



An Online Pattern Based Activity Discovery: In Context of Geriatric Care

Moumita Ghosh^(✉) , Sayan Chatterjee , Shubham Basak ,
and Sankhayan Choudhury 

Department of Computer Science and Engineering, University of Calcutta,
Kolkata, India
ghoshmoumita06@gmail.com, scsayan96@gmail.com, shubhamsk@gmail.com,
sankhayan@gmail.com

Abstract. The behavioral analysis of an elderly person living independently is one of the major components of geriatric care. The day-long activity monitoring is a pre-requisite of the said analysis. Activity monitoring could be done remotely through the analysis of the sensory data where the sensors are placed in strategic locations within the residence. Most of the existing works use supervised learning. But it becomes infeasible to prepare the training dataset through repeated execution of a set of activities for a geriatric person. Moreover, the geriatric people are annoyed to use wearable sensors. Thus it becomes a challenge to discover the activities based on only ambient sensors using unsupervised learning. Pattern-based activity discovery is a well-known technique in this domain. Most of the existing pattern based methods are offline as the entire data set needs to be mined to find out the existing patterns. Each identified pattern could be an activity. There are a few online alternatives but those are highly dependent on prior domain knowledge. In this paper, the intention is to offer an online pattern based activity discovery that performs satisfactorily without any prior domain knowledge. The exhaustive experiment has been done on benchmark data sets ARUBA, KYOTO, TULUM and the performance metrics ensure the strength of the proposed technique.

Keywords: Geriatric care · Unsupervised learning · Activity detection

1 Introduction

Nowadays, population aging and care for population aging are common phenomena [1]. It seeks out attention from the people of various domains such as media-persons, policymakers, physicians, and adult-care social workers. The major concern is their well-being that can be assessed through their daily activities. There are several types of basic activities such as eating, bathing, getting dressed, etc. and these are called “Activities of Daily Living” (ADL) [13, 14]. Moreover, based on the existing socio-economic status it can be apprehended

that within a couple of decades the elderly people need to reside independently. Certainly, there is a strong association between “Independent Living” and being able to perform daily activities. The smooth execution of ADLs play a crucial role to determine whether a person needs assistance; especially the degree of assistance from caregivers to survive alone in his/her residence.

There are many ways to assess the performance of daily activities. A person is monitored for a long time for the necessary behavioral analysis. The first step towards measuring performance is to get information about the activities. There are primarily two approaches to track the activities. These are either vision based or sensor based [2]. In vision based, a video camera is used to track the activities. But, the prominent form of activity tracking is sensor based as a camera-based tracking system violates some sort of privacy of the users. The sensor-based tracking system uses two types of sensors. Ambient sensors are placed in some strategic locations to sense the human movement. As opposed to this, wearable sensors are tied in some specific body position to track the activities. It may create some extent of annoyance to the geriatric people.

The activity recognition addresses the challenge of identifying the activity through the necessary analysis of sensory data. The said technique mostly relies on supervised learning [3, 7]. The common practice of supervised learning is to perform some predefined activities in a scripted setting for the generation of training data. The generation of training data is a difficult job, especially in the target domain. Most of the time the geriatric persons are not willing to perform the same task repeatedly. As a result, using a scripted setting, the generation of training data is almost infeasible. The only alternative could be recognizing the activities in an unsupervised way and this is termed as “Activity Discovery” [11] in literature.

The approach of activity discovery is broadly categorized in two ways. Most of the activity discovery method relies on finding motifs/patterns to identify activities. It identifies the frequent and repeatable sequences over the entire data set. These repetitive sequences are called motifs [12]. This approach can detect only those activities for which motifs are found [5, 10–12]. Thus, it fails to detect activities with lower frequency compared to the activities with higher frequencies. Another widely used approach is feature based activity discovery. In this approach, the similar data points are included in a cluster. The widely used approaches are k -Means, DBScan, Agglomerative, etc. [9]. The similarity among data points is measured using the different features [8] and similar data points form a cluster. Some of the features used to measure the similarity are mean, median, standard deviation, energy, integral, skewness, kurtosis, and root mean square, etc. [8].

In reality, geriatric people are annoyed to use wearable sensors. In the case of wearable sensors, finding the similarity among sensor events is not a challenging issue as distinct values trigger for different activities. But, in an ambient sensor especially in the case of binary motion sensors, finding the similarity among motion sensors is a challenging task. Location and time act as major features in most of the existing works. The domain knowledge is to be incorporated

for measuring the correlation among motion sensors. It puts an extra burden to acquire the domain knowledge [15]. From the above discussion, it can be apprehended that to work with ambient motion sensors, pattern-based activity discovery could be a better solution for activity monitoring in a geriatric care environment.

In general, pattern means frequent occurrences of event sequences. The patterns that occur more than given predefined support are treated as activities [12]. Most of the pattern based approaches are offline [5] as the entire dataset needs to be present beforehand for necessary mining. In contrast, the existing online approaches depend on prior knowledge [10]. The above discussion demands the need for proposing an online discovery technique that uses only ambient sensors for a geriatric care application without using domain knowledge.

The paper is structured as follows. Section 2 describes the problem with an illustrative example. In Sect. 3, the solution of the problem, i.e. the proposed method of activity discovery is discussed in detail. Section 4 describes the experiment for the necessary validation. Section 5 concludes the discussion highlighting the specific contributions of the work.

2 Scope of the Problem

It is assumed that there is an elderly person in the house alone. She is capable of doing all her daily activities. Our concern is to identify some basic activities like Sleeping, Bed-to-toilet, Eating, Meal preparation, etc. The frequency of the activities in respective days may be different. The person is monitored through the generated spontaneous signals from the ambient (motion) sensors as placed in the strategic locations within the house. Also, we assume that the activities are performed sequentially. The challenge is to discover the activities online. Here the term “online” implies that within a finite amount of time (not real-time), the proposed solution will be able to discover the activities.

The ARUBA [4] baseline data, as created by Washington State University, is considered here for experimentation of the proposed discovery algorithm. Let us, consider the floor plan as given in ARUBA [4] and is depicted in Fig. 1. It can be seen from the diagram that several sensors are placed in some strategic locations. The sensors are only triggered whenever a person is within the range of a sensor as well as he is moving position. The scope includes an online activity discovery technique proposal and subsequent experimentation on the benchmark data set for ensuring the effectiveness of the proposed approach. However, we have used the raw dataset assuming that there is no noise. Moreover, the activity recognition, based on the output of the proposed discovery algorithm is also within the scope. The experimentation on existing benchmark data sets is also one of the targets to assess the effectiveness of the algorithm irrespective of the nature of a specific data set. As a result, the datasets TULUM and KYOTO [6] are also identified beside ARUBA [4] for necessary experimentation.

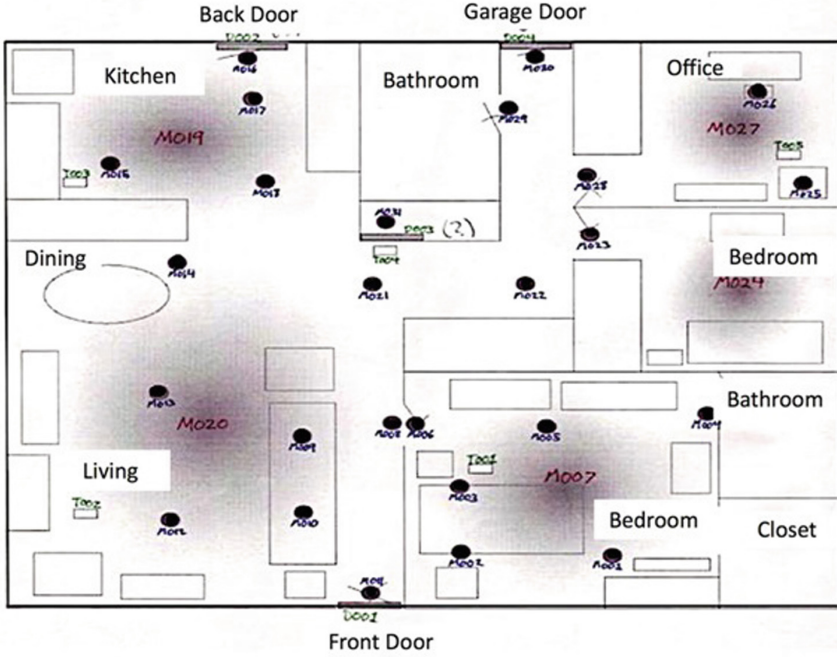


Fig. 1. Floor plan as mentioned in ARUBA.

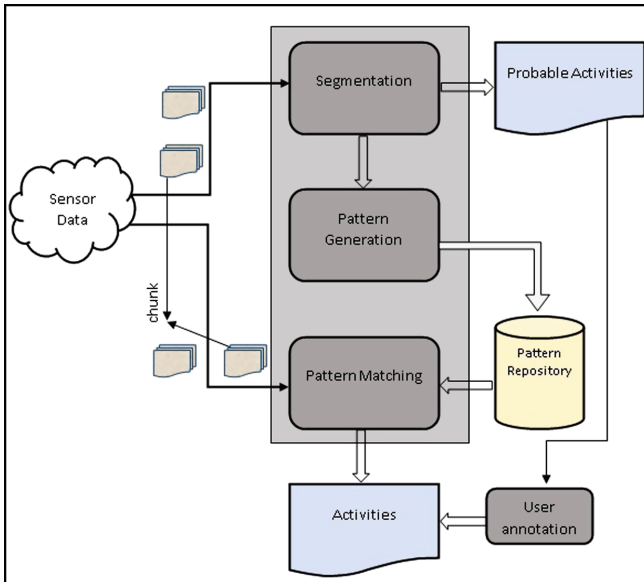


Fig. 2. Block diagram of activity discovery.

Table 1. Raw data as triggered in ARUBA data set

Date	Time	SensorId	Value	Annotation
04-11-2010	08:11:10 AM	M018	ON	Meal_Preparation begin
04-11-2010	08:11:15 AM	M019	ON	
04-11-2010	08:11:16 AM	M015	ON	
04-11-2010	08:11:17 AM	M018	OFF	
04-11-2010	08:11:19 AM	M019	OFF	
04-11-2010	08:11:21 AM	M015	OFF	
04-11-2010	08:11:23 AM	M019	ON	
04-11-2010	08:11:37 AM	M019	OFF	
04-11-2010	08:11:37 AM	M019	ON	
04-11-2010	08:11:39 AM	M019	OFF	

3 Proposed Approach

In general, activity discovery is conceptualized as in the block diagram depicted in Fig. 2. The sensors trigger as and when an activity takes place. The continuous execution of activities generates a stream of sensor data. Our approach is a two-fold approach. In the first form, it aims to segment the stream of sensor data activity wise and the subsequent pattern generation is carried out to detect the further occurrences of an activity. It is assumed that after activity-based segmentation, the discovered segment is labeled manually. The detailed discussion on each phase in the discovery process is given below.

3.1 Preliminaries: Proposed Solution

The following preliminaries are illustrated for substantiating the proposed discovery mechanism.

Data Representation: In this present article, we use the data set collected from the sensor based smart home like ARUBA [4], TULUM and KYOTO [6]. The data set consists of several tuples or rows. Each row represents one single event. The field of the tuples is Date, Time, sensor identifier and sensor value. A snapshot of related events is shown below in Table 1. This is the typical example of an activity called Meal Preparation.

Artifacts: The artifacts are described in the following section.

Event: Each tuple of a data set is called an event. From the above table, each of the rows is treated as an event. For example, the first row of the above Table 1 signifies that motion sensor M019 triggers on dated 2010-11-04 at 08:12:23.

Sequence: Several consecutive events are called sequence.

Chunk: A sequence of sensor data is called a chunk. This may or may not pertain to a particular activity. A chunk can be fixed length or variable length. The length of the chunk depends on the chunk selection strategy. If the chunk is selected event wise then a chunk must include a fixed number of events. The number of entries in all the subsequent chunks is equal. On the other hand, in the time-based chunk selection strategy, the number of entries differs from one chunk to another. Here, time is fixed. In the above table, a Chunk of length 10 (Chunk_SIZE) has been shown.

Buffer: One or more chunks containing sensor events pertaining to only a particular activity is defined as a buffer. It is supposed to be a chunk of a sensor event sequence that occurs from the starting of an instance of an activity to its end.

Segment: A buffer is cleared from memory after waiting for a predefined time, with the assumption that the activity instance has been completed. This buffer, after being mature, is termed as a segment.

Unique_sensor: The set of distinct sensors that appear in chunk or Buffer.

Dominant_sensor: The dominant sensors are those whose frequency of occurrence in a chunk or buffer is higher than a predefined threshold T.

Pattern: The descending order of top n most trigger events of a chunk. So, from the above example, pattern P is defined as $\langle M019, M015 \rangle$. In the next subsection, the segmentation procedure is discussed in depth.

3.2 Segmentation

The activity discovery is executed by several distinct modules as depicted in Fig. 2. The stream of sensor data is the input to the activity discovery block. The details of each module are given below.

Before processing, the stream data is divided into chunks(collection of events). The chunk is either taken event wise or time wise. A fixed number of events are recorded in event-based strategy whereas, in time-based segmentation, the number of events in a segment may vary. In the target domain, the stream data may generate in discretely as the concerned sensors are only ambients. As a result, our choice could be time-based. In an online environment, event-based segmentation, one needs to wait a long time to fill the chunks in terms of event number such as twenty events. The most challenging part is to choose the interval of time as sometimes no data event may come or sometimes data appears at a higher rate. But, the problem persists irrespective of the fact of using time-based chunk selection or event-based technique as we deal with stream data. One solution is to select the chunk activity wise and that could be the fittest solution for our target domain. We have used the time-based segmentation for creating the initial chunks and the generated chunks are merged to form an activity through the proposed activity based segmentation. For the sake of completion in the experimentation phase, the time based as well as event-based segmentation are used for chunk creation to asses the performance variation between the said techniques.

The activity-based segmentation, i.e. the merging of the existing chunks, as generated through time-based/event-based technique, is done without any prior

Algorithm 1: Online Pattern based Segmentation Technique (*OPST*).

Input : *Wait.time* , Stream of sensor events (*S*), *Chunk_Size* , *Buffer.list*
Output: Pool of Activity Patterns

```

1 Chunk_size = c.size
2 Buffer_list = b.list
3 start = 0
4 while start in range of data stream do
5   Cur.Chunk = content [start:start+Chunk_size]
6   if Buffer_list == {} then
7     | Cur_Buffer = Cur_Chunk
8   end
9   else
10    | set_diff = Unique_Sensor.Cur_Chunk \ Unique_Sensor.Cur_Buffer
11    | // perform the set difference between the unique sensors of Current
12    | Chunk and Current Buffer and UniqueSensors is a function that
13    | calculates unique sensors of Current Chunk.
14    | if set_diff == {} then
15    | | // all the unique elements of current chunk is common to current
16    | | buffer.
17    | | Cur_Buffer = Cur_Buffer + Cur_Chunk // Immediately current
18    | | chunk is appended to current buffer.
19    | end
20    | if set_diff == {all unique elements of Cur_Chunk} then
21    | | // When all the unique elements of current chunk are uncommon to
22    | | current buffer
23    | | Cur_Buffer = Cur_Buffer + 1 // the current chunk is not
24    | | appended to current buffer; it will search for another new buffer
25    | | to append the current chunk
26    | | Cur_Buffer = Cur_Chunk
27    | end
28    | if set_diff == {Cur_Chunk \ (Cur_Chunk ∩ Cur_Buffer)} then
29    | | // When, some common elements are present in the current chunk, we
30    | | have to wait for next chunk. temp_index = start + Chunk_size + 1
31    | | temp_Chunk = content [temp_index:temp_index+Chunk_size]
32    | | // Bringing the next Chunk in the temporary Chunk.
33    | | set_differ = Unique_Sensor.Cur_Chunk \ Unique_Sensor.temp_Chunk
34    | | if set_differ == {} then
35    | | | Cur_Buffer = Cur_Chunk
36    | | end
37    | | if set_differ == {Cur_Chunk \ (Cur_Chunk ∩ temp_Chunk)} then
38    | | | output_common = Cur_Chunk ∩ temp_Chunk
39    | | | Cur_Buffer = all output common from Cur_Chunk
40    | | end
41    | end
42  end
43  foreach buf in Buffer_list do
44    | Eva_Buffer = Buffer_list[buf]
45    | if time.Cur_Buffer[last_ele] + Wait.time ≤ time.Cur_Chunk[first_ele]
46    | then
47    | | Output Buffer as probable activity segment
48    | end
49  end
50 end

```

Algorithm 2: Pattern Generation.

```

Input : Pool of Committed Activities(Com_Act)
Output: Pool of Activity Patterns
1 foreach Activity in Pool of Activity Patterns do
2   Sensor_List = getFrequency(Com_Act) // getFrequency is a function
   that calculates frequency of each distinct sensor in a segment of
   committed activity.
3   Dom_Sensor_List = getDominant(Sensor_List) // It prunes top n
   number of sensors from the sensor list.
4   Pattern = nonDecreasing(Dom_Sensor_List) // Arrange the dominating
   sensors in non decreasing order.
5   Pattern_repository = Pattern // Store each pattern in pattern
   repository.
6 end

```

domain knowledge. The rationale behind the concept is that if a set of events occurs in several consecutive chunks, then it could be a part of the same activity. As a result, the chunks must be merged in a single segment. This process will be continued until a new set of sensors is generated and that creates the initiation of a new chunk i.e. a new activity segment. Every new segment is given a predefined threshold time for its maturity. The activity is clear from the memory after the threshold time, and it is given to the experts for annotation. The maturity time selection strategy is described later. The detailed of the segmentation procedure is given in Algorithm 1.

Threshold Time: The selection of threshold time is dynamic. It has been decided for event-based and time-based respectively. The threshold time for maturity in event-based is fixed and can be derived through the experiment. In the case of time-based segmentation, the difference in times between the last event of the active segment and the first event of the next chunk is treated as threshold time. It adapts the varying property of different activities in terms of time duration.

Pattern Generation: The pattern is generated from a probable activity segment. The frequency of the dominating sensors is noted. Then sensors are arranged in nondecreasing order. The non-decreasing order of the sensors is called a pattern. The patterns generation process is described in the Algorithm 2.

Pattern Matching: The evolving patterns are kept in a repository. Later, when another chunk will arrive, at first, the chunk is compared with predefined patterns to infer the same pattern can be derived from the chunk or not. If the pattern is found from the previously discovered set of patterns, we simply mark the chunk as an instance of a predefined activity and wait for its maturity. The basic principle of matching is depicted in the Algorithm 3.

Algorithm 3: Pattern Matching.

```

Input : Cur_Chunk, Pool of Activity Patterns
Output: Committed Activity Segments
1 foreach Activity in Pool of Activity Patterns do
2   | Dom_Cur_Chunk = getDominant(Cur_Chunk) // It prunes top n
   | number of sensors of Current Chunk.
3   | Pattern_Chunk = nonDecreasing(Dom_Cur_Chunk) // Arrage the
   | dominating sensors in non decreasing order.
4   | Result = doMatch(Pattern_Chunk, Cur_Act_Pat) // Matching in
   | between current chunk pattern and all available patterns.
5   | if Result is True then
6     | | Activity_segment = Cur_Chunk // Assigned current chunk as
   | | activity segment.
7   | end
8   | else
9     | | OPST(Cur_Chunk) // Create new activity segment.
10  | end
11 end

```

4 Experimental Findings

We have identified three benchmark data sets ARUBA [4], KYOTO, TULUM [6] for the necessary experimentation. The objective is to measure the performance of the proposed discovery technique in terms of activity detection ratio. Aruba dataset [4] contains sensor data that was collected from the home of a volunteer adult. Aruba dataset collected from a house that consists of a bedroom, a kitchen, a bathroom, a dining room, and an office. The home Aruba included 31 sensors to collect environmental information. All activities are collected from a single inhabitant within the period of 2010-11-04 to 2011-06-11.

Kyoto dataset represents sensor events collected in the WSU smart apartment testbed during the summer of 2009. The apartment housed two residents R1, R2 and they performed their normal daily activities. Herein, 51 motion sensors, five temperature sensors, fifteen door sensors, a burner sensor, hot and cold-water sensors, and an electric sensor were used for the necessary data recording. In our problem, we consider only motion sensors as per the problem definition. WSU Tulum Smart Apartment 2009 is a two Resident test-bed. This dataset represents sensor events collected in the WSU smart apartment test-bed from April to July of 2009. The sensors consist of 18 motion sensors (M001 through M018) and two temperature sensors (T001 and T002). Ten activities are annotated by denoting the begin and end of each activity occurrence. The data set TULUM and KYOTO consider two residents. But, in this work, our intention is to detect the activities irrespective of the association of activity to its corresponding residence.

Table 2. Detection ratio in ARUBA.

Activity name	No. of activity instances occurring			Detection ratio in	
	In ground truth data ARUBA	In event based chunk selection	In time based chunk selection	In event based chunk selection	In time based chunk selection
Sleeping	2	1	2	0.5	1.00
Bed to toilet	1	1	1	1	1.00
Eating	4	3	4	0.75	1.00
House keeping	2	8	-	4.00	-
Meal preparation	9	12	9	1.33	1.00
Relax	5	7	4	1.40	0.80
Wash dishes	2	3	2	1.50	1.00
Work	2	2	2	1.00	1.00

Table 3. Detection ratio in TULUM.

Activity name	No. of activity instances occurring			Detection ratio in	
	In ground truth data ARUBA	In event based chunk selection	In time based chunk selection	In event based chunk selection	In time based chunk selection
R1_cook_breakfast	1	1	1	1.00	1.00
R1_eat_breakfast	2	1	2	0.50	1.00
R1_cook_lunch	1	1	1	1.00	1.00
Leave_home	1	1	1	1.00	1.00
R1_snack	14	11	9	0.78	0.64
Watch_tv	16	13	10	0.81	0.63

We have executed the said discovery on these three data sets. The time-based segmentation is used for creating initial chunks. To assess the variation between time-based and event-based, the second approach is also used in the experimentation phase. We present the result as obtained using both the techniques. We have done the experimentation with different chunk sizes for the event based chunk creation. We found that the chunk sizes of 10, 5, 5 respectively to perform the test for the dataset ARUBA, TULUM and KYOTO. All the cases, we assume the fixed wait time and is set as five minutes. As an outcome of the experimentation phase, the number of activity segments is 127, 109, 68 respectively in three data sets. In time-based chunk creation, the number of activity segments is found 133, 107, 68 respectively in three data set.

Table 4. Detection ratio in KYOTO.

Activity name	No. of activity instances occurring			Detection ratio in	
	In ground truth data ARUBA	In event based chunk selection	In time based chunk selection	In event based chunk selection	In time based chunk selection
R1 sleep	1	1	1	1	1
R1 wakeup	1	1	1	1	1
R1 grooming	1	1	1	1	1
R1 shower	1	2	1	2	1
R1 work	12	14	7	1.17	0.58

Table 5. Accuracy measured in ARUBA.

Activity name	Precision		Recall		F1-Measure	
	Event based	Time based	Event based	Time based	Event based	Time based
Sleeping	0.90	0.88	0.86	0.59	0.87	0.69
Bed to toilet	0.72	0.73	0.58	0.76	0.59	0.72
Eating	0.77	0.81	0.76	0.79	0.74	0.80
House keeping	0.89	0.99	0.28	0.09	0.37	0.17
Meal preparation	0.88	0.81	0.71	0.61	0.75	0.66
Relax	0.84	0.80	0.78	0.77	0.78	0.76
Wash dishes	0.65	1.00	0.66	0.65	0.63	0.77
Work	0.79	0.82	0.76	0.77	0.74	0.77

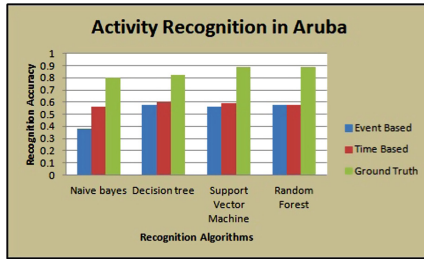
Table 6. Accuracy measured in TULUM.

Activity name	Precision		Recall		F1-Measure	
	Event based	Time based	Event based	Time based	Event based	Time based
R1_cook_breakfast	1.00	1.00	0.64	0.86	0.78	0.93
R1_eat_breakfast	0.19	0.74	0.41	0.77	0.26	0.75
R1_cook_lunch	0.69	1.00	0.73	0.86	0.71	0.93
Leave_home	0.89	0.94	0.56	0.89	0.69	0.92
R1_snack	0.75	0.85	0.30	0.51	0.36	0.58
Watch_tv	0.85	0.82	0.50	0.50	0.39	0.60

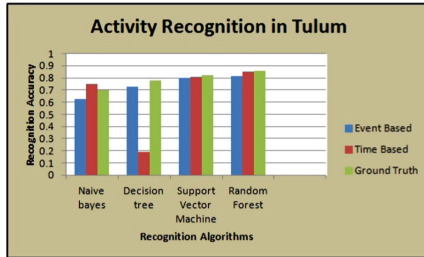
The following tables show the performance of the proposed discovery technique in terms of well-known parameters called Precision, Recall and F1 measure. are shown in Tables 5, 6 and 7. The activity detection ratio is also depicted in

Table 7. Accuracy measured in KYOTO.

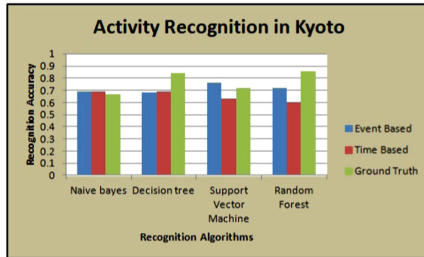
Activity name	Precision		Recall		F1-Measure	
	Event based	Time based	Event based	Time based	Event based	Time based
R1 sleep	0.96	0.99	1.00	0.82	0.98	0.90
R1 wakeup	0.06	1.00	1.00	0.67	0.11	0.80
R1 grooming	0.70	0.71	1.00	1.00	0.82	0.83
R1 shower	0.24	0.29	0.77	0.90	0.37	0.43
R1 work	0.38	0.87	0.69	0.58	0.41	0.67



(a) Activity Recognition in ARUBA.



(b) Activity Recognition in TULUM.



(c) Activity Recognition in KYOTO.

Fig. 3. Evaluation of activity recognition algorithms in different data sets.

the Tables 2, 3 and 4. The overall performance is satisfactory except few cases. It ensures the effectiveness of the proposed discovery algorithm. Moreover, activity recognition is done using five days of annotated data. We train the model using five days annotated data as per the detection is done by the proposed discovery on the five days data in the original data set. Based on the said training, recognition is performed using well-known approaches like Naive Bayes, Decision Tree, Support Vector Machine, and Conditional Random Field. The accuracies of the recognized activities are depicted in Fig. 3.

5 Conclusion

The scope of the paper is to deliver an online activity discovery in an ambient sensor based home care environment without any prior domain knowledge. The requirements are set keeping the geriatric care application in mind. The proposed solution will be suitable for use in the geriatric care domain where the online monitoring of an old aged person is highly needed. The solution does not require any training as well as any prior domain knowledge as it becomes an almost infeasible task in the above-said domain. The proposed pattern-based discovery uses time-based segmentation on the stream data for the creation of initial chunks. These created chunks are merged based on the said pattern matching and generate a segment that could be a probable activity. Thus the overall approach can be treated as an activity-based segmentation. The overall performance is satisfactory except few cases e.g. activity recognition result in ARUBA is not good enough whereas it shows much better for KYOTO and TULUM data set. The proposed online segmentation cum discovery approach is memory efficient. The algorithm works with the growing segment in memory and outputs the matured one. Moreover, further activities like annotation can be going on parallel while the discovery process is running for the other activities. Several issues remain for further investigation. The discovery needs to be modified for detecting scattered activity. The transitions between the activities have to be identified for better accuracy.

References

1. Elderly in india. http://mospi.nic.in/sites/default/files/publication_reports/ElderlyinIndia_2016.pdf
2. Benmansour, A., Bouchachia, A., Feham, M.: Multioccupant activity recognition in pervasive smart home environments. *ACM Comput. Surv. (CSUR)* **48**(3), 34 (2016)
3. Brdiczka, O., Crowley, J.L., Reignier, P.: Learning situation models in a smart home. *IEEE Trans. Syst. Man Cybern. B Cybern.* **39**(1), 56–63 (2008)
4. Cook, D.J.: Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2010**(99), 1 (2010)
5. Cook, D.J., Krishnan, N.C., Rashidi, P.: Activity discovery and activity recognition: a new partnership. *IEEE Trans. Cybern.* **43**(3), 820–828 (2013)

6. Cook, D.J., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. *Methods Inf. Med.* **48**(05), 480–485 (2009)
7. Fleury, A., Noury, N., Vacher, M.: Supervised classification of activities of daily living in health smart homes using SVM. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6099–6102. IEEE (2009)
8. Gjoreski, H., Roggen, D.: Unsupervised online activity discovery using temporal behaviour assumption. In: Proceedings of the 2017 ACM International Symposium on Wearable Computers, pp. 42–49. ACM (2017)
9. Kisilevich, S., Mansmann, F., Nanni, M., Rinzivillo, S.: Spatio-temporal clustering. In: Maimon, O., Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*, pp. 855–874. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09823-4_44
10. Rashidi, P., Cook, D.J.: Mining sensor streams for discovering human activity patterns over time. In: 2010 IEEE International Conference on Data Mining, pp. 431–440. IEEE (2010)
11. Rashidi, P., Cook, D.J., Holder, L.B., Schmitter-Edgecombe, M.: Discovering activities to recognize and track in a smart environment. *IEEE Trans. Knowl. Data Eng.* **23**(4), 527–539 (2011)
12. Saives, J., Pianon, C., Faraut, G.: Activity discovery and detection of behavioral deviations of an inhabitant from binary sensors. *IEEE Trans. Knowl. Data Eng.* **12**(4), 1211–1224 (2015)
13. Thapliyal, H., Nath, R.K., Mohanty, S.P.: Smart home environment for mild cognitive impairment population: solutions to improve care and quality of life. *IEEE Consum. Electron. Mag.* **7**(1), 68–76 (2018)
14. Urwyler, P., Stucki, R., Rampa, L., Müri, R., Mosimann, U.P., Nef, T.: Cognitive impairment categorized in community-dwelling older adults with and without dementia using in-home sensors that recognise activities of daily living. *Sci. Rep.* **7**, 42084 (2017)
15. Ye, J., Stevenson, G.: Semantics-driven multi-user concurrent activity recognition. In: Augusto, J.C., Wichert, R., Collier, R., Keyson, D., Salah, A.A., Tan, A.-H. (eds.) *AmI 2013. LNCS*, vol. 8309, pp. 204–219. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03647-2_15