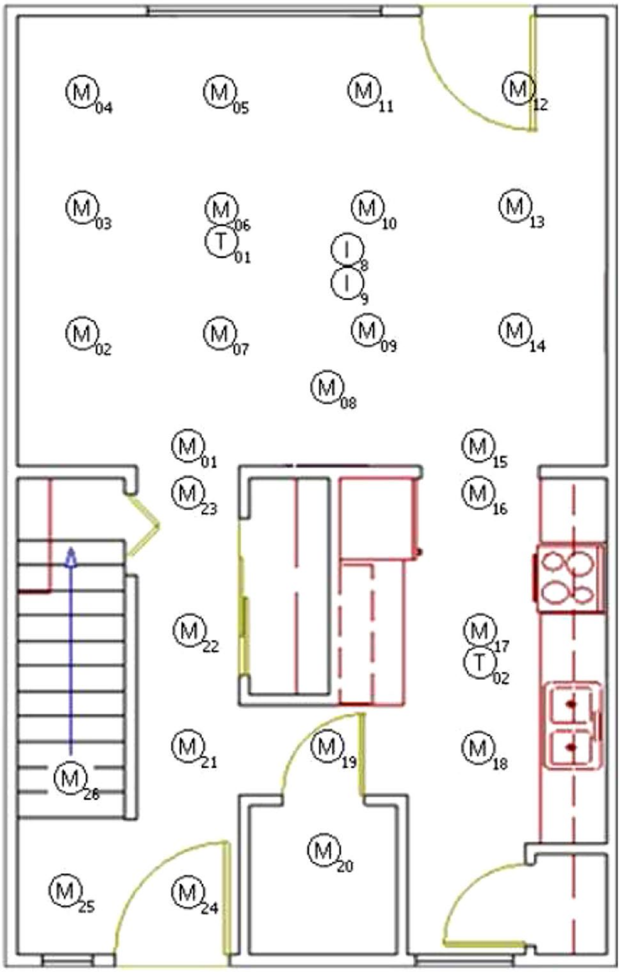


Fig. 12 Sensor layout of CASAS apartment

Similar to LIARA items, a CASAS item also has three fields: timestamps, sensor identifiers and states. The values of sensor states are either binary or numerical. To simplify the knowledge modeling, in the current design, we have not considered the influence of temporal intervals between sensors. Thus, we ignored the timestamps and only kept the successive orders. For the numerical states, we also transferred them into nominal forms to adapt to the construction mechanism of FCA lattice.

While the motion sensors could not directly provide information to distinguish which activities produced the movements (Cook et al. 2009), in the ontological clustering, we classified activities based on the spatial areas defined by motion sensors. The clustering results are shown in Fig. 13.

As shown in Fig. 13, we could see that most of homogeneous activities are performed in similar areas. Because an occupant has to approach to the specific positions to interact with objects.

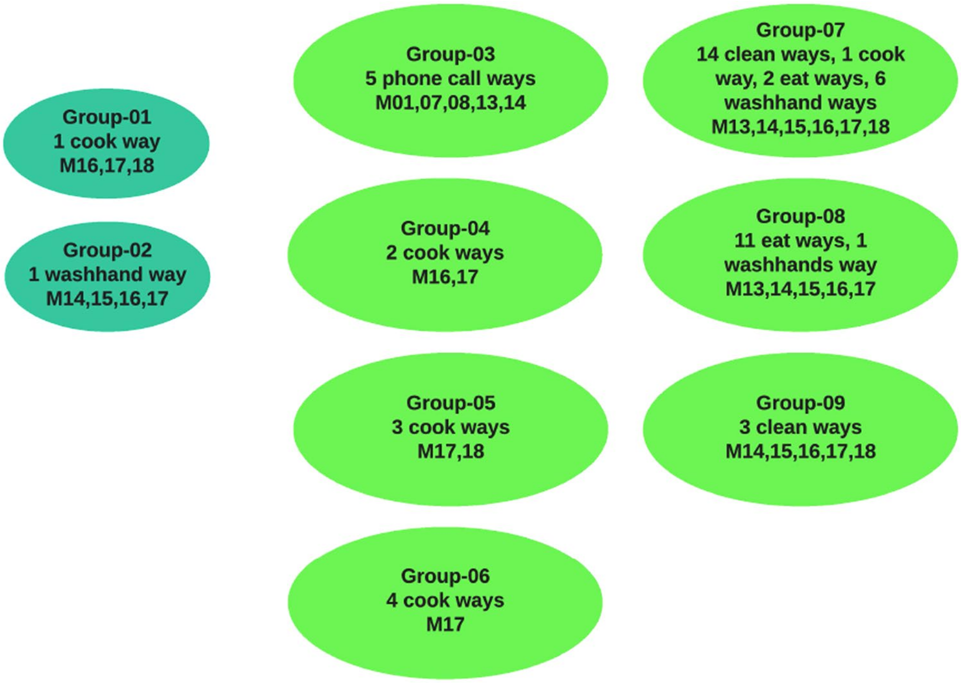


Fig. 13 Ontological clusters of CASAS dataset

6.3 Anomaly detection

Table 5 sketches the comparison about abnormal pattern detection applied on *EDATA* and *Kyoto-2* datasets. From the listed results in Table 5, we could see that our engine received excellent detection rates for four patterns except the cognitive distraction. The accuracy about detecting cognitive distraction strongly depends on the singular position when the distraction occurs. It is worth mentioning that the result of causal conflict detection was based on the manually defined causal constraints (marked as "M").

For *CASAS*, there are only two predefined abnormal patterns existing in the test items: omission[3] and repetition.[4] In Table 5, we used "–" to represent nonexistent results. Furthermore, we evaluated the performances under the chosen evaluation metrics in Table 6, including precision, recall and F-measure.

7 Discussions

In this section, we discuss the results shown in Sect. 6 and analyze the advantages and disadvantages of our FCA-based inference engine.

Table 6 Anomaly detection in CASAS dataset

Abnormal patterns	Precision	Recall	F-score
Omission	0.656250	1.0	0.792453
Repetition	1.0	1.0	1.0

In the aspect of generalization, we could see that the FCA-based method has received high recognition rates while handling LIARA datasets. One reason is that we have chosen a more suitable granularity in the multilevel structure defined in Fig. 1. Compared to sensor events, activities have strong semantic dependencies with their component actions. Different actions have different abilities to distinguish activities. For any activity, its essential actions are more distinguishable than the optional ones. For instance, action *TakeOutCoffee* is more distinguishable than *TakeOutMilk* in the discrimination between *PrepareCoffee* and *PrepareCereal*. Strong semantic dependencies assume that

Table 5 Anomaly detection in EDATA dataset

Abnormal patterns	Datasets/ACC	
	EDATA	CASAS
Omission of essential data	100%	88.5%
Mixture of irrelevant data	100%	–
Irrational repetition	100%	100%
Causal conflict	100% (M)	–
Cognitive distraction	≥97.8%	–

[3] Did not turn the water off, did not turn the burner off, did not bring the medicine container, did not use water to clean and did not dial a phone number.

[4] Dialed a wrong phone number and redialed, duplicate sampling of motion sensors, etc.

Table 7 Activity recognition in different scenarios

Existence of element a_i	a_i is essential	a_i is alternative
Not available in the training set, but in the test set	Activity g_j could be normally recognized, but a_i will be identified as an irrelevant element of g_j	Activity g_j could be normally recognized, but a_i will be identified as an irrelevant element of g_j
Available in the training set, but not in the test set	Activity g_j could be normally recognized, but a_i will be identified as an omitted element of g_j	Activity g_j could be normally recognized, but g_j is in fact a derived activity of the recognized one
Available in both training and test sets	Activity g_j could be normally recognized	Activity g_j could be normally recognized

even if some observed data have been seen in the patterns of training items, the target activity may also be recognized by partial highly distinguishable essential data.

As shown in Table 7, we analyze the reasons behind high recognition rates by each scenario. Because the support of set theory, FCA-based inference engine is sensitive to detect the component difference between the test items and the historical ones in the training dataset. Unseen patterns describing derived activities could be normally recognized, but sometimes they will be treated as normal activities having irrelevant actions. In another situation, unseen patterns having fewer actions than the historical ones could also be recognized as subsequences of normal ones, which means incomplete ones.

In LIARA dataset, each activity has strong semantic dependency with its component actions. Theoretically, there will not have two highly similar activities which have not inherited relations in semantics. As a consequence, we could perfectly distinguish activities by the actions having strong discrimination. For CASAS datasets, the situation is more complicated. Two different activities in semantics could be highly similar, even have the same pattern due to the weak semantic dependency between numerical values and activities. Most of discrete sensor states are weak to distinguish activities. So it is natural to spontaneously form highly similar or even the same patterns without strong semantic relations. As a disadvantage of our inference model, Hasse diagram could not directly consider temporal information as one part of the knowledge base. Thus, we could not make use of temporal information to help us distinguish patterns and depended only on the alternative assessment based on the RMSD to indirectly evaluate the context similarity between current pattern and historical ones.

Here is an extreme example: suppose that we only install three motion sensors in the kitchen to identify two activities. Due to the limited amount of sensors, there is a very high probability that two activities have the same sensor event pattern. In this case, activities have weak correlations with the sensors and it is extremely difficult to distinguish two activities depending on the weak dependencies having semantic gaps. For the suitable mapping between actions and affiliated activities, because of strong semantic correlations, there are rare activities having the same set of performed actions with others. Thus, the results of models with suitable semantic mapping (e.g. LIARA datasets) have better results than the ones having semantic gaps (e.g. CASAS datasets).

7.1 Advantages

False alarm is always a tough problem for pattern recognition, because a knowledge base may not be able to contain all the possible patterns in advance. In our case, human behaviors are quite complex and the patterns executing the same activity could be numerous due to alternative execution order, optional or repetitive data. However, our FCA-based engine still has strong robustness to predict, recognize activities and detect abnormal patterns when patterns do not match the ones in the knowledge base.

Different from the majority of expert systems, our FCA-based engine provides a unified inference framework. It clearly represents complicated activity prediction and recognition tasks as a graph search problem and achieves incremental inferences. The scope of probable activities is progressively reduced when new data are observed.

Next, compared with statistical or probabilistic methods, our model requires less training data due to the particular data structure based on the set and graph theories. In the training phase, patterns having different execution orders, but the same observed data set do not affect the structure of Hasse diagram. They only need to update the accumulation matrix for the RMSD-based assessment. The modular design of each component of engine could also assure the convenience of maintenance. We decoupled most of the components in the engine as independent units, and reusable for the other scenarios.

After the feature analysis of behavioral patterns, customized solutions are given out to detect predefined abnormal patterns. Omission of essential data and irrational repetition are two abnormal patterns strongly related to the set theory of discrete mathematics. Through simple algebra of sets and binary operations on sets, they could be easily detected. As shown in Table 5, repetition errors in the data stream were 100% detected, but not all of them are unreasonable (see Sect. 5.2.3). In CASAS, due to the deployment of motion sensors and periodic sampling, data streams are filled with repetitive sensor data. The existence of motion sensors in CASAS also affects the results of the omission error detection. Irregular movements of occupants derive massive patterns having negligible movements as the elements in the optional set O. Thus, the omission error in the sensor events leaded high false-positive rates (12.3%).

In order to reduce false-positive rates and to increase true-positive rates at the same time, it is worthy to note that a weighted array was defined to automatically adjust the detection sensitivity on the basis of the predefined threat level of each data.

To detect causal conflicts in a data stream, the biggest challenge to overcome is the prohibitively expensive definition of causal constraint rules. A essential solution was proposed in Sect. 5.2.5 to automatically extract the rules from historical data. As the result shown in Table 5, causal constraints defined by human expert are accurate and easy to be verified.

The rest two abnormal patterns, mixture of irrelevant data and cognitive distraction, are more complex than the others because of the ambiguous singular position between original intention and the abnormal one. Multi-level inheritance and varied singular positions also aggravate the complexity of situations. In the worst case, some items indicating distraction patterns will be identified as a series of errors mixing irrelevant data in this case.

With our new proposition, we do not need to consider imbalanced class distribution. Only regular patterns corresponding to normal classes could be used for modeling and identifying anomalies in the test data.

7.2 Disadvantages

However, our approach has severe constraints on the training data. Insufficient training items will cause a high false alarm rate in detecting omission of essential data (i.e. wrongly identify derivative subsequences) and mixture of irrelevant data (i.e. wrongly identify derivative longer sequences, see Table 7).

Moreover, conventional FCA construction methods could only build lattices from Boolean binary relations. This restriction limits that if we try to analyze certain numerical relations, we have to convert them into Boolean values by losing precision. For example, in the CASAS dataset, we converted all the positive sensor values into Boolean True when we described the correlations between ubiquitous sensors events and activities. Briefly, if tiny difference between numerical values in binary relations is crucial, we need at least transfer them into enumerable nominal values. Even then, it is not achievable in some extreme cases. A possible solution to deal with this issue is to introduce fuzzy techniques to manage uncertainty and vague information in the relationship (De Maio et al. 2012; Belohlavek 2012).

Due to the limitation of original FCA structure, our current design did not considerate temporal information. However, time intervals between two successive behavioral data is also a very important factor. Many behaviors and abnormal patterns are related to their temporal durations and time intervals. Another potential solution is to use a time extension of Fuzzy FCA to explore useful information according to chronological order among temporal and sequential data (De Maio et al. 2016, 2017).

8 Conclusion and future work

In this paper, we introduced a new incremental activity inference engine to predict and recognize ongoing activities in real-time for the purpose of providing cognitive assistance to elderly people suffering from cognitive impairment. Common cognitive errors are defined by their abnormal behavioral patterns. The inference engine detects these errors and uses weighted arrays to control their detection sensitivities. To improve prediction accuracy, an ontological clustering method is proposed to merge activities according to their semantic similarities. Thus, the engine will predict the ontological superclass instead of directly predicting an activity using few and limited observed data at the early stages.

As the preliminary stage of the research, several complex scenarios such as multiple occupant problem, interleaved, parallel, concurrent and cooperative scenarios have also not been concerned in this paper. In our future improvement, the improvements about dealing with complex scenarios, reducing the restriction of binary relations and combining temporal extension will be taken into consideration.

References

Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7(1):39–59

Acampora G, Cook DJ, Rashidi P, Vasilakos AV (2013) A survey on ambient intelligence in healthcare. Proc IEEE 101(12):2470–2494

Aggarwal CC (2014) Data classification: algorithms and applications. CRC Press, Boca Raton

Aggarwal CC, Reddy CK (2013) Data clustering: algorithms and applications. CRC Press, Boca Raton

Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput Surv (CSUR) 43(3):16

Alsheikh MA, Lin S, Niyato D, Tan HP (2014) Machine learning in wireless sensor networks: algorithms, strategies, and applications. IEEE Commun Surv Tutor 16(4):1996–2018

Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. Lect Notes Comput Sci 3001:1–17

Barberger-Gateau P, Fabrigoule C (1997) Disability and cognitive impairment in the elderly. Disabil Rehabil 19(5):175–193

Bareiss R (1989) Exemplar-based knowledge acquisition: a unified approach to concept representation, classification, and learning, vol 2. Academic Press, London

Belley C, Gaboury S, Bouchard B, Bouzouane A (2015) Nonintrusive system for assistance and guidance in smart homes based on electrical devices identification. Expert Syst Appl 42(19):6552–6577

Belohlavek R (2012) Fuzzy relational systems: foundations and principles, vol 20. Springer Science and Business Media, Berlin

Bouchard B, Giroux S, Bouzouane A (2007) A keyhole plan recognition model for alzheimer's patients: first results. Appl Artif Intell 21(7):623–658

Boudreault A, Whittom-Ross S, Bouchard J, Bouchard B (2016) Validation of a smart stove with traumatic brain injury patients in a cooking task. In: 12th international conference on intelligent environments, IEEE

Boujemaa N, Detyniecki M, Nürnberger A (2008) Adaptive multimedia retrieval: retrieval, user, and semantics, vol 4918. Springer, Berlin

Broekens J, Heerink M, Rosendal H (2009) Assistive social robots in elderly care: a review. Gerontechnology 8(2):94–103

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15

Chen C, Das B, Cook DJ (2010) A data mining framework for activity recognition in smart environments. In: International conference on intelligent environments, pp 80 – 83

Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C Appl Rev 42(6):790–808

Chien BC, Huang RS (2014) Activity recognition using discriminant sequence patterns in smart environments. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 329–341

Cook DJ, Das SK (2007) How smart are our environments? An updated look at the state of the art. Pervasive Mobile Comput 3(2):53–73

Cook DJ, Schmitter-Edgecombe M (2009) Assessing the quality of activities in a smart environment. Methods Inf Med 48(5):480

Cook DJ, Augusto JC, Jakkula VR (2009) Ambient intelligence: technologies, applications, and opportunities. Pervasive Mobile Comput 5(4):277–298

De Maio C, Fenza G, Loia V, Senatore S (2012) Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. Inf Process Manag 48(3):399–418

De Maio C, Fenza G, Gallo M, Loia V, Senatore S (2014) Formal and relational concept analysis for fuzzy-based automatic semantic annotation. Appl Intell 40(1):154–177

De Maio C, Fenza G, Loia V, Parente M (2016) Time aware knowledge extraction for microblog summarization on twitter. Inf Fusion 28:60–74

De Maio C, Fenza G, Loia V, Orciuoli F (2017) Unfolding social content evolution along time and semantics. Future Gener Comput Syst 66:146–159

De Silva LC, Morikawa C, Petra IM (2012) State of the art of smart homes. Eng Appl Artif Intell 25(7):1313–1321

Formica A (2006) Ontology-based concept similarity in formal concept analysis. Inf Sci 176(18):2624–2641

Fortin-Simard D, Bilodeau JS, Gaboury S, Bouchard B, Bouzouane A (2015) Method of recognition and assistance combining passive rfid and electrical load analysis that handles cognitive errors. Int J Distrib Sens Netw 2015:190

Ganter B, Wille R (1999) Formal concept analysis: mathematical foundations. Springer Science and Business Media, Berlin

Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener Comput Syst 29(7):1645–1660

Halle S, Gaboury S, Bouchard B (2016) Towards user activity recognition through energy usage analysis and complex event processing. In: Pervasive technologies related to assistive environments, ACM

Hao J, Bouzouane A, Bouchard B, Gaboury S (2016) Real-time activity prediction and recognition in smart homes by formal concept analysis. In: 12th international conference on intelligent environments, IEEE

Hao J, Gaboury S, Bouchard B (2016) Cognitive errors detection: mining behavioral data stream of people with cognitive impairment. In: Pervasive technologies related to assistive environments, ACM

Jonyer I, Cook DJ, Holder LB (2001) Graph-based hierarchical conceptual clustering. J Mach Learn Res 2(Oct):19–43

Kim E, Helal S, Cook D (2010) Human activity recognition and pattern discovery. IEEE Pervasive Comput 9(1):48–53

Kolodner JL (1992) An introduction to case-based reasoning. Artif Intell Rev 6(1):3–34

Leskovec J, Rajaraman A, Ullman JD (2014) Mining of massive datasets. Cambridge University Press, Cambridge

Loia V, Fenza G, De Maio C, Salerno S (2013) Hybrid methodologies to foster ontology-based knowledge management platform. In: 2013 IEEE symposium on intelligent agent (IA). IEEE, pp 36–43

Minor B, Doppa JR, Cook DJ (2015) Data-driven activity prediction: algorithms, evaluation methodology, and applications. In: Proceedings of the 21th ACM sigkdd international conference on knowledge discovery and data mining. ACM, pp 805–814

Nazerfard E, Cook DJ (2013) Using Bayesian networks for daily activity prediction. In: AAAI workshop: plan, activity, and intent recognition

Nazerfard E, Cook DJ (2015) Crafft: an activity prediction model based on Bayesian networks. J Ambient Intell Humaniz Comput 6(2):193–205

Poelmans J, Ignatov DI, Kuznetsov SO, Dedene G (2013) Formal concept analysis in knowledge processing: a survey on applications. Expert Syst Appl 40(16):6538–6560

Porter BW, Bareiss R, Holte RC (1990) Concept learning and heuristic classification in weak-theory domains. Artif Intell 45(1–2):229–263

Ramos C, Augusto JC, Shapiro D (2008) Ambient intelligence—the next step for artificial intelligence. IEEE Intell Syst 23(2):15–18

Remagnino P, Foresti GL (2005) Ambient intelligence: a new multidisciplinary paradigm. IEEE Trans Syst Man Cybern Part A Syst Hum 35(1):1–6

Roy P, Giroux S, Bouchard B, Bouzouane A, Phua C, Tolstikov A, Biswas J (2011) A possibilistic approach for activity recognition in smart homes for cognitive assistance to Alzheimer's patients. In: Activity recognition in pervasive intelligent environments. Springer, pp 33–58

Ryoo M (2011) Human activity prediction: Early recognition of ongoing activities from streaming videos. In: International conference on computer vision, pp 1036–1043

Sadri F (2011) Ambient intelligence: a survey. ACM Comput Surv (CSUR) 43(4):36

Suryadevara NK, Mukhopadhyay SC, Wang R, Rayudu R (2013) Forecasting the behavior of an elderly using wireless sensors data in a smart home. Eng Appl Artif Intell 26(10):2641–2652

United Nations (2013) World population ageing 2013. Economic and social affairs. UN Publications. http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf

United Nations (2016) Alzheimer's disease fact sheet. National Institute on Aging. NIH Publication. https://www.nia.nih.gov/alzheimers/publication/alzheimers-disease-fact-sheet

Van Kasteren TL, Englebienne G, Kröse BJ (2011) Hierarchical activity recognition using automatically clustered actions. In: International joint conference on ambient intelligence. Springer, pp 82–91

Xing Z, Pei J, Keogh E (2010) A brief survey on sequence classification. ACM SIGKDD Explor Newslett 12(1):40–48

Yau SS, Liu J (2006) Hierarchical situation modeling and reasoning for pervasive computing. In: International workshop on collaborative computing, integration, and assurance, IEEE

Ye J, Dobson S (2010) Exploring semantics in activity recognition using context lattices. J Ambient Intell Smart Environ 2(4):389–407