Contextualized behavior patterns for ambient assisted living

Paula Lago, Claudia Jiménez-Guarín, Claudia Roncancio

Contextualized behavior patterns for ambient assisted living

Paula Lago[1], Claudia Jiménez-Guarín[1], Claudia Roncancio[2]

[1]Systems and Computing Engineering Department, School of Engineering, Universidad de Los Andes, Colombia
{pa.lago52,cjimenez}@uniandes.edu.co
[2]Univ. Grenoble, LIG, France
claudia.roncancio@imag.fr

Abstract. Human behavior learning plays an important role in ambient assisted living since it enables service personalization. Current work in human behavior learning do not consider the context under which a behavior occurs which hides some behaviors that are frequent only under certain conditions. In this work, we present the notion of a contextualized behavior pattern which describes a behavior pattern with the context on which it occurs (i.e. nap when raining) and propose an algorithm for finding these patterns in a data stream. This is our main contribution. These patterns help to better understand the routine of a user in a smart environment, as is evidenced when testing with a public dataset. This algorithm could be used to learn behaviors from users in an ambient assisted living environment in order to send alarms when behavior changes occur.

Keywords: human behavior learning, sequential patterns, context analysis, personalization, data streams

1 Introduction

Human behavior learning plays an important role in ambient assisted living since it enables service personalization like sending alerts when changes in behavior are detected. For example, if a user normally takes a walk after lunch but lately she has been napping instead of walking, an alert could be sent to her family to check if everything is fine.

Behavior learning is widely studied and different behavior models have been proposed [1–4]. One such model is sequential patterns in which a behavior is represented as an ordered set of activities. Thus, sequential pattern mining algorithms can be used to learn frequent behaviors of a user. However, most of these algorithms do not consider the context on which a behavior occurs, which can make the applications that use them: (1) miss some behaviors that are frequent only under certain context (i.e. only on weekends) and (2) generate false alarms about behavior changes. To continue with the first example, if it has been raining lately and the user likes to nap when it is raining, then skipping the walk after lunch is normal and should not be notified. A high false alarm rate can cause users to be reluctant to use these systems or to ignore all alarms, so they should be avoided.

A behavior learning algorithm for ambient assisted living should consider context when learning frequent patterns to find those frequent only under certain conditions. It should also learn behaviors in an online fashion (as data arrives from sensors) and be adaptive, that is learn new behaviors as they appear.

In this paper, our goal and main contribution is to find sequential patterns of activities that are frequent over a specific day, time or after a specific activity (section 2). We present the definition of a contextualized sequential pattern and propose an algorithm for their extraction. We test the algorithm with a public database and show that considering context yields different patterns than just mining frequent patterns over the entire period of time (section 3). Finally we present related work (section 4) and conclusions and future work (section 5).

2 Proposed Algorithm

In this section we first introduce the notion of a contextualized sequential pattern and then present the data structure and proposed algorithm for mining them from a data stream. We consider day, hour and activity as part of the context but more variables will be included in future work.

2.1 Definitions

Given a data stream of activities S_a, each with a timestamp, a sequence is an ordered (by timestamp) set of activities $a_1, a_2, \dots a_n$. A sequence s occurs at the day of a_1's timestamp and starts at the hour indicated by it. In this work, a context c is a specific day (i.e. Tuesday), a specific time (i.e. at 9:00am) or a specific activity (i.e. sleeping) or the combination of any of them. A sequence s is said to support a context if it occurs at the day, hour or starts with the activity specified by c. We denote $occurs(s, c)$ to indicate that sequence s occurs at context c.

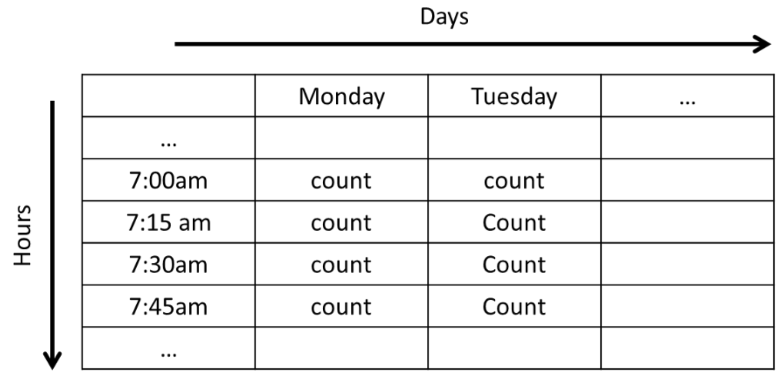
A contextualized sequential pattern is a sequence of activities that frequently appear together over a specific context. The pattern describes both the sequence and the context on which it occurs. A sequence is said to be frequent in a context if the number of times it occurs on the specific context over the total number of sequences that support the context is greater than a user-specified minimum threshold (φ). This is expressed by equation 1.

$$\frac{|occurs(s,c)|}{\sum_j |occurs(s_j,c)|} \geq \varphi \qquad (1)$$

2.2 Contextualized prefix-tree

For mining the contextualized sequential patterns we use a contextualized prefix tree, which is an extension of the prefix-tree used for prefix span [5]. The root of the tree keeps the total count of sequences that have been seen and also the counts by day and time in a context count matrix (**Fig. 1**). In this work, we divide time in 15-minute slots (96 timeslots in total for each day), but this could be configured as a parameter for the algorithm. Each cell in the matrix indicates how many sequences have supported both the day and the time represented by the cell.

Each node of the tree represents a sequence, which is the path from the root to the node. A node has a label (indicating an activity label), a total count of the times this sequence has been seen and a context count matrix, which counts how many times the sequence has been seen in each day and hour.



	Monday	Tuesday	...
...			
7:00am	count	count	
7:15 am	count	Count	
7:30am	count	Count	
7:45am	count	Count	
...			

Fig. 1. Context count matrix, a 7x96 int matrix that stores the number of times a sequence has been seen on each day and timeslot.

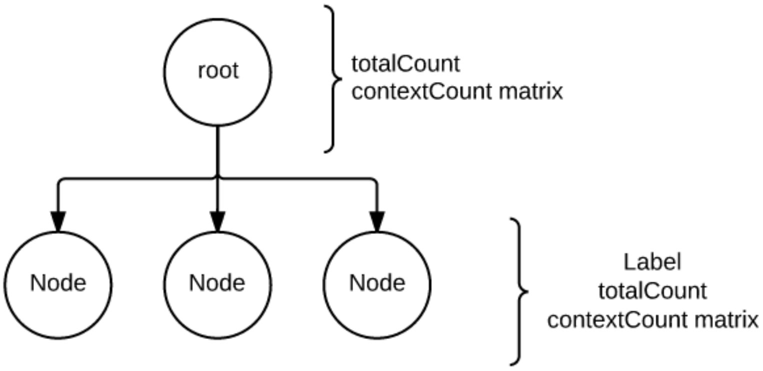


Fig. 2. Contextualized prefix-tree

2.3 Algorithm

The algorithm for extracting frequent contextualized patterns consists of three steps: segmentation of the stream, insertion in the tree and extraction of frequent sequences. Each step is explained next considering the following stream of activities as example:

$$
\begin{aligned}
&S_a = \{(2010 - 11 - 07\ 15{:}40{:}57.80\ \ Meal_{Preparation}),\\
&(2010 - 11 - 07\ 15{:}42{:}09.46\ \ Eating),\\
&(2010 - 11 - 07\ 17{:}19{:}51.47\ \ Meal_{Preparation}),\\
&(2010 - 11 - 07\ 17{:}30{:}26.96\ \ Eating),\\
&(2010 - 11 - 07\ 17{:}40{:}51.0\ \ Wash_{Dishes}),\\
&(2010 - 11 - 07\ 17{:}47{:}05.0\ \ Meal_{Preparation})\\
&(2010 - 11 - 07\ 19{:}52{:}24.28\ \ Work),\\
&(2010 - 11 - 07\ 20{:}37{:}41.43\ \ Eating),\\
&(2010 - 11 - 07\ 20{:}40{:}41.5\ \ Meal_{Preparation}),\\
&(2010 - 11 - 07\ 23{:}52{:}03.47\ \ Sleeping)\}
\end{aligned}
$$

Segmentation

First, the stream of activities is segmented to form sequences of equal length (n). From the timestamp of each activity in the sequence, the day of the week and the timeslot are extracted. Each time a sequence of length n is formed (n activities have entered the stream), it is inserted in the tree. The length of the sequences n can be configured as a parameter. Having a very small length results in uninteresting patterns and having a large length could result in many spurious patterns since randomness could make small sequences frequent. Statistical experts advise against large n.Therefore, in this work we use a sequence length of 4.

The stream of the example, would be segmented in the following sequences (the first integer represents the day of the week and the second integer represents the timeslot of the first activity in the sequence, for example the first sequence has time 62 because meal preparation starts at 15:40 which corresponds to the timeslot 62):

$$
\begin{aligned}
&(2, 62, MealPreparation, eating, mealPreparation, eating)\\
&(2{,}62, eating, mealPreparation, eating, washDishes)\\
&(2{,}69, mealPreparation, eating, washDishes, mealPreparation)\\
&(2{,}70, eating, washDishes, mealPreparation, work)\\
&(2{,}70, washDishes, mealPreparation, work, eating)\\
&(2{,}71, mealPreparation, work, eating, mealPreparation)\\
&(2{,}79, work, eating, mealPreparation, sleeping)
\end{aligned}
$$

Insertion

To insert a sequence in the tree, the root checks if it has the first activity of the sequence in its children. If it does, then the totalCount and context counts are updated for this node. If it doesn't, then a new node is created with totalCount 1 and the corresponding context count in 1 also. Recursion is used for each element in the sequence.

Following our example, the first sequence is inserted as the first branch of the empty tree with total count = 1 and the corresponding context count = 1(Tuesday, 62) as shown in **Fig. 3**. Then, the second sequence is inserted as a second branch like in **Fig. 4**, updating the root's total counts. Finally, when all sequences have been inserted the tree looks like **Fig. 5**

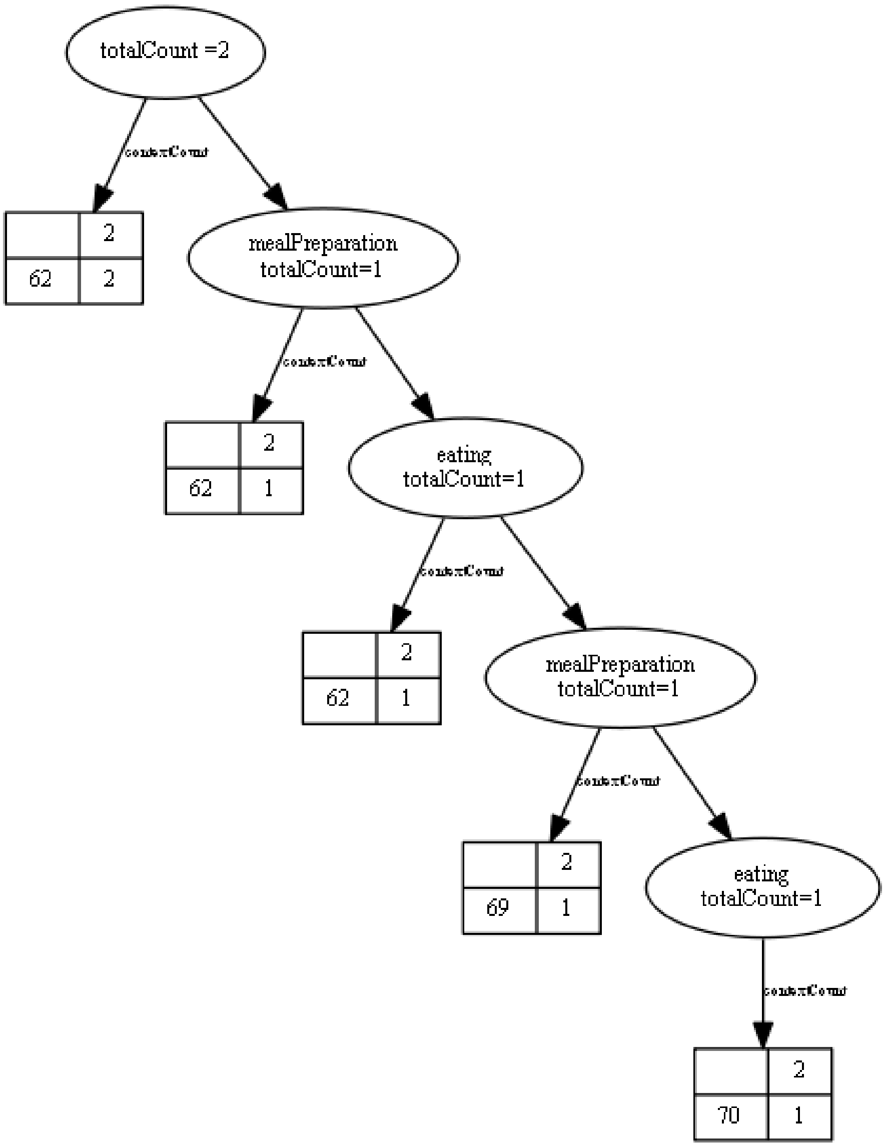


Fig. 3. The prefix tree after the insertion of first sequence

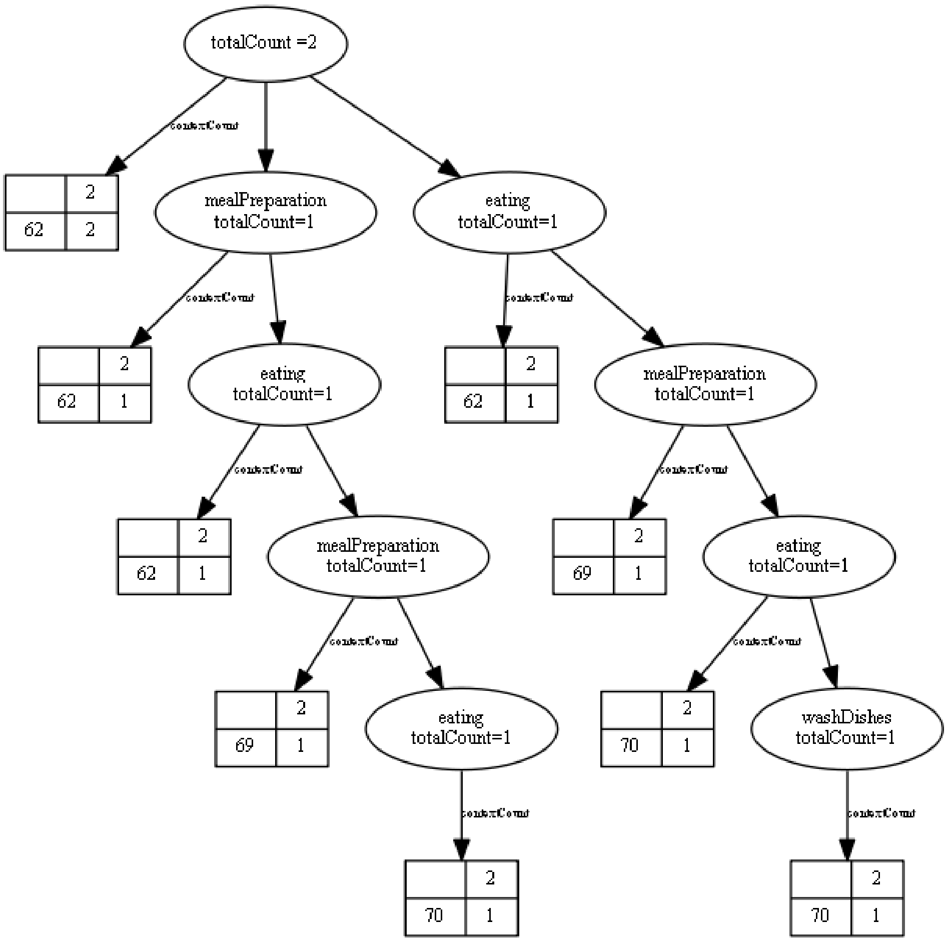


Fig. 4. Prefix tree after the insertion of second branch

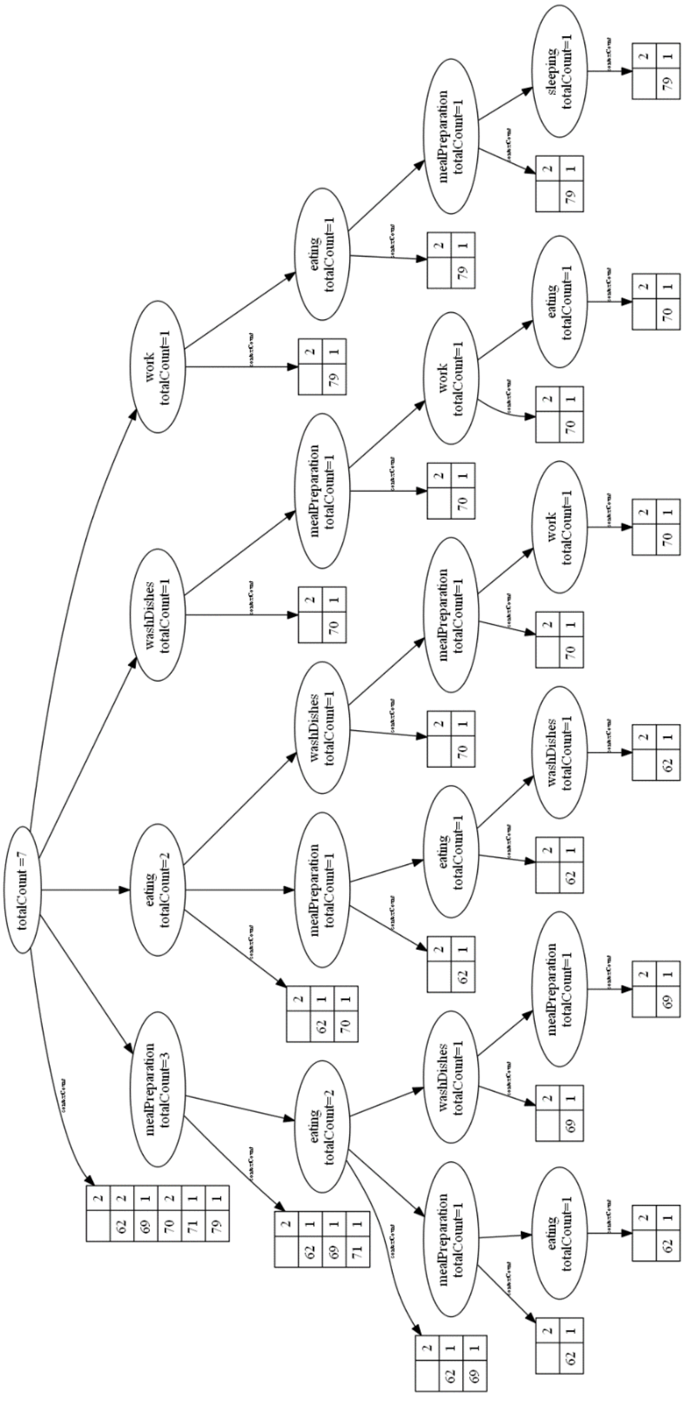


Fig. 5. Prefix tree after the insertion of all sequences

Extraction of frequent sequences

To extract frequent sequences by context, the total count of the context $t_c$is obtained from the root node. If patterns are to be extracted by only day or time then the total count is obtained by summing the corresponding row or column of the root node count matrix. Then, each node checks if its context count is greater than the total count of the context times the minimum support ($t_c * \varphi$). If it is, then the sequence it represents is added to the frequent sequence set.

In the tree of the example, with a minimum support of 0.45, the patterns extracted without considering context would only be meal preparation since it is the only sequence (length 1) that has a support greater than the threshold. Considering context we can obtain some of the following patterns:

- At timeslot 62 (around 15:30), it is frequent to prepare meal and then eat.
- At timeslot 70 (around 17:30), it is frequent to eat and then wash dishes.
- It is frequent to eat after work.

Although these toy example has too few data to be really interesting (many sequences have a total count of only 1), it shows that not considering context yields different patterns than considering it and that our algorithm can find patterns that are associated to a time and a day.

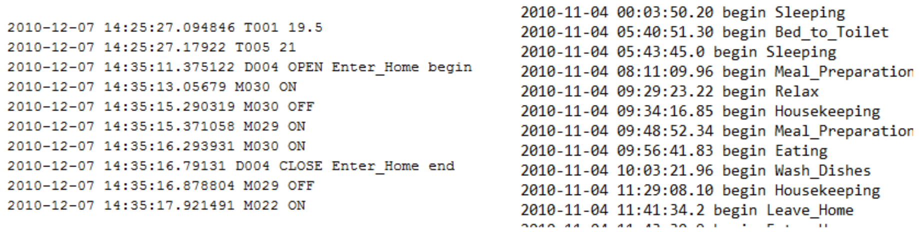
3 Results

We tested our algorithm with the Aruba CASAS Dataset, made public by the CASAS project [6] through their repository[1]. The dataset consists of sensor data that was collected in the home of a volunteer adult with annotated activities. We used this dataset because it has data for a long period of time (193 days) and it is collected in real-life settings. The annotated activities in this dataset are: meal preparation, sleeping, relax, wash dishes, work, bed to toilet, enter and leave home, housekeeping and resperate[2].

Each record in the dataset contains a timestamp, a sensor id and a sensor reading. If the event corresponds to the start or end of an activity, then the record also contains the activity label and the word start or end accordingly (see Fig. 6 (a)). Since we are concerned with finding sequential patterns of activities and not of sensor activations, we first transformed the dataset in order to have records that contain the start and end time of the activity, the location and the label (see Fig. 6 (b)). Also, because in the dataset it is common to have the same activity begin and end at very close times, we consider only an occurrence of the activity if no more than fifteen minutes have passed from the end of one occurrence to the beginning of the next.

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

[2] Resperate is a device used to lower blood pressure



```
2010-12-07 14:25:27.094846 T001 19.5
2010-12-07 14:25:27.17922 T005 21
2010-12-07 14:35:11.375122 D004 OPEN Enter_Home begin
2010-12-07 14:35:13.05679 M030 ON
2010-12-07 14:35:15.290319 M030 OFF
2010-12-07 14:35:15.371058 M029 ON
2010-12-07 14:35:16.293931 M030 ON
2010-12-07 14:35:16.79131 D004 CLOSE Enter_Home end
2010-12-07 14:35:16.878804 M029 OFF
2010-12-07 14:35:17.921491 M022 ON
```

(a)

```
2010-11-04 00:03:50.20 begin Sleeping
2010-11-04 05:40:51.30 begin Bed_to_Toilet
2010-11-04 05:43:45.0 begin Sleeping
2010-11-04 08:11:09.96 begin Meal_Preparation
2010-11-04 09:29:23.22 begin Relax
2010-11-04 09:34:16.85 begin Housekeeping
2010-11-04 09:48:52.34 begin Meal_Preparation
2010-11-04 09:56:41.83 begin Eating
2010-11-04 10:03:21.96 begin Wash_Dishes
2010-11-04 11:29:08.10 begin Housekeeping
2010-11-04 11:41:34.2 begin Leave_Home
```

(b)

Fig. 6. (a) Original dataset and, (b). Transformation done for this paper. We used data that defines the start time and label for each activity.

We first find sequential patterns without considering context using a minimum support of 0.4. On this dataset the patterns found are: relax, meal_preparation. Only these two sequences are found to be frequent because there are many more occurrences of these two activities than of the others in the dataset. We omit the patterns that only contain these two activities from the following results because of this.

We then find time patterns and patterns based on each of the activities (i.e. what is usually done after eating?). These patterns allow a better understanding of the routine of the person that lived in the smart environment. Some of the patterns found are described below.

- From 21:30 to 1:00 the only frequent activity is Sleeping. This indicates the usual sleeping hour of the person.
- From 1:00 to 4:45, Sleeping is a frequent activity as well as the sequence Sleeping – Bed to toilet – Sleeping. This indicates that the person wakes up at night frequently.
- At 6:00 am Meal Preparation and Relax are frequent activities, which could mean that around this time the user wakes up.
- At 6:45 the sequence Meal_Preparation – Relax - Work is frequent.
- At 20:45 the sequences Relax - Meal_Preparation – Relax - Work and Relax-Meal_Preparation – Relax - Wash_Dishes are frequent.

Patterns by activity include the following:

- Resperate – Relax - Meal_Preparation
- Wash_Dishes -Relax – Sleeping. This sequence can be interpreted as the user usually washing dishes at night, before going to sleep.
- Sleeping – Bed to Toilet – Sleeping. This is an indication of waking up frequently at night.
- Sleeping - Meal_Preparation – Relax - Meal_Preparation – Eating. These sequences indicates that shortly after waking up, the person prepares meal and then eats (breakfast).

These results are consistent with patterns found with prefixspan when data is partitioned to use only activities that occurred in a specific day or time.

3.1 Using the tree for activity prediction

The contextualized prefix tree can be used for activity prediction by obtaining the most probable next activity according to the current context (day, hour and activity). Using the same dataset, we use the tree for activity prediction by predicting the most probable activity as the data arrives. Table 1 shows the precision and recall for all the activities in the dataset and Table 2 shows the confusion matrix obtained.

Some activities show good results such as meal preparation, enter home, bed to toilet, sleeping and relax, but the others have very poor performance. We believe that if the patterns are adapted (removing old patterns) these results could be improved.

Table 1. Precision and Recall for each activity in the dataset using activity prediction

	Precision	Recall
Meal_Preparation	0,54	0,48
Sleeping	0,44	0,98
Wash_Dishes	0,23	0,17
Resperate	0,00	0,00
Work	0,02	0,06
Enter_home	0,96	0,79
Leave_home	0,23	0,28
Bed_to_Toilet	0,50	0,37
Housekeeping	0,00	0,00
Relax	0,69	0,64
Eating	0,31	0,33

Table 2. Confusion matrix for activity prediction

	Meal_ Preparation	Sleeping	Wash_ Dishes	Resperate	Work	Enter_ home	Leave_ home	Bed_to_ Toilet	House keeping	Relax	Eating	Could not predict
Meal_Preparation	545	3	16	1	11	30	138	125	26	94	12	2
Sleeping	187	170	4	0	1	1	4	0	3	9	4	2
Wash_Dishes	19	0	15	0	0	0	4	0	0	25	0	1
Resperate	3	0	0	0	0	0	1	0	0	1	1	0
Work	60	0	3	0	2	7	14	0	9	32	6	0
Enter_home	1	0	0	0	1	317	6	0	0	3	0	1
Leave_home	160	0	12	1	5	21	99	1	8	110	12	2
Bed_to_Toilet	76	0	0	0	0	0	0	78	0	0	0	1
Housekeeping	14	0	1	0	1	0	3	0	0	7	5	2
Relax	67	1	35	1	11	22	75	8	8	742	102	4
Eating	13	0	0	0	3	1	4	0	5	129	70	1

3.2 Discussion and analysis of results

Mining contextualized sequential patterns helps to better understand user behavior at home. Using a contextualized prefix tree with a sequence length of 4 we found sensible results, although the parameter could be better configured with further experiments. The contextualized patterns can be used for activity prediction as was tested in this paper, even though the results must be further improved for use in application. The patterns can also be used for measuring behavior drifts which can be early signs of disease or could be due to new routines. Measuring these deviations and differentiating between alerts and new habits that should be added to the patterns is one of the next steps in our research.

4 Related Work

Learning frequent behaviors of users has been recognized as a crucial aspect for realizing the ambient assisted living as well as the more general smart environments paradigms. Learning frequent behaviors enables service personalization, future behavior prediction which can be used for the environment to act proactively and to detect abnormalities or behavior shifts which can be used to generate alerts. Different models have been used for describing and learning human behaviors such as context-free grammars [7, 8], Hidden Markov Models [9, 10], topic models [11, 12] and sequential patterns [13, 14]. Among these models, we select sequential patterns since they are easily understandable by end users and can be used to create readable alerts.

Another key feature in ambient assisted living is context-awareness because it enables the adaptation of services to the circumstances in which they are used [15]. Typical context variables used in smart environments describe both the environment (humidity, noise, lighting levels) and the user (location, physiological variables, posture, activity). In our work, we consider activity, time and weekday as context variables since we believe behavior can change according to these two features. In future work, we will add weather and other typical contextual variables.

Our main objective in this paper is to learn frequent sequential patterns from data streams under different contexts, day and time of day in this work, to better understand the behavior of users. Therefore, we focus on sequential pattern mining with contextual aspects for behavior learning. Surveys of sequential pattern mining can be found in [16, 17] and a survey of algorithms for incremental and progressive sequential pattern mining can be found in [18].

The algorithm Evolving Agent behavior Classification based on Distributions of Relevant events (EVABCD) [3] creates evolving user profiles that are represented by a sequence of events. To solve the problem of unbounded sequences, this algorithm first segments the sequence of events in subsequences of equal length and then stores them in a trie. Each node of the trie stores the number of the times an event (a linux command in the article) has been seen, so calculating support for each sequence is easy. Each user profile is represented with a probability distribution of the frequent sequences. When a sequence becomes less probable it is eliminated from the profile, thus making it an evolving profile. This method not only mines frequent sequences but it also is adaptive, one of the main features of our proposal. One difference with our problem is that we maintain frequent sequences for each user and that EVABCD only takes into account the events and not the context in which they appear, nor the temporal relations among them.

The SPEDS (Sequential Patterns on Evolving data streams) algorithm [19] obtains the set of sequential patterns over a period of time at any moment. It uses a lexicographic tree in which each node denotes an item and a sequence is represented by a path in the tree. Each node contains a tilted time table that counts the frequency of the item on the corresponding tilted-time window. The tree is incrementally updated after a batch of data is processed with prefixSpan [5] to extract frequent sequences from the batch, and update nodes and counters in the tree. This algorithm allows finding patterns at different time granularities which is an idea in which we base our proposal. We extend the tilted-time table to consider other contextual variables.

The MobileMiner [14] service discovers personalized frequent co-ocurrence patterns indicating which context events frequently occur together. Since the service runs on mobile phones it is efficient and could be used for mining streams. However, the service does not mine sequences, and thus, does not consider the order in which events occur. We believe that in some cases, the order of activities can be crucial for detecting emergency or must-watch situations. Another difference with our approach is that they use an apriori-based strategy for mining patterns, while we use the prefix-growth strategy.

Moshtaghi, Zukerman and Rusell's model [20] combines the time of day with the region of the house to infer inactivity thresholds that are used to detect abnormal inactivity periods to generate alerts. This approach is similar to ours in that it learns different behaviors according to context (location and time) but it differs from our research in that they are considering inactivity as the only interesting behavior while we would like to learn sequential patterns to describe user behaviors. This is because although there could be activity at home, painting at night could be normal while going out or wandering around could not.

We describe sequential patterns with the context in which they occur, i.e. if a sequence is frequent when it is raining. In this case, context can be seen as a constraint for the sequential patterns. Although some algorithms have been proposed for mining sequential patterns with constraints [21], these constraints must be defined a priori by the user. Our algorithm can discover the context on which the sequences are frequent, as it is done in some other work. For example, in [22], the time period over which a itemset is frequent is automatically discovered. This is done by pattern-growth and by maintaining for each candidate itemset, the period on which it might be frequent. This work has some differences with ours: (1) they are not mining sequential patterns and (2) their algorithm is not online, meaning it is not suitable for data stream processing as is the case in ambient assisted living (mining data coming from sensors).

5 Conclusions

In this work we propose the notion of contextualized sequential patterns, sequential patterns of activities that are associated to the context on which they are frequent, and an algorithm for mining them on data streams. These patterns can describe the behavior of a user in a smart environment and help to better understand the routine of a user. These patterns can be used for service personalization.

Although some algorithms for mining adaptive sequential patterns on streaming data have been proposed, they do not consider context and the algorithms that do consider context do not mine frequent sequences but itemsets. Our algorithm can find patterns that state that a behavior occurs on Mondays but not on other days or at 10 am but not in the afternoon. This, to the best of our knowledge, has not been achieved, as is also stated in [17]. This is the main contribution of this work.

5.1 Future work

In future work we will add more context variables such as weather and physiological variables to the patterns. Also, merging the context under which the same patterns occurs (i.e. sleeping is frequent from 23:00 to 06:00 on all days) and finding patterns that join different contexts (i.e. patterns occurring on Mondays at 7am) is an important research direction that we will explore. Finally, using these patterns to detect changes in behavior that could signal health deterioration or that an adaptation is needed in the patterns is a crucial step for the application of the patterns in ambient assisted living.

References

1. Monekosso, D.N., Remagnino, P.: Behavior Analysis for Assisted Living. IEEE Trans. Autom. Sci. Eng. 7, 879–886 (2010).
2. Díaz Rodríguez, N., Cuéllar, M.P., Lilius, J., Delgado Calvo-Flores, M.: A fuzzy ontology for semantic modelling and recognition of human behaviour. Knowledge-Based Syst. 66, 46–60 (2014).
3. Iglesias, J.A., Angelov, P., Ledezma, A., Sanchis, A.: Creating Evolving User Behavior Profiles Automatically. IEEE Trans. Knowl. Data Eng. 24, 854–867 (2012).
4. Chua, S., Marsland, S.: Unsupervised Learning of Human Behaviours. Twenty-Fifth AAAI Conf. 319–324 (2011).
5. Pei, J.P.J., Han, J.H.J., Mortazavi-Asl, B., Pinto, H., Chen, Q.C.Q., Dayal, U., Hsu, M.-C.H.M.-C.: PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth. Proc. 17th Int. Conf. Data Eng. (2001).
6. Cook, D.J., Crandall, A.S., Thomas, B.L., Krishnan, N.C.: CASAS : A Smart Home in a Box. IEEE Comput. 46, 62–69 (2013).

7. Turaga, P., Member, S., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine Recognition of Human Activities : A Survey. 18, 1473–1488 (2008).

8. Ryoo, M.S., Aggarwal, J.K.: Recognition of composite human activities through context-free grammar based representation. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2, 1709–1716 (2006).

9. Ordonez, F.J., Englebienne, G., de Toledo, P., van Kasteren, T., Sanchis, A., Krose, B.: In-Home Activity Recognition: Bayesian Inference for Hidden Markov Models. IEEE Pervasive Comput. 13, 67–75 (2014).

10. Forkan, A.R.M., Khalil, I., Tari, Z., Foufou, S., Bouras, A.: A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. Pattern Recognit. 48, 628–641 (2014).

11. Rieping, K., Englebienne, G., Kröse, B.: Behavior analysis of elderly using topic models, (2014).

12. Seiter, J., Amft, O., Rossi, M., Tröster, G.: Discovery of activity composites using topic models: An analysis of unsupervised methods. Pervasive Mob. Comput. 15, 215–227 (2014).

13. Aztiria, A., Augusto, J.C., Basagoiti, R., Izaguirre, A.: Accurate Temporal Relationships in Sequences of User Behaviours in Intelligent Environments. In: Augusto, J.C., Corchado, J.M., Novais, P., and Analide, C. (eds.) Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010). pp. 19–27. Springer Berlin Heidelberg (2010).

14. Srinivasan, V., Moghaddam, S., Mukherji, A., Rachuri, K.K., Xu, C., Tapia, E.M.: MobileMiner. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp ’14 Adjunct. pp. 389–400. ACM Press, New York, New York, USA (2014).

15. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities. Pervasive Mob. Comput. 5, 277–298 (2009).

16. Mabroukeh, N.R., Ezeife, C.I.: A taxonomy of sequential pattern mining algorithms. ACM Comput. Surv. 43, 1–41 (2010).

17. Mooney, C.H., Roddick, J.F.: Sequential Pattern Mining – Approaches and Algorithms. ACM Comput. Surv. 45, (2013).

18. Mallick, B., Garg, D., Grover, P.S.: Incremental mining of sequential patterns : Progress and challenges. Intell. Data Anal. 17, 507–530 (2013).

19. Soliman, A.F., Ebrahim, G. a., Mohammed, H.K.: SPEDS: A framework for mining sequential patterns in evolving data streams. Proc. 2011 IEEE Pacific Rim Conf. Commun. Comput. Signal Process. 464–469 (2011).

20. Moshtaghi, M., Zukerman, I., Russell, R.A.: Statistical models for unobtrusively detecting abnormal periods of inactivity in older adults. User Model. User-adapt. Interact. 25, 231–265 (2015).

21. Pei, J., Han, J., Wang, W.: Constraint-based sequential pattern mining: The pattern-growth methods. J. Intell. Inf. Syst. 28, 133–160 (2007).

22. Saleh, B., Masseglia, F.: Discovering frequent behaviors: time is an essential element of the context. Knowl. Inf. Syst. 28, 311–331 (2010).