Real-time Activity Prediction and Recognition in Smart Homes by Formal Concept Analysis

Jianguo Hao, Bruno Bouchard, Abdenour Bouzouane, Sébastien Gaboury
LIARA Laboratory
Université du Québec à Chicoutimi (UQAC)
555 Boulevard de l'Université, Chicoutimi (Québec), Canada, G7H 2B1
{jianguo.hao1, bruno.bouchard, abdenour.bouzouane, sebastien.gaboury}@uqac.ca

***Abstract*—In this paper, we introduce a new knowledge-driven approach based on the Formal Concept Analysis (FCA) to predict and recognize Activities of Daily Living (ADLs) in ubiquitous computing environments, in order to duly provide continuous assistance for residents. The proposed approach constructs an incremental inference engine and achieves progressive deductive reasoning to recognize an unfinished ongoing ADL in real-time. For the purpose of finding out the most probable ongoing activity among possible candidates, we propose an assessment based on the root-mean-square deviation (RMSD) to evaluate the relevance of each intermediate prediction. Besides the on-the-fly recognition mode, our approach also possesses high discrimination in differentiating derived and similar activities. Excellent recognition results (almost 100%) and high prediction accuracies (more than 70%) are obtained in the experiments.**

***Index Terms*—Smart environment, activity prediction, activity recognition, data mining, formal concept analysis.**

I. Introduction

In the 21st century, due to the low fertility rate all over the world, population ageing has become a crucial worldwide issue. As one of the most common diseases of old age [1], Alzheimer's disease has attracted more and more attention from multidisciplinary scientific communities. The patients suffering from Alzheimer's disease are always in need of specific intervention and supervision [2], so Alzheimer's disease is actually one of the most financially costly diseases in developed countries. Its main symptom is related to the planning problems [3]. Compared to healthy people, when carrying out Activities of Daily Living (ADLs), Alzheimer's patients tend to produce more implementation errors due to cognitive deficits [4].

With huge commercial prospects and rapid development of information and communication technology in recent years, smart environments have become a very active research topic. As a promising solution, the design of smart homes tries to make people with cognitive impairment 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 first have to recognize and understand an ongoing ADL in real-time, even predict it in advance, and then offer assistance. Hence, in this paper, we focus on the issue of real-time ADLs prediction and recognition.

II. Related Work

Under the help of home automation and ubiquitous computing, new generations of smart homes devote to provide dynamic, intelligent, suitable and considerate personal services to their residents. To accomplish this mission, the main work of research concentrates on analysing massive heterogeneous data captured from various sensors and wearable devices [5]. According to different types of data, the solutions about ADLs recognition could be broadly classified as two categories: vision-based and non-vision-based approaches.

Vision-based approaches firstly use cameras to capture image sequences for tracking real-time positions of objects and for identifying the states of moving objects by image processing or pattern recognition, and then recognize related activities by machine learning [6], semantic ontology [7], or by the other methods. According to information entropy [8], vision-based approaches possess more complete information and interpret more details than non-vision-based ones.

Although excellent results have been shown in some specific issues like multiple people activity recognition [9], vision-based approaches have also brought in the debate surrounding privacy at the same time. Non-vision-based approaches have no such controversy over privacy because their captured data contains less private information. They usually rely on object-attached radio-identification devices or on numerical sensors (e.g. wearable, binary, infrared, pressure, etc.) to collect diverse values about the current states (e.g. distance, motion, ambient change, device usage, energy consumption, etc.). Because of the inherently existent large semantic gaps between low-level sensor data and high-level human behaviors [10], the ultimate goal of non-vision-based approaches always focuses on choosing appropriate analytic granularity and furthermore on interpreting the raw captured data. Solutions about the latter case are mainly based on signal analysis [11], logical reasoning [12], semantic ontology [13], probabilistic statistics [14] or on data mining [15].

Low-level *sensor data*, intermediate-level *atomic actions* and high-level *ADLs* are the three most commonly chosen levels of granularity in the research of ubiquitous computing [13]. We define their interrelations as follows: each atomic action (hereafter referred as *action*) is the smallest topic-related meaningful behavioral unit describing one short-term intention of a resident and should be detected by one or more sensors. Each ADL consisting of more than one action indicates the real long-term intention of the resident. Thus, considering the issue about providing timely cognitive assistance in smart homes, the combination of intermediate-level actions and highest-level ADLs should be the most appropriate analytic granularity.

In our previous researches, the mapping between sensor data and related actions was successfully established. Sufficiently accurate sequences of actions were obtained by electrical devices identification [11] and by passive RFID technology [15]. These previous work could make us concentrate on the suitable analytic granularity in the next study. Moreover, a logic framework based on four-level lattice was also defined for plan recognition [16], and it prompts us to solve the activity recognition problem with more mature lattice-based models. On the basis of these experiences, we introduce a novel approach based on the Formal Concept Analysis (FCA) to predict and recognize complex ADLs.

The rest of the paper is organized as follows. In the next section, we put forward the new solution about ADLs prediction and recognition in our non-intrusive, passive RFID-based ubiquitous computing environment. In Section IV, we demonstrate the excellent performances on various criteria and evaluate the experimental results. The advantages and disadvantages are discussed in Section V. Finally, we conclude in Section VI.

III. Real-time Activity Prediction and Recognition Based on FCA

In this section, we first start by giving some knowledge about FCA, including its basic definitions, construction methods and visual representation. Then, through feasibility analysis, we explain the rationale why FCA-based models are effective in handling ADLs recognition and varied related problems. Next, as one of the most important roles of our approach, we introduce in detail our conception about how to incrementally search and locate a given sequence of actions in the FCA-based knowledge base for real-time ADLs prediction and recognition. Finally, we present how to assess each intermediate prediction to find out the most probable ADL.

A. Basic Definitions

An objective concept is an abstract representation describing a thing or a scenario in reality. As a subfield of applied mathematics, FCA clusters seemingly scattered concepts and creates interrelations among them by mathematical order theory [17]. FCA has excellent ability in concept hierarchy analysis and could well describe binary relations between two sets of data which commonly appear in many areas of human activities [18]. In the following statements, we introduce each FCA component and its corresponding mathematical representation.

1) Formal Context (Context): is a triplet $\mathcal{K} = \langle G, M, I \subseteq G \times M \rangle$ where G and M are two non-empty sets that define the analytic domains. The elements in G and M are severally called the *objects* and the *attributes* of $\mathcal{K}$. Set I is one subset of Cartesian product $G \times M$ representing binary relations between G and M, which defines the issue to be solved.

The content of $\mathcal{K}$ could be visualized as a binary matrix consisting of $|G|$ rows and $|M|$ columns (see Figure 1). If an object g has a binary relation I with an attribute m, there will be a cross in row g and column m in the corresponding binary matrix, expressed as gIm [17].

	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. 1. An example of binary matrix

With the help of such triplet structure, we could derive different issues in light of different analytic granularity. Let's take our activity recognition problem as an example. Suppose that in ubiquitous computing environment, a set of **S**ensor states is denoted as S, a set of **O**bservable sequences of actions is denoted as O and an **A**DLs set is denoted as A. Thus, we represent the three most common issues as follows:

- $\langle O, S, I' \rangle$: action recognition by sensor states analysis.
- $\langle A, S, I'' \rangle$: ADLs recognition by sensor states analysis.
- $\langle A, O, I''' \rangle$: ADLs recognition based on the analysis of sequences of actions.

Figure 2 illustrates the mappings above. As mentioned earlier, to recognize complex ADLs, the combination $\langle A, O \rangle$ is more meaningful than the other ones. Therefore, in this paper, non-empty set G denotes the ADLs to be recognized and M represents the actions in the observed sequences.

2) Concept-Forming Operations: Each context induces a pair of operations, so-called the *concept-forming operators* [18], that are very important to form formal concepts. Their main function is to match all shared elements quickly. For $A \subseteq G$, we define:

$$A' := \{m \in M \mid \text{for each } g \in A,\ gIm\} \quad (1)$$

to obtain all attributes shared by all objects from A.

Likewise, for $B \subseteq M$, we define:

$$B' := \{g \in G \mid \text{for each } m \in B,\ gIm\} \quad (2)$$

to obtain all objects shared by all attributes from B.

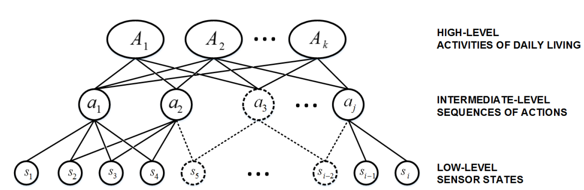

Fig. 2. Multilevel structure existing in ubiquitous computing environments.

3) Formal Concept (Concept): is a pair $(A \subseteq G, B \subseteq M)$ meeting the conditions of $A' = B$ and $B' = A$ (or we can express $A'' = A$ or $B'' = B$) under $\mathcal{K}$. A and B are respectively called the *extent* and the *intent* of the concept. As a matter of fact, each concept is a stable tiny closure. No matter how many times the *Concept-Forming Operations* have been taken, the extent and intent will not be spontaneously changed. $\mathcal{B}(G, M, I)$ denotes the universe containing all concepts of $\mathcal{K}$.

4) Concept Lattice (Lattice): is an ordered lattice structure. Suppose that (A_1, B_1) and (A_2, B_2) are two concepts of a context $\mathcal{K}$, (A_1, B_1) is called a *subconcept* of (A_2, B_2) if either $A_1 \subseteq A_2$ or $B_2 \subseteq B_1$. Meanwhile, (A_2, B_2) is a *superconcept* of (A_1, B_1), and written as $(A_1, B_1) \preceq (A_2, B_2)$. The relation $\preceq$ is called the *hierarchical order* (*partially order* or simply *order*) that orders and connects two hierarchical concepts.

All the concepts in the universe $\mathcal{B}$ ordered by $\preceq$ form a lattice $\underline{\mathcal{B}}(G, M, I)$. It is another closure of $\mathcal{K}$: once a stable concept has been activated by external factors, it becomes unstable and will transit to another adjacent stable concept.

The question concerning about how to construct a lattice from a binary matrix is outside the scope of this paper. The construction of lattice is the process that enumerates all the concepts of a context $\mathcal{K}$ and orders them simultaneously. In recent decades, great effort has been devoted to the study of efficient constructions [19]. Our implementation is based on Christian Lindig's research [20] due to excellent performance.

5) Hasse Diagram: in mathematical order theory, every finite ordered set can be represented by a line diagram called *Hasse diagram* [17]. It is also a visualization of lattice that represents component concepts as nodes of graph-based multilayer structure (see Figure 3). There are two special nodes: the topmost node $\{G, \varnothing\}$ called the *Supremum* and the lowermost node $\{\varnothing, M\}$ called the *Infimum*. Each layer connects another one or two adjacent layers by *Galois connection* [17]. Furthermore, any two adjacent layers generate a bipartite graph and no concept connects with another one in the same layer.

B. Feasibility Analysis

In the following statements, we analyze the feasibility about using FCA to predict and to recognize ADLs from real-time sequences of actions captured by the sensor network in a smart home. Our discussion surrounds in three aspects as follows:

1) Knowledge Representation: as shown in Figure 2, the relations in the multilevel structure are easy to be transformed into binary relations. If an element m in a lower level connects to an element g in a higher one, it means that there is a relation I between them, represented as gIm. All the connections between two different levels could be represented as the crosses in the corresponding binary matrix.

2) Hierarchical Structure: as a visualization of FCA, lattice $\underline{\mathcal{B}}(G, M, I)$ is made up of all the concepts belonging to a context $\mathcal{K}$. The property of concept closure ensures the uniformity that any constraint restricting to any component element of a concept must be concurrently respected by the rest of the other component ones in the same extent or intent.

As we referred before, a stable concept will be activated while its inner structure is being changed by intent extension or extent reduction. According to the predefined hierarchical order, an unstable concept will transit to one of its connected concepts in the Hasse diagram to become stable again. Such transition is also the basic principle of our ADLs recognition strategies. It is worth mentioning that this kind of transitions will not cause the structure change of lattice.

As shown in Figure 3, the topmost Supremum indicates the initial state of the process and the lowermost Infimum indicates the final state. The intent of a concept indicates the given observed actions and the extent indicates all probable ADLs. Their meanings behind the mathematical abstractions are also consistent with the realistic logic: For the Supremum, if a resident did not do anything (i.e. intent is empty), his real intention is unpredictable and any indexed ADLs could be done next (i.e. extent is the universe of indexed ADLs). For the Infimum, there is normally no activity (i.e. extent is empty) consisting of all indexed actions (i.e. intent is the universe of indexed actions).

For any concept in $\underline{\mathcal{B}}$ except the Infimum, its extent will shrink in size when a new observed action extends the intent. The activated concept will transit to one of its subconcepts, or we can say that it is shifting towards the Infimum. The shrunken extent $A_{\downarrow}$ is the subset of the old one A, denoted as $A_{\downarrow} \subset A$. Equation (3) describes this property as below:

$$B_{\uparrow} := B \cup m \Longrightarrow (A_{\downarrow}, B_{\uparrow}) \preceq (A, B) \tag{3}$$

where $m \in M$ is the incoming action which has already been indexed, but not existed in the intent ($m \notin B$) before the extension. The subscript arrows indicate the changes in size.

With the successive loading of actions (referred as intent extensions), the scope of possible ADLs will be reduced progressively (referred as extent reductions). This process produces many intermediate transited concepts called *local optimal concept* containing inferred ADLs called *local optimal solutions* in each extent. Meanwhile, a non-Infimum concept is considered *global optimal concept* if and only if the Infimum is the sole subconcept in the following possible intent extensions. Thus, the issue about ADLs prediction and recognition in sequences of actions has been rewritten as how to progressively locate local optimal concepts in a Hasse diagram during successive intent extensions. For this reason, it could also be regarded as a graph search process from the Supremum to Infimum in a Hasse diagram.

3) Properties: Due to specific data structure based on set and graph theories, FCA-based models have strong robustness against several common errors. This feature is mainly reflected in:

Fault Tolerance: FCA-based models are compatible with redundancy and incompleteness. Because of the set-based feature described in Equation (4), incoming redundant elements that have already existed in a stable concept will not trigger the concept transition during intent extensions.

$$m \in B \Longrightarrow m \cup B = B \tag{4}$$

For incompleteness, global optimal concepts could also be located at most of the time. Even though in worst case, the incompleteness only makes a global optimal concept have more than one possible prediction (i.e. more than one element in the extent), but it will not lead to illogical inferences.

Excludability: FCA is naturally sensitive to irrelevant elements. If m is an irrelevant element having no binary relation with any element in the extent, after intent extensions, there is definitely no subconcept containing the subset $m \cup B$. Furthermore, current unstable concept will shift directly to the Infinum.

The two properties above have demonstrated the great potential of our FCA-based approach in cognitive error detection. The former not only makes the real-time ADLs recognition in noise data become possible, but also provides a feasible solution for detecting and reminding unfinished ADLs. The latter could be used to detect those irrelevant actions mixing into current plan.

C. Real-time Prediction and Recognition

As mentioned above, the issue about ADLs prediction and recognition based on FCA is similar to a graph search process. Their core ideas are both to establish an efficient strategy to look for the most similar patterns in a graphical knowledge base through limited observed clues. A FCA-based model is such a well-constructed knowledge base which contains all predefined patterns to deduce possible ADLs like an expert system through given sequences of actions.

Suppose a *sequence* α is an ordered list of actions, denoted as $\alpha = \{\alpha_1 \prec \alpha_2 \prec ... \prec \alpha_m\}$, where $\alpha_j \prec \alpha_{j+1}$ means action α_j occurs before α_{j+1}. In this paper, sequence α is defined as an ordered cache successively loading all the detected actions obtained by previous work [11][15]. At the same time, we also define a *token* to simulate serial transitions of the concepts in the excited state in the Hasse diagram.

Because of the successive manner loading actions, classical graph traversal strategies could not perfectly meet our requirements in incremental reasoning. Most of them traverse the whole graph to locate their target and could not recall and continue the previous interrupted position in the following searches. When a new observed action is loaded into α, they have to start over again for searching the updated α and abandon all the previous reasoning.

For these reasons, we propose an incremental graph search algorithm for locating local optimal concepts in the FCA-based models. Each new search could continue from the previous interrupted position. The advantage is not only reflected on the efficiency, but also on the consistency of reasoning. Unlike searching for a particular node or the shortest path, our issue concentrates on matching the expanding α with all concepts in the Hasse diagram and locating local optimal concepts (A, B) that $\alpha \subseteq B$. There is usually more than one superset satisfying $\alpha \subseteq B$. If $\alpha \subseteq B_1$ and $B_1 \subseteq B_2$, then $\alpha \subseteq B_2$, but only the topmost superset containing α could ensure the consistency of searches. This hypothesis is described in Equation (5) as follows:

$$\forall B \supseteq \alpha, T_b \subseteq B \Longrightarrow (A, B) \preceq (T_A, T_b) \text{ and } A \subseteq T_A \tag{5}$$

where T_b is the topmost and also the smallest superset of α. That is, all the other supersets of α are also the supersets of T_b. Meanwhile, the affiliated concept (T_A, T_b) is the topmost superconcept, and T_A is the superset of all the other extents. For example, in Figure 3, if $\alpha = \{b\}$, then node four $n_4(124, ab)$ is the topmost superconcept we are looking for. If we choose another concept (e.g. node six $n_6(24, abc)$) instead of the topmost one, several inferred elements in the extent will be lost (e.g. $\{124\} - \{24\} = \{1\}$) and could never be found in the following searches (i.e. if $\{1\}$ is missed in n_4, it could never be found in node n_6 and its subconcepts).

The following statements are divided into two parts. Firstly, we discuss our proposition about incremental graph searching algorithm for ADLs prediction and recognition. Secondly, considering a common problem named *multilevel inheritance*, we introduce a greedy search strategy as a simplified solution.

1) Graph Search Algorithm: Breadth-first search (BFS) is one of the most common graph traversal algorithms. The main idea is to explore all the neighbor nodes in the same level before moving to the next level neighbors. Because of the successive manner loading α, BFS is more efficient than another common search algorithm called depth-first search (DFS) in our case. However, just like the other classical algorithms, BFS itself could not well locate each topmost superset. If we simply suppose that the first concept containing α discovered by BFS is the topmost one, it seems like no problem. However, considering the example in Figure 3, if $\alpha = \{abc\}$, then n_7 is always the first suitable concept discovered because of the paths $n_0 \rightarrow n_2 \rightarrow n_7$ 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 search algorithm (HGS) to locate each local optimal concept, which is also the topmost superconcept containing each expanded α. As can be seen from the name, HGS is a half-duplex search consisting of two directions search. The *top-down search* firstly locates the first discovered superconcept containing α. And then, starting from there, the *down-top search* turns back along the hierarchical order and looks for the topmost superconcept. More details about HGS algorithm are sketched in Pseudo-codes (1) and (2).

In Pseudo-code (2), we need to pay attention to Line (12) which seeks the superconcept containing the minimal cardinality of intent in set S. The cardinality of an intent means the minimal amount of steps (actions) needed to completely achieve the concept. Dues to Equation (5), we could see that the topmost superconcept has smaller cardinality than any other supersets of α.

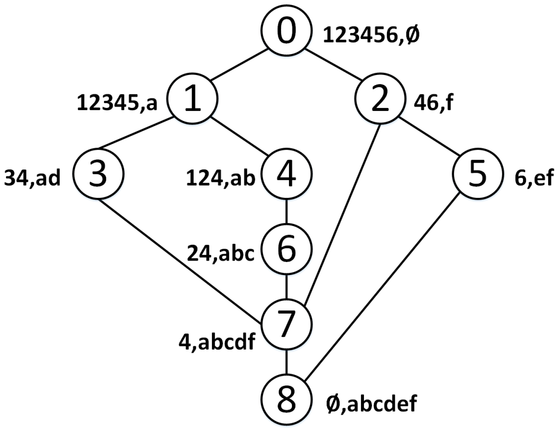

Fig. 3. Hasse diagram generated from the binary matrix in Figure 1.

Algorithm 1: Top-down search of HGS algorithm

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

Algorithm 2: Down-top search of HGS algorithm

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

2) Graph Search Strategy: Beyond the issue of locating each topmost superconcepts, we also need to consider another tough issue about multilevel inheritance. Multilevel inheritance is a very common situation existing in ADLs due to diverse lifestyles and personal habits. Besides flexible execution orders, an ADL could also be accomplished by adding or omitting optional actions. Thus, the derived activities of an ADL have always had the multilevel inheritance relations between each other. 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 $A_1 \subset A_2 \subset A_3 \subset A_4$.

Thus, the graph search strategy that we adopt is based on the *greedy* manner. That is, if an ADL is recognized, the central system will be notified by an alert. At the same time, the completeness of the recognized ADL will also be verified until all the integral actions in the corresponding intent have been done. If the recognized ADL is one of the inherited activities, we continue receiving observed actions from α until the token shifts to the Infimum.

D. Prediction assessment

In local optimal concepts, there are normally more than one predicted candidate (probable ADL) in the extent because the observed actions belonging to an ADL are given by the successive manner. Without an efficient assessment, redundant predictions will be useless to make decisions for real-time cognitive assistance. In this case, we desire to evaluate the relevance of each predicted candidate in a local optimal concept and choose the most relevant one as *local optimal prediction*. A real-time assessment is discussed as follows.

The way to accomplish an ADL g_i could be various due to flexible execution orders, optional or repetitive actions. At the same time, each person also has his own preference to execute a same ADL, and there are normally a few deviations among each execution order. Based on this hypothesis, we make use of historical behavioral data containing the preferences of a resident to train an *accumulation binary matrix*.

Normally, binary matrices used to create lattices only store Boolean values to describe binary relations between two non-empty sets. Therefore, some valuable historical information beyond Boolean relations has been abandoned. The core idea of accumulation binary matrix is to establish a series of naive distributions about expected positions of component actions of indexed ADLs.

We propose our assessment as follows: for each predicted ADL g_i in the extent, under the condition of executing g_i, we calculate the deviation between the actual normalized positions in the sequence α and the accumulated expected ones. Thus, the local optimal prediction should be the one with minimal deviation which has the most suitable path comparing with the historical data. Obviously, our assessment consists of two modules: accumulation and assessment.

1) Accumulation: for each action α_j in a new training item α, a complete execution sequence of ADL g_i (i.e. $\alpha_j \in \alpha$, $\alpha \in g_i$), we update the accumulated value at corresponding element (g_i, α_j) in the binary matrix by Equation (6):

$$\sigma_{ij} = \sigma'_{ij} + j \tag{6}$$

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 (7) represents the same accumulation in a global view:

$$\sigma_{ij} = \sum_{k=1}^{N_{ij}} \sigma_{(ij,k)} \tag{7}$$

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

2) Assessment: after receiving each new observed action, we calculate the normalized position of each action α_j in the sequence α given each predicted ADL g_i in the extent. The normalized position $\overline{\varphi_{ij}}$ of action α_j executing g_i is calculated by Equation (8).

$$\overline{\varphi_{ij}} = \frac{1}{\#\alpha_j} \sum_{k=1}^{|\alpha|} k, \quad \text{if } \alpha_k = \alpha_j \tag{8}$$

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 all $\#\alpha_j$ times appearances of α_j to calculate the normalized position.

Equation (9) expresses the root-mean-square deviation D_i of current sequence α executing g_i:

$$D_i = \sqrt{\frac{1}{|\alpha|} \sum_{\forall \alpha_j \in \alpha} (\overline{\varphi_{ij}} - \frac{1}{N_{ij}} \sigma_{ij})^2} \tag{9}$$

where $\forall \alpha_j \in \alpha$ restricts that the variance of each action in α should be used to calculate the root-mean-square deviation.

Thus, a set of RMSD $\{D_1, D_2, ..., D_i\}$ is calculated by α and by the current extent $A = \{g_1, g_2, ..., g_i\}$. The element $g \in A$ possessing the minimal RMSD value is the local optimal prediction because of the best fitting with historical data.

IV. EXPERIMENTS

In this section, we first describe the characteristics of experimental datasets, and then evaluate the experimental results using 10-fold cross-validation at the following criteria: time cost (in both training and execution phase), ADLs prediction and recognition accuracies.

A. Experimental Data

Our experimentation is based on the real dataset *RDATA* captured by the sensor network in our LIARA laboratory, the synthetic dataset *DDATA* derived from RDATA, and the benchmark dataset *CASAS* [21]. Because of complicated interactions and more multilevel inheritance ADLs, we choose kitchen activities as our main research objects.

TABLE I
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 II
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

The statistical information about RDATA is shown in Table (I). There are twelve kitchen activities named in camel case with the amount of non-redundant component actions.

Normally, the complete data structure of training data RDATA contains four kinds of information: *temporal information (TI)* indicating execution orders, *sensor information* indicating topological relationships between physical entities [11][15], recognized *action* and corresponding *ADL*. To simplify our problem, we transferred the temporal information to step counts and removed the part of sensor information. Table (II) sketches the final data structure of our training items.

B. Evaluation

The evaluations that involve the performance of our algorithm introduce in this subsection. It is worth mentioning that all the related evaluations were carried out on a single laptop with tech specs of Intel Core i7 Processor (2.4GHz) and 8GB RAM, under Ubuntu 14.04.

1) Time Cost: The time costs for training different lattices with different sizes are shown in Table III. Compared to RDATA, DDATA has the same statistical information in size because the lattice construction only depends on the binary relations (i.e. lattice structure only depends on the set of component actions of each ADL). That is also the reason why FCA-based models could well handle the ADLs with flexible execution orders without additional training costs.

TABLE III
TIME COST FOR TRAINING CONCEPT LATTICES

Dataset	Lattice Size			Time Cost
	activities	actions	concepts	
RDATA	12	69	24	0.0023
DDATA	12	69	24	0.0047
CASAS	5 (120)	25	430	0.7112

We also used the benchmark dataset CASAS as reference. Unlike our research topic, it is based on the mapping between low-level sensor data and high-level ADLs as we mentioned in Figure 2. The data in CASAS was captured by a series of motion, binary and numerical sensors, and represents five main ADLs in a smart environment. However, each ADL in CASAS has diverse sets of component actions describing a same ADL due to the motion sensors. In this case, each different set of component actions was treated as a different ADL in the lattice (i.e. 120 different ADLs derived from five ADLs, see Table III). Once any set of actions is identified by our approach, the affiliated ADL will be predicted and recognized as well.

2) Recognition Accuracy: The results shown in Table IV shows the excellent performance in ADLs recognition under the limitation of no missing values in the training dataset. To generate DDATA, we disrupted the internal execution orders of the training items in RDATA under the condition of following the causal restrictions between component actions.

TABLE IV
TIME COST AND ACCURACY OF ADLS RECOGNITION

Dataset	No. Items	Accuracy	Time Cost
RDATA	240	100%	0.0081
DDATA	96972	100%	5.1789
WSU CASAS	120	86.7%	0.0261

We evaluated RDATA and DDATA by 10-fold cross-validation to reduce overfitting. The k-fold cross-validation could reduce the unreliable estimate of future performance while increasing the bias [22]. The recognition accuracy of CASAS is better than the experimental results using incremental training by the classical HMM method, but inferior to the ones using off-line training [23].

3) Prediction Accuracy: ADLs prediction and related assessment occur after each transition happened in the diagram. Successive operations loading new observed actions into sequence α are called the *serial stages* and a local optimal prediction will be chosen at each stage.

Figure 4 depicts the average prediction accuracies at different stages and the trends. For RDATA and DDATA, the scopes of valid stages are from 1 to 20, and for CASAS, the one is from 1 to 80 (accuracy is 100% after Stage 25). Assessment accuracies have been gradually improved when more and more actions are being observed and loaded. Additionally, a resident firstly has to move to the right place to carry out an ADL. Thus, in CASAS, the prediction accuracies are better than another two datasets at the early stages due to the fixed motion

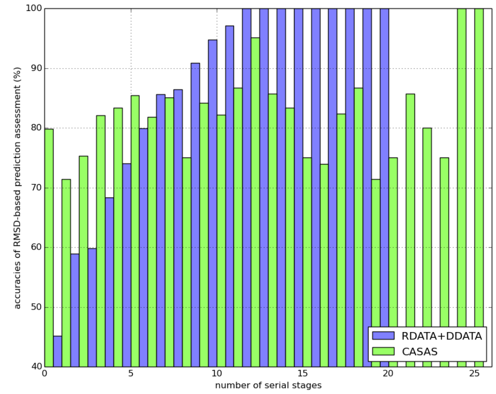

Fig. 4. Prediction accuracies based on the RMSD at different stages.

sensors. Compared with [24], our prediction is more stable at the early stages.

V. DISCUSSION

Summing up the results obtained in Section IV, it is possible to conclude that our incremental approach is suitable and efficient for real-time activity prediction and recognition in ubiquitous computing environments.

A. Advantages

First, different from the majority of expert systems based on scattered deductive reasoning, our FCA-based hierarchical model provides a unified powerful deductive logic framework. It clearly represents complicated ADLs prediction and recognition as a graph search problem and spontaneously achieves progressive deductive reasoning. Through representing ADLs and component actions as binary relations, we could obtain enumerable concepts consisting of shared actions (intent) and affiliated similar activities (extent). With the successive manner loading actions in real-time, the scope of probable ADLs in the extent shrinks gradually and the global optimal concept will be located at the end. All related inferences are automatically deduced by the closure transitions in the Hasse diagram.

Then, as an improved version of BFS, our graph search algorithm has obvious advantages in efficiency and in consistency of reasoning. Unlike classical graph traversal algorithms abandoning all the previous reasoning, our incremental approach need neither to start over again nor to traverse the whole graph to look for the local optimal concept after each intent extension. On the premise of no effect for the final results, our HGS algorithm continues each new round of reasoning from previous interrupted positions. Moreover, our graph search strategy could also distinguish multilevel inheritance ADLs.

Next, compared with the other statistical or probabilistic methods, our FCA-based approach has fewer volume requirements of training data due to the data structure based on the set and graph theories (see RDATA and DDATA statistical information in Table III). In the training phase, the chaotic execution orders and the repetitive actions in the new training items are insignificant if the sets of component actions are same as the ones existing in the context, we do not need to rebuild models and only to update the accumulation binary matrix for the RMSD-based assessment.

After that, real-time ADLs prediction assessment will be triggered in each iteration looking for each local optimal concept. Once a local optimal concept is located, the relevance of each candidate in the extent will be evaluated to choose the most probable ADL that may occur.

Finally, our approach has great superiority in the knowledge reuse and self-adaptation. The trained Hasse diagram and the accumulation binary matrix are designed as two independent uncoupled modules. If one module has been modified, there is no influence to another one. As a consequence, accumulation binary matrices could also be reusable for the other scenarios.

B. Disadvantages

Our FCA-based approach works if and only if different sets of attributes (eg. actions) could well distinguish different objects (eg. ADLs). Besides, there exists several unnegligible disadvantages as well. The first disadvantage refers to the loss of precision. Classical 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 values of sensors into Boolean True when we described the interactions between ubiquitous sensors network and ADLs. Briefly, if tiny difference between numerical values in binary relations is crucial, we need at least transfer them into the enumerable nominal form. Even then, it is not achievable in some extreme cases. Moreover, there are critical requirements in correctness and completeness for the training data. Any error in the training items will affect the overall accuracy.

The second one is about the boundary segmentation. The current version of our approach is designed for bounded ADLs prediction and recognition. If we would like to use it to handle a sequence of actions containing more than one successive activity, it is necessary to manually define boundaries for these adjacent activities. If not, it will cause the alignment errors.

The third one is related to the assessment based on RMSD. The natural lattice structure does not contain temporal information about execution orders, so the bias in the assessment due to incidental factors could not be completely avoided.

At last, as a common problem appearing in the other state-of-the-art prototypes, unseen ADLs could not be predicted or recognized if no corresponding training items are available in the dataset [10].

VI. CONCLUSION

For the purpose of knowing which ADL is being done by a resident in a smart home, in this paper, we presented our current efforts towards real-time ADLs prediction and recognition in ubiquitous computing environments. We designed a real-time knowledge-driven approach based on FCA to progressively analyze sequences of actions. We also proposed an assessment evaluating the relevance of each intermediate prediction to obtain the most probable ADL.

As mentioned earlier, our FCA-based approach has shown great potential to detect several typical cognitive errors caused by the deterioration of cognitive skills. Hence, in the future works, we intend to concentrate on this issue and to extend our approach to more complicated real-world scenario like identifying complex parallel or interleaved activities.

REFERENCES

[1] National Institute on Aging. *Alzheimer's Disease Fact Sheet*. NIH Publication, 2015.

[2] United Nations, Economic and Social Affairs. *World population ageing 2013*. UN Publications, 2013.

[3] P. Roy, S. Giroux, B. Bouchard, A. Bouzouane, C. Phua, A. Tolstikov, and J. Biswas. A Possibilistic Approach for Activity Recognition in Smart Homes for Cognitive Assistance to Alzheimer's Patients. *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58, 2011.

[4] H. Förstl and A. Kurz. Clinical features of alzheimer's disease. *European archives of psychiatry and clinical neuroscience*, pages 288–290, 1999.

[5] A. S Crandall and Diane J Cook. Smart home in a box: A large scale smart home deployment. In *Workshop Proceedings of the 8th International Conference on Intelligent Environments*, page 169, 2012.

[6] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, pages 129–162, 2004.

[7] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra. Event detection and recognition for semantic annotation of video. *Multimedia tools and applications*, pages 279–302, 2011.

[8] C. E. Shannon. A mathematical theory of communication. 1949.

[9] S. Ke, H. Thuc, Y. Lee, J. Hwang, J. Yoo, and K. Choi. A review on video-based human activity recognition. In *COMPUTERS*, 2013.

[10] H. Cheng. Learning and recognizing the hierarchical and sequential structure of human activities. 2013.

[11] C. Belley, S. Gaboury, B. Bouchard, and A. Bouzouane. Nonintrusive system for assistance and guidance in smart homes based on electrical devices identification. *Expert Systems with Applications*, 2015.

[12] D. Riboni and C. Bettini. Owl 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing*, 2011.

[13] L. Chen, J. Hoey, Chris D Nugent, Diane J Cook, and Z. Yu. Sensor-based activity recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, pages 790–808, 2012.

[14] N. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *Proceedings of the 17th British Machine Vision Conference*, pages 1239–1248, 2006.

[15] D. Fortin-Simard, J. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane. Exploiting passive rfid technology for activity recognition in smart homes. *IEEE Intelligent Systems*, 2015.

[16] B. Bouchard, S. Giroux, and A. Bouzouane. A keyhole plan recognition model for alzheimer's patients. *Applied Artificial Intelligence*, pages 623–658, 2007.

[17] B. Ganter and R. Wille. Formal Concept Analysis: Mathematical Foundations. 2012.

[18] R. Belohlavek. Introduction to formal concept analysis. *UPOL*, 2008.

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

[20] Christian L. Fast concept analysis. In *ICCS 2000*, pages 152–161. Shaker Verlag, 2000.

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

[22] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection.

[23] B. Chien and R. Huang. Activity recognition using discriminant sequence patterns in smart environments. In *Trends and Applications in Knowledge Discovery and Data Mining*, pages 329–341. 2014.

[24] R. Ross and J. Kelleher. Accuracy and timeliness in ml based activity recognition. 2013.