



HAL
open science

Representing and Learning Human Behavior Patterns with Contextual Variability

Paula Lago, Claudia Roncancio, Claudia Jiménez-Guarín, Cyril Labbé

► **To cite this version:**

Paula Lago, Claudia Roncancio, Claudia Jiménez-Guarín, Cyril Labbé. Representing and Learning Human Behavior Patterns with Contextual Variability. International Conference on Database and Expert Systems Applications. DEXA 2017, Aug 2017, Lyon, France. pp.305-313. hal-01659750

HAL Id: hal-01659750

<https://hal.science/hal-01659750>

Submitted on 8 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Representing and Learning Human Behavior Patterns with Contextual Variability

Paula Lago¹, Claudia Roncancio², Claudia Jiménez-Guarín¹, and Cyril Labbé²

¹ Universidad de los Andes {pa.lago52, cjimenez}@uniandes.edu.co

² Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
{claudia.roncancio, cyril.labbe}@univ-grenoble-alpes.fr

Abstract. For Smart Environments used for elder care, learning the inhabitant’s behavior patterns is fundamental to detect changes since these can signal health deterioration. A precise model needs to consider variations implied by the fact that human behavior has a stochastic nature and is affected by context conditions. In this paper, we model behavior patterns as usual activity start times. We introduce a Frequent Pattern Mining algorithm to estimate probable start times and their variations due to context conditions using only one single scan of the activity data stream. Experimentation using the Aruba CASAS and the ContextAct@A4H datasets and comparison with a Gaussian Mixture Model show our proposition provides adequate results for smart home environments domains with a lower computational time complexity. This allows the evaluation of behavior variations at different context dimensions and varied granularity levels for each of them.

1 Introduction

Smart Homes used for elderly home care, can increase independent living time and reduce long-term care costs. By monitoring Activities of Daily Living (ADL), behavior patterns, their changes and their evolution, it is possible to infer a person’s health status and her ability to live independently [14].

One element to consider about behavior patterns is the probable start time of each monitored activity. It is possible to model start times since we tend to do the same activities at *approximately* the same time [14]. Changes in the start time of an activity can signal health disturbances. Nonetheless, the time when an activity is done may vary depending on context conditions. For example, one may wake up later on weekends or take a bath earlier when it is sunny outside to go out. Learning and modeling these variations avoids false alarms and improves routine understanding for both the elder and her caregivers.

If context conditions are not considered *during* learning, some patterns may go undetected [12]. But the large number of context dimensions and attribute values increases the complexity of the analysis specially since not all activities are affected by the same dimensions nor by the same scale of them (hour, day, month. . .). While the probable start time of waking up may vary on weekends, that of cooking may vary only on Fridays and that of going out may vary when it

rains. Although we can model start times using normal distributions, analyzing their possible variations at different scales is difficult.

Current approaches either don't consider possible variations in the start time of each activity, or consider the same attributes and the same scale for every activity (Section 2). In this paper, we propose the TIME algorithm (Section 3) to automatically learn probable start times of daily living activities and variations due to context conditions based on frequent pattern mining. We compare TIME's start time intervals to the intervals found by fitting a Gaussian Mixture Model using two real-life datasets (ContextAct@A4H [7] and Aruba CASAS [2]). Our method, though less accurate than fitting a normal distribution, has a lower cost and is processed in a stream setting. TIME evaluates raw data simultaneously with multiple context dimensions and scales to detect variations, without storing raw observation data, to analyze patterns. We can analyze changes and evolution in real-time, protecting user privacy (Section 4).

2 Behavior Pattern Learning in Smart Homes

Analyzing the temporal dimension of behavior can improve the accuracy of anomaly detection [9, 6]. This includes both relative ordering of activities and absolute temporal characteristics such as activity durations, start times and the temporal gaps between activities. The main approaches for modeling behavior, knowledge-driven and data-driven [1, 4], emphasize the relative sequential structure while absolute temporal characteristics are overlooked.

We model behavior start times since we tend to do the same activities at approximately the same time. Proposed approaches regard time as either a continuous or a discrete dimension. When considered as a continuous dimension, start time is mostly modeled using a Gaussian Distribution (GD) [5]. Most accurately, with a linear combination of GD (a Gaussian Mixture Model, GMM) multiple occurrences of the same activity during a day are modeled [14]. When seen as a discrete dimension, a day is segmented into chunks and the behavior occurring each interval is modeled [10] [11]. While the Gaussian Distribution method models the tightness or flexibility of a routine with its variance, discretizing the temporal space is faster and allows easier computations. Dawadi et al. [3] use m equal-sized windows to find activity frequency on each but model each activity distribution over the entire day-long period, thus finding a trade-off between both representations.

The usual start time of a behavior is modeled as the interval(s) with the highest observation frequencies or as $[\mu - \sigma_i, \mu + \sigma]$. But these times vary according to context conditions without them being anomalies [8]. To model these variations some authors create a different model for each possible situation [9, 10]. This approach is not scalable with the number of dimensions defining a situation. Other authors include context as an activity feature, thus every variation is a different concept [6]. This hinders routine understanding since the semantic relationship between variations of the same activity is lost. Finally, other approaches model

each context dimension separately [5]. While this method is scalable, it does not find frequent behaviors on rare contexts [12].

In sum, finding the probable start time(s) of an activity while considering context dimensions and different granularities (scales) for each of them with a scalable algorithm is still an open issue. We propose a frequent pattern mining method to find a suitable *trade-off between continuous and discrete representations that is computationally efficient* and precise enough to model behavior.

3 Discovering Most Probable Start Time Intervals and Probable Context Variations

In this section we formalize the problem of mining most probable start time(s) for an activity and its possible variations in light of context conditions as a frequent pattern mining problem (Section 3.1). By introducing the notion of a maximum start time interval and the expected number of observations in such interval (Section 3.2), our approach to detect start times does not previously need neither the number of possible occurrences in a day of each activity nor the context dimension granularity. TIME (Section 3.3), allows not only the analysis of different context dimensions and scales but also the analysis of pattern evolution in real-time and can be run on small computers.

3.1 Mining Activity Start Time Intervals Problem

Let $\mathcal{T} = (T, \leq)$ be a time domain. A time instant $t_x \in T$ is called a *timestamp*. Since we are interested in finding *periodic* patterns, we define a periodic granularity in which each granule groups W time instants of T . There exists a mapping $f_W(t) = t \bmod W$ for each timestamp $t \in \mathcal{T}$ allowing to represent the timestamp in a *circular time domain*. We denote this circular domain as \mathcal{T}' .

Example 1. Let \mathcal{T} be a time domain whose time instants are represented as a Unix epoch. Let $W = 86400\text{seconds}(24\text{hours})$. Let, $t_1 = 1488656696$ (2017-03-04 19:44:56) and $t_2 = 1489520696$ (2017-03-14 19:44:56), then $f_W(t_1) = 71096$ (19:44:56) and $f_W(t_2) = 71096$ (19:44:56). Both t_1 and t_2 represent the same point within the defined period.

Let $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ be a set of activity labels representing the activities to be monitored. For example, $A = \{\text{SLEEP, COOK, GO_OUT}\}$

An activity observation is a tuple $o_i = (e_i, t_j)$ with $e_i \in \mathcal{A}, t_j \in \mathcal{T}$. e_i represents the label of the observed activity and t_j represents its start time for the current observation. A contextualized activity observation is a tuple $o_i = (e_i, t_j, [c_1, c_2, \dots, c_k])$, where c_1, c_2, \dots, c_k define context attribute values of the observation such as day of the week, weather description, temperature, noise level, etc. Context attributes are a vector in which each item takes value in a specific context domain. A stream of activity observations is an unbounded sequence $\mathcal{S} = \langle o_1, o_2, o_3, \dots \rangle$, where o_i is a contextualized activity observation.

Finding the probable start time of each monitored activity a_i is equivalent to finding all the time intervals $[s, e]$, where $s, e \in \mathcal{T}'$, at which the probability that an observation of the activity belongs to the interval is greater than a user chosen parameter φ . Given a sequence \mathcal{S} , the problem can be expressed as follows:

$$\forall a_i \in \mathcal{A} \text{ find the time intervals } [s, e] \text{ where } s, e \in \mathcal{T}' \text{ such that:} \quad (1)$$

$$P(\exists o_i = (a_i, t_j) \in \mathcal{S} \mid f_W(t_j) \in [s, e]) > \varphi$$

The time interval $[s, e]$ can be interpreted as: "the time interval when there is a high probability of starting activity a_i " or as "the time interval around which activity a_i usually starts". Notice that for each activity a_i , there may be 0 or more probable start time intervals, each of which may have a different size $e - s$.

Similarly, we find variations due to a given context value c_k as the time interval(s) with a probability greater than the given parameter (see eq. 2). We assume all c_k are independent from each other, so they are considered separately.

$$\frac{P(\exists o_i = (a_i, t_j, c_k) \in \mathcal{S} \mid f_W(t_j) \in [s, e])}{P(\exists o_i = (a_i, t_l, c_k) \in \mathcal{S})} > \varphi \quad (2)$$

3.2 Maximum Start Time Interval Size

As said before, we define the total size of the timespace \mathcal{T}' by W , expressed in a corresponding temporal unit. For example, if the space is a day, W has 24 hours or 1440 minutes. This is the period of analysis. To mine probable start time intervals, we divide this space into smaller, w -sized chunks called *slots*.

Let $ntotal$ be the number of observations of an activity seen thus far and φ be the minimum probability for a time interval to be considered frequent (see eq. 1). Following (see eq. 2), and assuming events are evenly distributed, the number y of expected observations in a frequent time interval is given by: $y = ntotal * \varphi$. The maximum size T of a frequent interval is given by the minimum number of slots that would be needed to have a frequent interval under a uniform distribution. Thus T is defined by: $T = \frac{W}{w} * \varphi$. From y and T , the number n of expected observations in a slot belonging to a frequent interval is given by $n = y * \frac{1}{T}$ (3)

A probable start time interval is thus, a set of contiguous slots that have each more than n observations.

3.3 TIME: An algorithm for mining start time intervals

The algorithm relies on counting the total number of activity observations and the number of observations per activity per slot. Since there are a total of $\frac{W}{w}$ slots in \mathcal{T}' , a matrix of counters, $M_{|\mathcal{A}| \times \frac{W}{w} + 1}$ is used for this. In the matrix, $m_{i,0}$ counts the frequency of activity a_i in the stream and $m_{i,j}$ for each $j > 0$ counts the frequency of activity a_i in slot s_j .

TIME runs in two phases. The first (alg 1), runs when a new activity observation is made. It updates both the total count of activity observations and that

of the corresponding slot and activity (alg 1 lines 3 to 5). The second (alg 2), runs at the end of each period. It finds the frequent intervals for each activity by first finding the slots with counts greater than n (Eq. 3). These are the candidate slots. All contiguous candidate slots are merged into a single interval. If its total count is greater than the count of the activity times φ , then the interval is considered a probable start time for the activity. To find contextual variability of the most probable start times, we use a tree structure representing different scales of each context dimension. Each node has a matrix keeping counts for observations at the specific context value represented by the node. The root of the tree keeps the total number of observations.

Algorithm 1 TIME(w, \mathcal{S})

```

    total := 0, counts[0...|A| - 1][0... $\frac{W}{w}$ ]
2: for  $o_i \in \mathcal{S}$  do
    total := total + 1
4:   count[oi.a][0] := count[oi.a][0] + 1
    count[oi.a][oi.t] := count[oi.a][oi.t] + 1
6:   if Reached end of period then
    call merge_intervals(counts, total)
8:   end if
end for

```

Algorithm
2

merge_intervals($c[\square\square], total$)

```

 $\mathbb{I} \leftarrow \emptyset$ , min := total *  $\frac{w}{W}$ 
2: for  $i \in [0 \dots |\mathcal{A} - 1]$  do
    s, e := -1
4:   for  $j \in [1 \dots \frac{W}{w}]$  do
    n := c[i][j] / c[i][0]
6:     if n > min then
    if s == -1 then
8:       s, e := j
    else
10:      e := j
    end if
12:    else if s! = -1 then
     $\mathbb{I} \leftarrow \mathbb{I} \cup (i, s, e)$ , s, e := -1

```

4 Empirical Evaluation

In this section, we prove three properties of TIME by empirical evaluation: (1) It can find multiple start times in a day for an activity with a precision similar to that of a GMM (Section 4.1), (2) It detects contextual variability at different scales for such intervals (Section 4.2) and, (3) Its patterns may be used to increase routine awareness from sensor data (Section 4.3). We used the Aruba dataset (222 days) from CASAS project [2] and the ContextAct@A4H dataset (28 days) from the Amiqal4Home Lab [7]. From the former dataset, we focus on the SLEEP(401 observations), EATING(259 obs.), MEAL PREPARATION(1605 obs.) and WASH DISHES(67 obs.) activities. From the latter, we focus on the SLEEP(26 obs.), SHOWER(24 obs.), COOK(77 obs.) and WASH DISHES(52 obs.) activities. Both datasets contain real-life sensor data of activities of daily living annotated by the inhabitant.

4.1 Comparing intervals to GMM

This experiment assessed the precision of the intervals found by TIME by comparing them to those found by fitting a GMM on the Aruba dataset. We found the optimal number of components for the GMM using the BIC criterion [13] and fit

the model using the scikit-learn library ³. Taking $[\mu - \sigma, \mu + \sigma]$ of each component as start interval, we compared them to those found by TIME graphically (Fig. 1). Notice how both methods find different number of probable intervals each with a different size for each activity. Eating shows the highest difference because it is too irregular around midday so TIME cannot find any usual time but GMM finds a low probability interval.

As a quantitative measure, we calculated the Jaccard similarity of the intervals whose size is measured in minutes (Table 1). Eating has the lowest similarity due to the interval between 9:30 and 17:00, not recognized by TIME but found by GMM. Even though, the average similarity for every other activity is suitable.

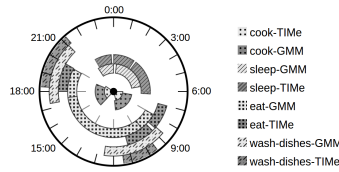


Fig. 1: GMM v.s. TIME start time intervals

Activity	Average similarity
Sleep	76,7 %
Eating	38,8 %
Cook	87,1 %
Wash Dishes	59,9 %

Table 1: Jaccard similarity of GMM and TIME intervals

4.2 Finding pattern variability

This experiment evaluated how TIME can detect variability with respect to different context conditions. For the Aruba dataset we consider the day of the week as a context dimension and evaluated differences in week and weekend patterns (Fig. 2) and in day to day variability (Figures 3 and 4). These results show how an activity can start at different times each day.

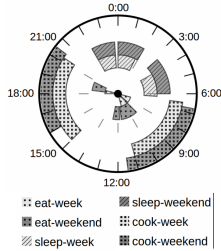


Fig. 2: Week v.s. Weekend time variations

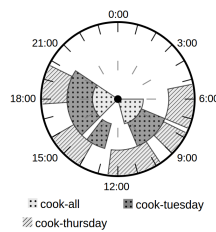


Fig. 3: Cooking time variations

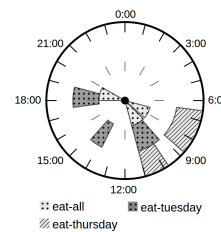


Fig. 4: Eating time variations

To explore variability with respect to other context dimensions, we used ContextAct@Home considering day of the week and weather conditions (Fig. ??). Wake up corresponds to the end time of sleep. The variations found help to better reason about routines. For example, if only patterns on the whole dataset were

³ <http://scikit-learn.org/>

mined, taking a bath after midday (Fig. 5) or going to sleep later on Fridays (Fig. 6) or waking up later when it rains (Fig. 7) would be considered an anomaly but those changes are common given their context.

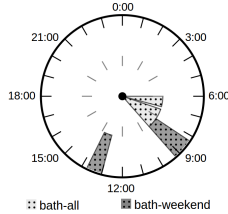


Fig. 5: Bath variations

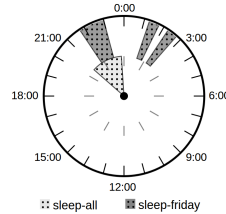


Fig. 6: Sleep variations

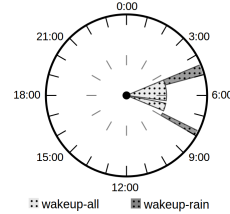


Fig. 7: Wake variations

4.3 Understanding sensor data using patterns

To show how sensor events patterns increase understanding, we compared start time intervals of the bed pressure, stove state, shower faucet state and dishwasher faucet state sensor events to the intervals of sleeping, cooking, showering and washing dishes activity annotations respectively (Fig. 8). We chose these activities because the sensors highly correlate to doing the activity. Intervals are very similar for all activities (Table 2).

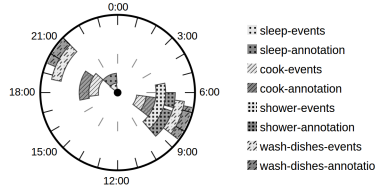


Fig. 8: Activity v.s. Sensor events intervals (A4H dataset)

Showering and washing dishes have lower similarities because these activities started a while after stating it and because the faucet is used for other activities (like washing vegetables). Still, patterns resemble the inhabitant’s routine greatly.

<i>Activity</i>	<i>Jaccard similarity</i>
Sleep	84,6 %
Shower	81,2 %
Cook	89,3 %
Wash Dishes	77,1 %

Table 2: Average similarity for activity and sensor events intervals

5 Discussion, Conclusion and Perspectives

In this paper, we presented TIME, a frequent pattern mining algorithm on stream data to learn start times and their variations without manual data slicing. With GMM as a baseline, our experiments show that probable start time intervals mined with TIME are suitable (Section 4.1) but at a significant lower cost since TIME uses a single data scan to analyze incoming data in a streaming mode.

TIME can analyze variability due to multiple context dimensions each at multiple scales without manual data slicing. Our results show that there are in fact variations in patterns when analyzed in the light of different context

conditions, but that not all patterns change with respect to the same context condition nor scale (Section 4.2). Given this, it is specially important to analyze each pattern separately and that each dimension is evaluated independently from others.

Being able to find pattern variations can help to better understand behavior and better analyze events so that false alarms are reduced. We have shown that patterns from sensor events are highly similar to those of annotated activities (Section 4.3).

TIME maintains the relationship between an activity and its variations. This not only allows a separation of concerns but also creates a richer semantic model of behavior. As such, activity recognition methods can focus on global characteristics and personal characteristics such as start time are dealt with by a behavior analysis method.

As future work we will apply TIME to analyze pattern evolution and also study how to learn other properties of behavior (expected frequency, frequent sequences). These aspects complete knowledge about personal behavior patterns.

References

1. Chen, L., Hoey, J., Nugent, C.D., Cook, D.J., Yu, Z.: Sensor-based activity recognition. *IEEE T Syst Man CY C* 42(6), 790–808 (Nov 2012)
2. Cook, D., Crandall, A., Thomas, B., Krishnan, N.: CASAS : A Smart Home in a Box. *IEEE Computer* 46(7), 62–69 (2013)
3. Dawadi, P.N., Cook, D.J., Schmitter-Edgecombe, M.: Modeling patterns of activities using activity curves. *Pervasive and Mobile Computing* 28, 51–68 (2015)
4. Díaz-Rodríguez, N., Cuéllar, M., Lilius, J., et.al.: A survey on ontologies for human behavior recognition. *ACM Comput. Surv.* 46(4), 43:1–43:33 (Mar 2014)
5. Forkan, A., et.al.: A context-aware approach for long-term behavioural change detection and abnormality prediction in AAL. *Patt. Recognit* 48(3), 628–641 (2015)
6. Jakkula, V.R., Crandall, A.S., Cook, D.J.: Enhancing Anomaly Detection Using Temporal Pattern Discovery. In: *Adv. Intel. Env.*, pp. 175–194. Springer US (2009)
7. Lago, P., et.al.: The ContextAct@A4H real-life dataset of daily-living activities. Activity recognition using model checking. In: *LNAI*. Springer (jun 2017)
8. Lago, P., Jiménez-Guarín, C., Roncancio, C.: A case study on the analysis of behavior patterns and pattern changes in smart environments. *LNCS* 8868, 296–303 (2014)
9. Monekosso, D.N., Remagnino, P.: Behavior Analysis for Assisted Living. *IEEE Transactions on Automation Science and Engineering* 7(4), 879–886 (oct 2010)
10. Moshtaghi, M., Zukerman, I., Russell, R.A.: Statistical models for unobtrusively detecting abnormal periods of inactivity in older adults. *User Model User-Adap* 25(3), 231–265 (Aug 2015)
11. Nait Aicha, A., Englebienne, G., Kröse, B.: Unsupervised visit detection in smart homes. *Pervasive and Mobile Computing* 34, 157–167 (2016)
12. Rabatel, J., Bringay, S., Poncelet, P.: Contextual sequential pattern mining. In: *2010 IEEE ICDM Workshops*. pp. 981–988 (Dec 2010)
13. Schwarz, G.: Estimating the Dimension of a Model. *Ann. Stat* 6(2), 461–464 (1978)
14. Soulas, J., Lenca, P., Thépaut, A.: Unsupervised discovery of activities of daily living characterized by their periodicity and variability. *Eng. Appl. Artif Intel* 45, 90–102 (2015)