Activity Mining in a Smart Home from Sequential and Temporal Databases

Josky Aïzan[1,2], Cina Motamed[2] and Eugene C. Ezin[1]
[1]*Institut de Mathématiques et de Sciences Physiques, Université d'Abomey-Calavi, Bénin*
[2]*Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université du Littoral Côte d'Opale, France*

Keywords: Sequential Pattern Mining, Smart Home, Activity of Daily Living.

Abstract: In this paper, we implement the Sequential Pattern Mining from Temporal Databases to learn activity in a smart home. The Pre-processing is firstly conducted on sensor data by taking into account the timestamp of sensor events. Then we extract typical activities using a sequential pattern mining algorithm. In order to perform activities' recognition, features are extracted and activities are modeled. Experiments are carried out on the Massachusetts Institute of Technology (MIT) smart home data set. The results show the effectiveness of the proposed approach with 99% as recognition rate.

1 INTRODUCTION

The study about human activities and his behaviour is an important research area in computer vision. Nowadays, automatic activities and behaviour understanding have gained great deal of attention. Using machine learning, researchers try to observe a scene, learn prototypical activities and use prototypes for analysis. This approach has been of particular interest for surveillance (Stauffer and Grimson, 2000; Makris and Ellis, 2005) and traffic monitoring (Piciarelli and Foresti, 2006; Atev et al., 2006; Morris and Trivedi, 2008) where methods for categorizing observed behavior, abnormal actions detection for a quick response, even predicting and future occurrences prediction are highly solicited.

Due to the large amounts of data in use for these applications, it is difficult to manually analyze each individually. In these cases, the data mining in general and the Sequential Pattern Mining (SPM) in particular appear as promising solutions. This paper is concerned with SPM in tempoal databases and its application to learn activity of daily living.

This paper is organized as follows. In section 2, we present the state of art and related works on SPM. Section 3 gives a theoretical description of the proposed method while section 4 presents experimental results and analysis. A conclusion ends this work with its future directions.

2 STATE OF ART AND RELATED WORKS ON SPM AND ACTIVITIES LEARNING

The task of sequential pattern mining consists of discovering interesting subsequences in a set of sequences. The sequential ordering of events is considered unlike pattern mining introduced by Agrawal and Srikant (Agrawal and Srikant, 1994) for finding frequent itemsets. The first sequential pattern mining algorithm is called *AprioriAll* (Agrawal and Srikant, 1995). The improved version of this algorithm is Generalized Sequential Pattern algorithm (GSP) (Agrawal and Srikant, 1996). These two algorithms are inspired by the *Apriori* algorithm for frequent itemset mining (Agrawal and Srikant, 1994). GSP algorithm uses a standard database representation, also called a horizontal database and performs a breadth-first search to discover frequent sequential patterns. In recent years, other algorithms have been designed to discover sequential patterns in sequence databases. The SPADE algorithm (Zaki, 2001) inspired by the Eclat algorithm (Zaki, 2000) for frequent itemset mining is an alternative algorithm that uses a depth-first search. It uses the vertical database representation. The vertical representation of a sequence database indicates the itemsets where each item *i* appears in the sequence database (Zaki, 2001; Ayres et al., 2002; Fournier-Viger et al., 2014). For a given item, this information is called the *IDList* of the item. SPAM (Ayres

Aízan, J., Motamed, C. and Ezin, E.
Activity Mining in a Smart Home from Sequential and Temporal Databases.
DOI: 10.5220/0009061105420547
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 542-547
ISBN: 978-989-758-397-1; ISSN: 2184-4313

et al., 2002) is another algorithm that is an optimization of SPADE and also performs a depth-first search using bit vector *IDLists*. Recently, the SPAM algorithm (Ayres et al., 2002) and SPADE algorithm (Zaki, 2001) were improved to obtain the CM-SPAM and CM-SPADE algorithms (Fournier-Viger et al., 2014). The CM-SPAM and CM-SPADE algorithms are both based on the observations that SPAM and SPADE generate many candidate patterns and perform the costly join operation to create the *IDList* of each of them. Besides depth-first search algorithms and vertical algorithms, another important type of algorithms for sequential pattern mining is pattern-growth algorithms. These algorithms are designed to address a limitation of the previously described algorithms by generating candidate patterns that may not appear in the database. In this research work we used CM-SPADE algorithm. The use of this algorithm is motivated by the fact that CM-SPADE is claimed to be the current fastest Sequential Pattern Mining algorithm (Fournier-Viger et al., 2014).

Learning daily activities in a smart home is a real challenge. Schweizer et al. (Schweizer et al., 2015) proposed a frequent sequential pattern mining algorithm to learn consumer behaviour and then reduce energy consumption in smart homes. This algorithm named Window Sliding with De-Duplication (WSDD), uses a window with a prefixed size over the chronologically ordered events to find all possible frequent patterns. The approach does not consider the time between two events. In the same field of energy consumption behaviour analysis, Singh and Yassine in (Singh and Yassine, 2017) proposed an unsupervised progressive incremental data mining mechanism.

(Li et al., 2017) used frequent episode mining to discover sequential behaviour patterns. Suryadevara (Suryadevara, 2017) developed a framework to discover data model for smart home and IoT Data Analytics. Hassani et al. (Hassani et al., 2015) employed a novel sequential pattern mining algorithm called *PBuilder* which uses a batch-free approach to mine activities in a smart home. Menaka and Gayathri (Menaka and Gayathri, 2013) proposed high utility pattern mining to model activity in a smart home. Their approach used linked sensor data stream to save processing time and memory space.

(Moutacalli et al., 2012) used temporal data mining algorithm to model activities. Their approach uses in the mining process the activities temporally segmentation. Raeiszadeh and Tahayori (Raeiszadeh and Tahayori, 2018) proposed a novel method named UP-DM used sequential pattern mining based on the longest common subsequence to model behaviour in smart home.

The main contribution of the paper is the use of an efficient activity recognition approach based on sequential pattern mining, which incorporates feature extraction with temporal information and Random-Forest model (SPM+RandomForest).

3 PROPOSED METHOD

In this work, we use sequential pattern mining to discover typical activities in smart home. The proposed method has three phases namely pre-processing, sequential pattern mining and activity modeling. Fig. 1 presents the proposed approach architecture. For our experimentation, we have used the Massachusetts Institute of Technology (MIT) smart home data set (Tapia et al., 2004). This data set needs to be transformed to a temporal sequential database. The pre-processing represents the first stage of the architecture. The second step extracts typical activities using a sequential pattern mining approach, and the third stage operates on feature extraction and activity modeling based upon temporal constraints.

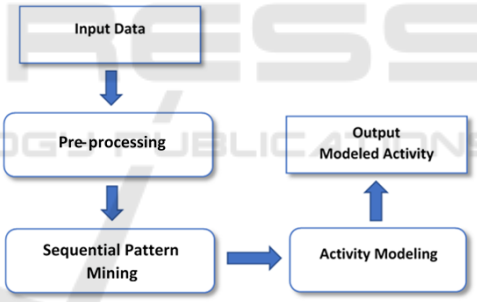


Figure 1: Architecture of the proposed approach.

3.1 Pre-processing

An activity is a time ordered records of events. Events are generated by sensors. The decision about activating an event is linked with the state changes (Boolean) from the sensor or when its value greatly changes numerically. A small change in value is considered as the noise and is therefore ignored. The pre-processing phase aims to convert sensor data into event sequences. For illustration we show the *"Washing dishes"* activity from the dataset in Table 1. In the pre-processing phase as shown in Fig. 2, raw sensor data are converted to $(t)eid$ format in which t represents sensor activation or deactivation timestamp, *eid* represents event id. The event id named *eid* is of the form XYZ where X represents sensor id, Y represents

sensor state which can be 1 if activated or 0 if deactivated. Z represents the number of times the sensor is activated or deactivated during the same activity.

Table 1: Sample of data.

Going out to work	***4/1/2003***	***12:11:26***	***12:15:12***
81	139	140	
Closet	Jewelry box	Door	
12:12:29	12:13:27	12:13:45	
12:13:0	12:13:35	12:13:48	
Toileting	***4/4/2003***	***12:30:17***	***12:31:10***
100	67		
Toilet Flush	Cabinet		
12:30:30	12:30:51		
14:2:12	12:30:54		
Washing dishes	4/5/2003	15:57:55	16:0:15
70	132	132	70
Dishwasher	Cabinet	Cabinet	Dishwasher
15:58:31	15:58:52	15:59:22	15:59:39
15:59:32	15:59:19	15:59:26	16:7:15

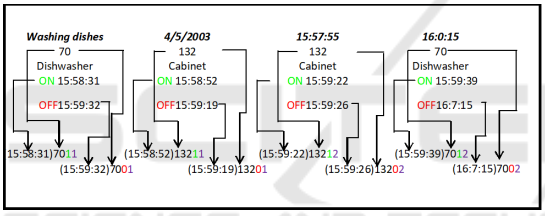


Figure 2: Pre-processing phase of the sensor data.

3.2 Sequential Pattern Mining

The second stage is performed by a sequential pattern mining to obtain frequent sequences.

3.2.1 Definitions

Let $\mathcal{S} = \{1, \cdots, p\}$ and $I = \{i_1, i_2, \cdots, i_m\}$ be respectively a set of sources and a set of items. An event e is a set of items such that $e \subseteq I$. A sequence database $D = \langle s_1, s_2, \cdots, s_p \rangle$ is an ordered list of sequences such that each $s_i \in D$ is of the form (eid_i, e_i, σ_i), where eid_i is a unique event-id, including a timestamp (events are ordered by this timestamp), e_i is an event and σ_i is a source.

A sequence is an ordered list of events $s = \langle e_1, e_2, \cdots, e_n \rangle$ such that $e_k \subseteq I$ $(1 \leq k \leq n)$. A sequence s is said to be of length k or a k-sequence if it contains k items, or in other words if $k = \sum_{j=1}^{n} |e_j|$. A sequence $s_a = \langle A_1, A_2, \cdots, A_n \rangle$ is a subsequence of another sequence $s_b = \langle B_1, B_2, \cdots, B_m \rangle$ denoted $s_a \preceq s_b$, if and only if there exist integers $1 \leq i_1 < i_2 < \cdots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \cdots, A_n \subseteq B_{i_n}$. Let $D_i = \{e | (eid, e, i) \in D\}$ be the sequence corresponding to a source i ordered by eid. For a sequence s and a source i, let $X_i(s, D)$ be an indicator variable, whose value is 1 if s is a subsequence of a sequence D_i, and 0 otherwise. For any sequence s, its support in D is denoted by $Sup(s, D) = \sum_{i=1}^{p} X_i(s, D)$. The goal is to find all sequences s such that $Sup(s, D) \geq \theta p$ for some user-defined threshold $0 \leq \theta \leq 1$.

A vertical database $V(D)$ is a database in which each entry represents an item and indicates the list of sequences where the item appears and the position(s) where it appears.

A sequence $s_a = \langle A_1, A_2, \cdots, A_n \rangle$ is a prefix of a sequence $s_b = \langle B_1, B_2, \cdots, B_m \rangle$, $\forall n < m$, if and only if $A_1 = B_1, A_2 = B_2, \cdots, A_{n-1} = B_{n-1}$ and the first $|A_n|$ items of Bn according to the lexicographical order are equal to A_n.

3.2.2 CM-SPADE Algorithm

Algorithm 1 presents the pseudocode of CM-SPADE algorithm (Fournier-Viger et al., 2014). It takes a sequence database D and *minsup* threshold as input. CM-SPADE first constructs the vertical database $V(D)$ and identifies the set of frequent sequential patterns F1 containing frequent items. Then, SPADE calls the ENUMERATE procedure with the equivalence class. The ENUMERATE procedure receives an equivalence class F as parameter. Each member A_i of the equivalence class is a frequent sequential pattern. Then, a set T_i, representing the equivalence class of all frequent extensions of A_i is initialized to the empty set. Then, for each pattern $A_j \in F$ such that $j \geq i$, the pattern A_i is merged with A_j to form larger patterns. For each such a pattern r, the support of r is calculated by performing a join operation between *IdLists* of A_i and A_j. The function *Prune* in (Fournier-Viger et al., 2014) uses co-occurrence pruning approach. If the cardinality of the resulting *IdList* is not less than *minsup*, it means that r is a frequent sequential pattern. It is thus added to T_i. Finally, after all pattern A_j have been compared to A_i, the set T_i contains the whole equivalence class of patterns starting with the prefix A_i. The procedure ENUMERATE is then called with T_i to discover larger sequential patterns having A_i as prefix. When all loops terminate, all frequent sequential patterns have been output.

3.3 Feature Extraction and Activity Modeling

In this phase, we build an activity model based on features of the activities and RandomForest model. In

Algorithm 1: The pseudocode of CM-SPADE.

```
1: procedure CM-SPADE(D,minsup)
2:     for all d ∈ D do
3:         create V(D)
4:         identify F1 the list of frequent items
5:     end for
6:     ENUMERATE(F1)
7: end procedure

8: procedure ENUMERATE(an equivalence class F)
9:     for all pattern A_i ∈ F do
10:        Output A_i
11:        T_i ← ϕ
12:        for all pattern A_j ∈ F, with j ≥ i do
13:            R ← MergePatterns(A_i, A_j)
14:            if Prune(R) = FALSE then
15:                if sup(R) ≥ minsup then
16:                    T_i ← T_i ∪ {R}
17:                end if
18:            end if
19:        end for
20:        ENUMERATE(T_i)
21:    end for
22: end procedure
```

addition to which sensors fired, temporal information would be necessary to achieve good recognition. The used features are as follows:

- **Activity Start Time:** The start time of an activity is one of the distinctive features for activity recognition. Based upon the start time, there are four periods as depicted in Fig. 3. These periods are labeled as shown in Table 2.
- **Activity Duration:** According to their duration, activities can be clustered into four classes as illustrated in Fig. 4. These four classes are labeled as represented in Table 3.
- **Density of Events:** The numbers of sensor events in an activity depends on the duration and mobility. We use event density to capture this feature. To calculate the value of an event density, the number of reported events for an activity is divided by the activity duration as expressed in (1).
- **Previous Activity:** The activity previously performed may provide a clue in recognizing the current activity.

$$Event\_density = \frac{Number\,of\,events}{Duration\,of\,activity} \qquad (1)$$

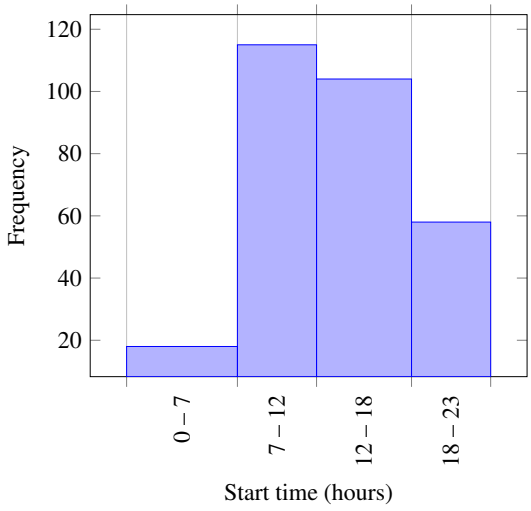


Figure 3: Frequency of activities along their start time for subjet 1 dataset.

Table 2: Activity's label according to its start time.

Start time interval (hours)	Label
$0 \leq time < 7$	Night
$7 \leq time \leq 12$	Morning
$12 < time \leq 18$	Afternoon
$18 < time \leq 23$	Evening

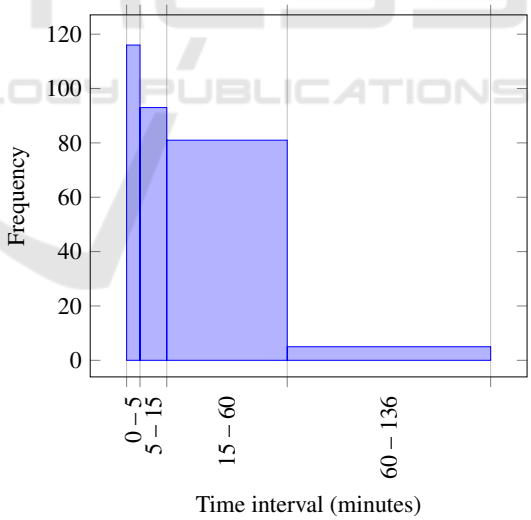


Figure 4: Frequency of activities along their duration for subjet 1 dataset.

Table 3: Labelling activities based on their duration.

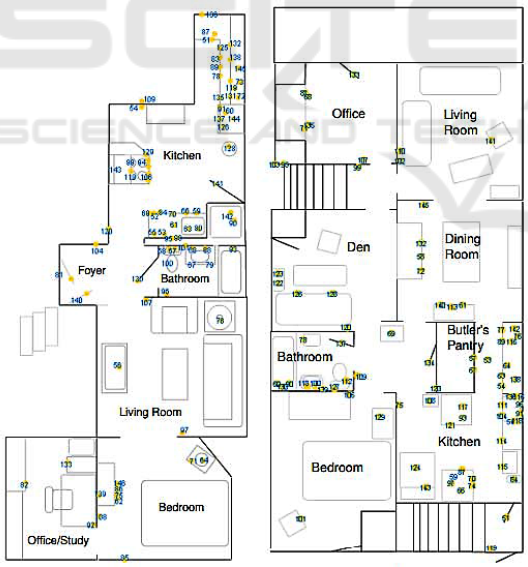
Time interval (minutes)	Label
$duration \leq 5$	Ultra-Short
$5 < duration \leq 15$	Short
$15 < duration \leq 60$	Medium
$duration > 60$	Long

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present the results obtained with the proposed method. We used the MIT data smart home testbed better described in subsection 4.1.

4.1 MIT Dataset

MIT dataset is a collection of human activity for two weeks in two single-person apartments containing respectively 77 and 84 sensors (see Fig. 5 for illustration). The first subject was a professional woman of 30 year old who lived in the apartment shown in Fig. 5(a) and spent her free time at home while the second subject was a woman of 80 year old who spent most of her time at home and lived in the apartment shown in Fig. 5(b). The sensors were installed in everyday objects such as drawers, refrigerators containers, etc. to record opening-closing events (activation deactivation events) as the subject carried out everyday activities. Activities are labeled into 16 different classes and the number of occurrences of each class by subject is showed in Table 4.



(a) (b)

Figure 5: (a) Apartment of subject one. (b) Apartment of subject two.

4.2 Results and Analysis

Our implementation in Java, is executed on a machine Intel(R) Core(TM) $i7-7500U$ CPU @2.70 GHz 2.90 GHz running on Windows 10. With a support value fixed to 0.8, our method discovered 30 sequential frequent patterns, with the lengths spanning from 1 to 11 events for subject 1 and 39 sequential frequent patterns, with the lengths spanning from 1 to 6 events for subject 2 when we use sequential pattern mining algorithm. This result shows that, sequential pattern mining algorithm return typical activities. We use RandomForest classification model, to recognize future activities of the users and obtained the accuracy level of 99.38% in this model for the first subject and 95.45% for the second subject. By returning useful and frequent pattern, our approach reduce activities features vectors dimension and then clearly performs better than the approach proposed by Raeiszadeh and Tahayori in (Singh and Yassine, 2017) (see Table 5).

Table 4: Activity label.

Number of Examples per Class		
Activity	**Subject 1**	**Subject 2**
Preparing dinner	8	14
Preparing lunch	17	20
Listening to music	-	18
Taking medication	-	14
Toileting medication	85	40
Preparing breakfast	14	18
Washing dishes	7	21
Preparing a snack	14	16
Washing TV	-	15
Bathing	18	-
Going out to work	12	-
Dressing	24	-
Grooming	37	-
Preparing a beverage	15	-
Doing laundry	19	-
Cleaning	8	-

5 CONCLUSIONS

We have used a sequential pattern mining algorithm from temporal databases to bring out typical activities in the smart home. We use temporal relationships between events for a more accurate characterization/classification of frequent activities.

In the future work, we will consider sensor uncertainty to focuse on reliable parts of the sensor data. So we will use activity recognition approach based on uncertain sequential pattern mining algorithm.

REFERENCES

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *The International Conference on Very Large Databases, pp. 487-499.*

Table 5: Comparison of results.

	Approach	Result
Proposed Method	(SPM+RandomForest)	Subject 1: 99.38%Subject 2: 95.45%
(Raeiszadeh and Tahayori, 2018)	(UP-DM+RandomForest)	Subject 1: 97.45%Subject 2: 91.37%
(Tapia et al., 2004)	(Naive Bayes Classifier)	Subject 1: 60.6% Subject 2: 41.09%

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *The International Conference on Data Engineering, pp. 3-14*.

Agrawal, R. and Srikant, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *The International Conference on Extending Database Technology, pp. 1-17*.

Atev, S., Masoud, O., and Papanikolopoulos, N. (2006). Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *Conf. Intell. Robots and Systems, Bejing, China, pp. 4851–4856*. IEEE.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *International Conference on Knowledge Discovery and Data Mining, pp. 429-435*. ACM SIGKDD.

Fournier-Viger, P., Gomariz, A., Campos, M., and Thomas, R. (2014). Fast vertical mining of sequential patterns using co-occurrence information. In *The Pacic-Asia Conference on Knowledge Discovery and Data Mining, pp. 40-52*.

Hassani, M., Beecks, C., ows, D. T., and Seidl, T. (2015). Mining sequential patterns of event streams in a smart home application. In *The LWA 2015 Workshops: KDML, FGWM, IR, and FGD*.

Li, L., Li, X., Lu, Z., Lloret, J., and Song, H. (2017). Sequential behavior pattern discovery with frequent episode mining and wireless sensor network. In *Communications Magazine*. IEEE.

Makris, D. and Ellis, T. (2005). Learning semantic scene models from observing activity in visual surveillance. In *Trans. Syst., Man, Cybern. B, vol. 35, no. 3, pp. 397–408*. IEEE.

Menaka, J. and Gayathri, K. S. (2013). Activity modeling in smart home using high utility pattern mining over data streams. In *The Journal of Computer Science and Network*.

Morris, B. T. and Trivedi, M. M. (2008). Learning, modeling, and classification of vehicle track patterns from live video. In *Trans. Intell. Transp. Syst., vol. 9, no. 3, pp. 425–437*. IEEE.

Moutacalli, M. T., Bouzouane, A., and Bouchard, B. (2012). Unsupervised activity recognition using temporal data mining. In *The First International Conference on Smart Systems, Devices and Technologies*.

Piciarelli, C. and Foresti, G. L. (2006). On-line trajectory clustering for anomalous events detection. In *Pattern Recognition Letters, vol. 27, no. 15, pp. 1835–1842*.

Raeiszadeh, M. and Tahayori, H. (2018). A novel method for detecting and predicting resident's behavior in smart home. In *6th Iranian Joint Congress on Fuzzy and Intelligent Systems*. IEEE.

Schweizer, D., Zehnder, M., Wache, H., and Witschel, H. (2015). Using consumer behavior data to reduce energy consumption in smart homes. In *14th International Conference on Machine Learning and Applications*.

Singh, S. and Yassine, A. (2017). Mining energy consumption behavior patterns for house holds in smart grid. In *Transactions on Emerging Topics in Computing*. IEEE.

Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. In *Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 747–757*. IEEE.

Suryadevara, N. (2017). Wireless sensor sequence data model for smart home and iot data analytics. In *First International Conferenceon Computational Intelligence and Informatics, Advances in Intelligent Systems and Computing*.

Tapia, E. M., Intille, S. S., and Larson, K. (2004). Activity recognition in the home setting using simple and ubiquitous sensors. In *Pervasive Computing*.

Zaki, M. J. (2000). Scalable algorithms for association mining. In *Transactions on Knowledge and Data Engineering, vol. 12(3), pp. 372-390*. IEEE.

Zaki, M. J. (2001). Spade: An effcient algorithm for mining frequent sequences. In *Machine learning, vol. 42(1-2), pp. 31-60*.