A New Algorithm Based On Sequential Pattern Mining For Person Identification In Ubiquitous Environments

Belkacem Chikhaoui  
Prospectus Laboratory  
University of Sherbrooke  
Canada  
Belkacem.Chikhaoui@USherbrooke.ca

Shengrui Wang  
Prospectus Laboratory  
University of Sherbrooke  
Canada  
Shengrui.Wang@USherbrooke.ca

Hélène Pigot  
Domus Laboratory  
University of Sherbrooke  
Canada  
Helene.Pigot@USherbrooke.ca

ABSTRACT

This paper presents an approach to person identification in ubiquitous environments. Our approach uses the sequential pattern mining principle to extract frequent patterns in data collected from the different sensors disseminated in the ubiquitous environment. In contrast with existing, intrusive, person identification algorithms that have been proposed in the literature, where the data is basically composed of audiovisual or image files recorded during experiments, our approach is fully non-intrusive and is based on event sequences collected from heterogeneous sensors. Our approach is divided into three main phases: (1) frequent pattern mining, (2) assignment of weights to extracted patterns, and (3) classification. Experiments using data collected in the Domus and Testbed smart homes demonstrate that our approach accurately identifies persons and improves classification results, outperforming two of the approaches reported in the literature.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—Data mining; I.2.6 [Artificial Intelligence]: Learning—Knowledge acquisition

General Terms

Algorithms, Design, Human Factors

Keywords

Smart homes, ubiquitous computing, sequential patterns, event sequence, episode discovery, person identification.

1. INTRODUCTION

Ubiquitous environments constitute a technological challenge for contemporary research. These environments are characterized by an increasing number of sensors, actuators, displays, devices, and computational elements embedded in everyday objects, and connected through a network. The recent emergence of ubiquitous environments such as smart homes has allowed the provision of housekeeping, assistance and monitoring for chronically ill patients, and enabled persons with special needs and the elderly to receive services in their own home environments [7, 24]. Using such technology can help to reduce costs considerably, and relieve pressure on healthcare systems. However, this technology poses many challenges, such as person identification, activity recognition, assistance, monitoring, and adaptation.

Person identification in ubiquitous environments is a subject of great interest, particularly in smart homes, where specific individuals must be monitored and assisted according to their needs. To deal with this issue, several research projects have been conducted using video and image processing [31, 6, 4]; some other systems are also discussed in [3]. However, these approaches are intrusive and do not preserve personal privacy. By “intrusive”, we mean that individuals are monitored by cameras which invade their privacy, or by other body-worn sensors which diminish their sense of autonomy. Little work has been done on using non-intrusive systems which utilize devices and sensors (for motion, pressure, RFID, etc.) disseminated in the environment [24, 20]. These sensors capture events about the state of the environment and any changes that occur in it. These events constitute a form of sequence. Each sequence of events is associated with a particular activity performed within the environment. The same activity can be performed by the same person in different ways, which means that the same activity can correspond to different sequences of events. This change in the person’s behavior leads to the generation of a set of frequent patterns (episodes) that characterize this person.

Mining event sequences is an important task for the ubiquitous computing community, and for the KDD community as well. In fact, by discovering frequent patterns, the underlying association rules, temporal constraints and progress, changes over time and expected utilities, it becomes possible to characterize the behavior of persons (and objects, such as those most purchased in a supermarket) and automate tasks such as service adaptation, activity monitoring and assistance, and many other complex activities and applications [22].

The general approaches to person identification in ubiq-
uitous environments are either intrusive, or resource- and time-consuming. In this paper we present a new algorithm for person identification, based on sequential pattern mining. Our algorithm consists of three main steps. In the first step, the frequent patterns (episodes) are extracted from the sequential dataset using the Apriori algorithm [2], one of the efficient pattern mining algorithms in the literature [9]. The second step evaluates the extracted episodes. A new scoring function for weighting the extracted episodes is proposed, based on the work of Chang-Rong Lin [16], and adapted to sequential patterns. This step is one of the innovations made by the method proposed in this paper. The weight assigned to an episode indicates the importance of the episode in the event sequence, and these weights improve the classification accuracy in the third step of our algorithm.

The strengths of our approach are, first, that it is based on a fully non-intrusive technology, which broadly preserves personal privacy; and second, that it takes events collected from sensors in the form of sequences, which means that our approach is generic and can be applied to any sequential data. In addition, our approach is infrastructure-independent: this means that it is applicable to evolving and scalable environments because the focus is on the generated sequences, not on the environment itself. We take advantage of the sequential form of events to extract frequent patterns that can be used to characterize each person and distinguish him or her from others living in the same environment. To our knowledge, no other paper in the literature addresses the problem of person identification by using sequential pattern mining with weights assigned to episodes. There is a single report of a study that assigned weights to sequential patterns to classify sequences [5]. However, this work used a very simple scoring function based on the pattern length, and did not take into account patterns of length 1. This motivates our effort to create a non-intrusive approach for automatic person identification, based on sequential pattern mining. We will show, through experiments, how the use of frequent pattern mining in our approach can significantly improve the quality of identification, allowing it to achieve higher classification accuracy than the existing methods.

This paper is organized as follows. Section 2 gives an overview of related work. In Section 3 we introduce our approach, presenting the overall architecture and the proposed algorithm. The experiment is described in Section 4 and the experimental results are presented in Section 5. Section 6 is devoted to a discussion of the results, followed by a conclusion in Section 7.

**Contribution of this paper**

This paper proposes a new algorithm for person identification in ubiquitous environments. While the problem of person identification using non-intrusive technology has recently been studied in a limited way in the literature [20, 33], this is the first comprehensive study which proposes a fully non-intrusive approach based on frequent pattern mining. In addition to the efficient algorithm proposed, our approach is based on the generation of long episodes which better characterize human behavior. The innovation of the proposed algorithm is its assignment of weights for frequent episodes, using a new scoring function that evaluates episodes with respect to the sequence as well as the class to which they belong. The episode weights allow us to find significant episodes in the event sequence, which can be used to characterize the person and distinguish him or her from other people, improving classification accuracy. Moreover, the assignment of weights reduces the dimensionality of the space by allowing consideration of significant episodes only, which is helpful in developing real-time applications. Finally, our approach is validated on real-life data which will be available on the Web very soon. This will make our experiments repeatable.

**2. RELATED WORK**

Several research studies reported in the literature have been interested in person identification in ubiquitous environments. J. Suutala et al. [24] introduced methods for footstep-based person identification using a large pressure-sensitive floor with a sensory system. This approach is related to the biometric identification domain [14, 13], which includes physiological (iris, fingerprints, hand shapes, etc.) and behavioral (handwriting, speech) characteristics. The behavioral characteristics of a walking person are used to model the person’s identity [24]. In this approach, the person identification system is based on sensor measurements derived from a pressure-sensitive floor. The authors used what they call ElectroMechanical Film (EMF) [21], which senses pressure changes affecting its surface and provides footstep profiles of the walking person as an input to the identification system.

Much work on person identification has been done using cameras and machine vision methods. Pfinder [31] is a real-time system for tracking people and interpreting their behavior. The Pfinder system uses a multiclass statistical model of color and shape to obtain a 2D representation of head and hands in a wide range of viewing conditions. In [6], the authors propose a real-time face recognition system for consumer/embedded applications. The system is embedded in an interconnected home environment and allows intelligent servicing via automatic identification of users. This system is based on four principal steps: face detection, model-based facial feature extraction, face normalization, and face recognition by discrimination analysis. Bernardin Keni et al. [4] presented a system for audiovisual multi-person tracking and identification of persons in smart environments. Information from several fixed cameras is fused in a particle filter framework to simultaneously track multiple occupants. In [17], the authors introduced a new model for recognizing people by their gait. This model is based on motion shape, which varies with the type of moving figure and the type of motion. The identification process in this system is based on modeling the walking stride sequence, using consecutive frames from a side-view camera. Different features are calculated from the posture and limb positions of the person and from the frequency and phase presentation of walking [17]. The main drawback of the audiovisual-based approaches remains privacy issues. Indeed, these approaches are intrusive and do not respect personal privacy.

Little work has been done on non-intrusive systems which use different types of devices and sensors (motion, pressure, RFID, etc.) disseminated in the environment. In [20], the authors used a graph-based data mining system to identify inhabitants of an intelligent environment. The activity patterns for individual inhabitants are represented as graphs, which can be used to identify persons. This system is used to identify inhabitants based on observed interactions with the home. Each event in a smart home corresponds to in-
interaction with a device, such as turning the light on or off or opening or closing the door. The event is considered as a device whose state is being changed at a particular time. The authors use the SubdueCLM [20] tool to find concepts describing each inhabitant’s activity pattern. These concepts can then be used to classify new activities. In this way, inhabitants can be identified according to the activity they perform. S. Zhang et al. [33] present a probabilistic learning approach to characterize behavioral patterns for multi-inhabitant smart homes. The authors propose a snowflake schema model to store the smart home activity data. Then, a supervised learning algorithm is proposed to learn the behavioral patterns of these activities. The authors aim to distinguish inhabitants and provide personalized services. However, in this study users are asked to indicate their name, the activity they plan to do and when they will begin it. This makes the recognition process less credible and sidesteps many crucial issues in inhabitant and activity recognition, such as event triggers.

Many of the aforementioned solutions suffer from computational difficulties. For instance, the audiovisual multi-person tracking approach needs very intensive computing to process the audiovisual data, in addition to the increasingly high costs of the equipment employed, such as the cameras needed for recording high-quality videos. The efficiency of the graph-based approach proposed by [20] depends basically on the size of the datasets to be processed. The larger the datasets, the more difficult and time-consuming the graph processing becomes. The use of sequential pattern mining appears to be a promising solution to overcome these drawbacks. Indeed, extracting frequent patterns from sequences can be achieved in a shorter time, as mentioned in [2, 15], which can significantly aid the development of real-time applications for identifying persons and activities.

3. OUR APPROACH

In this section we present our approach for person identification using frequent pattern mining. The overall architecture of our approach is presented in Figure 1. The different phases of our approach are detailed in the next sections.

3.1 Frequent Pattern Mining in Sequential Data Sets

Pattern mining is applied to sequences of various types, including protein, weblog, trace, customer purchase history and event sequences. The sequences most often studied are the event sequences generated by individuals or objects. The goal is to learn the behavior of the individuals or objects in these event sequences and determine how to deal with them. In our case, events correspond to sensor states. For each event, some additional information may be available, such as the sensor name, the sensor state/value, and temporal constraints indicating the event occurrence time. The events contained in a sequence are listed in timestamp ascending order. Table 1 shows an example of events collected in a smart home. Frequent pattern mining in sequential datasets has been the subject of intensive research efforts in the past decade, and several algorithms have been proposed. The first was put forward by Agrawal and Srikant [1], who also developed a generalized and refined algorithm called GSP (Generalized Sequential Patterns) [23], based on the Apriori property [2]. Since then, several sequential pattern mining algorithms have also been proposed for performance improvements [32, 8, 27]. Before describing them, we will introduce some additional notation and some definitions of frequent pattern mining.

3.2 Problem definition

In [28], the problem of frequent pattern mining is defined as follows:

**Definition 1 (Sequential Pattern Mining).** Let \( I = \{i_1, i_2, ..., i_n\} \) be a set of items. A sequence \( S \) is an ordered list of events, denoted by \( (e_1, e_2, ..., e_m) \), where \( e_i \) is an item, that is \( e_i \in I \) for \( 1 \leq i \leq m \). A sequence can also be written as \( e_1, e_2, ..., e_m \).

An event \( e_i \) corresponds to a sensor state. For example, \( e_i = \text{MotionSensorOn} \) or \( e_i = \text{MotionSensorOff} \). From the definition, an item can occur multiple times in different events of a sequence. An event is associated with a timestamp which indicates the occurrence time of the event. For example, \((e_1, 10)\) means that the event \( e_1 \) occurs at time \( t = 10 \).

The length of a sequence is determined by the number of events composing it. A sequence of length \( l \) is called an \( l \)-sequence. For example, \((e_1, 10)(e_2, 20)(e_3, 30)(e_2, 40)(e_4, 50)\) is a 5-sequence.

A sequence \( S_a = a_1, a_2, ..., a_m \) is contained in another sequence \( S_b = b_1, b_2, ..., b_m \) if there exist integers \( 1 \leq i_1 \leq i_2 \leq ... \leq i_m \leq m \) such that \( a_1 = b_{i_1}, a_2 = b_{i_2}, ..., a_m = b_{i_m} \). If \( S_a \) is contained in \( S_b \), then \( S_a \) is called a subsequence of \( S_b \) and \( S_b \) a superset sequence of \( S_a \), denoted by \( S_a \subseteq S_b \).

An input sequence database \( D \) is a set of tuples \((sid, S)\), where \( sid \) is a sequence identifier and \( S \) an input sequence. The number of tuples in \( D \) is called the base size of \( D \), denoted by \( |D| \). A tuple \((sid, S)\) is said to contain a sequence \( S_a \) if \( S_a \) is a superset sequence of \( S_a \).

Table 1: Example of events collected in a smart home

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Sensor Name</th>
<th>State / value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-02-02</td>
<td>12:18:44</td>
<td>MotionSensor16</td>
<td>ON</td>
</tr>
<tr>
<td>2009-02-02</td>
<td>12:18:46</td>
<td>MotionSensor17</td>
<td>OFF</td>
</tr>
<tr>
<td>2009-02-02</td>
<td>12:28:50</td>
<td>DoorSensor12</td>
<td>OPEN</td>
</tr>
<tr>
<td>2009-02-02</td>
<td>12:29:55</td>
<td>ItemSensor03</td>
<td>PRESENT</td>
</tr>
<tr>
<td>2009-02-05</td>
<td>08:05:52</td>
<td>HotWaterSensor-B</td>
<td>0.0448835</td>
</tr>
<tr>
<td>2009-02-05</td>
<td>12:21:51</td>
<td>DoorSensor09</td>
<td>CLOSE</td>
</tr>
<tr>
<td>2009-02-10</td>
<td>17:03:57</td>
<td>ItemSensor03</td>
<td>ABSENT</td>
</tr>
</tbody>
</table>
The absolute support of a sequence \( S_n \) in \( D \) is the number of tuples that contain \( S_n \), denoted by \( sup^D(S_n) \).

Given a specified minimum support called \((\text{min-sup})\), a sequence \( S_n \) is a frequent sequence on \( D \) if \( sup^D(S_n) \geq \text{min-sup} \).

Sequential pattern mining is the process that allows the discovery of all patterns with a particular significance. In our case, these patterns are called episodes. An episode is a collection of events that occur relatively close to each other in a given partial order \([18]\). The concept of episode discovery was first introduced by H. Mannila \([18]\). Formally, the episode concept is defined as follows:

**Definition 2 (Episode).** An episode \( \alpha \) is defined by a triple, \((V_\alpha, \leq_\alpha, g_\alpha)\), where \( V_\alpha \) is a collection of nodes, \( \leq_\alpha \) is a partial order on \( V_\alpha \), and \( g_\alpha : V_\alpha \rightarrow \mathcal{E} \) is a map that associates each node with an event type from a finite set of events \( \mathcal{E} \).

The episode \( \alpha \) is parallel if the partial order relation \( \leq \) is trivial (or empty). The episode \( \alpha \) is serial if the partial order relation \( \leq \) is a total order.

The measure of how often an episode occurs in an event sequence is called the episode frequency. Several methods for defining the episode frequency exist in the literature \([19, 18]\). An episode is considered interesting if it occurs sufficiently often in the event sequence. In our paper, the episode frequency is defined as the number of occurrences of the episode. Figure 2 shows an example of a frequent episode in an event sequence, where \( e_i \) is an event and \( t_i \) is the occurrence time of the event.

![Figure 2: Frequent episode “e\(_1\)e\(_2\)e\(_3\)”](image)

Our goal is to identify the person among the others living in the same environment and to distinguish him or her from the others. For this purpose, our approach uses frequent patterns as a key solution to characterize the person, first, and then to distinguish him or her from the others.

### 3.3 Algorithm

This section presents our proposed algorithm for person identification. Our algorithm is composed of three main phases. The first phase is episode generation. The second is the attribution of weights to the frequent episodes generated in the first phase and the construction of the frequent-episode weight matrix (FEWM). Finally, in the third phase, FEWM is provided to a classification algorithm.

Algorithm 1 requires the following inputs: the sequence database \( D \), the minimum support threshold \( \text{min-sup} \) defined by the user, and the length \( N \) of the episodes to be generated (for example, \( N=5 \) for episodes of length 5). The output of the algorithm is the FEWM matrix.

The steps of the algorithm 1 are described in detail below.

**Step 1:**
In this step, we parse all sequences in the sequence database using the Apriori algorithm, in order to extract frequent episodes. The extracted frequent episodes are used as input for the next step. As mentioned previously, an episode’s frequency indicates how often the episode occurs in the sequence database. There are many ways to define the episode frequency. For example, in \([19]\), the authors defined the episode frequency as the number of fixed-width sliding windows over the time where each contains an occurrence of the episode. \([15]\) proposed a frequency measure based on non-overlapped occurrences. The standard approach used for frequent episode discovery is to use an Apriori-style level-wise procedure.

**Algorithm 1 Person identification algorithm**

**Inputs:**
- Sequence database \( D = \{S_1, S_2, ..., S_m\} \);
- Minimum support threshold \( \text{min-sup} \);
- Length \( N \) of episodes to be generated;

**Outputs:**
- FEWM matrix;

**Steps:**
1. Parse the sequence database to extract frequent episodes using Apriori algorithm;
2. Extract the frequencies of the generated episodes for each sequence in the database, and build the frequent episodes frequency matrix (FEFM);
3. Update the FEFM matrix by attributing weights to frequent episodes, and construct the FEWM matrix;
4. Return FEWM;
5. Classify (FEWM)

For simplicity in the implementation part of this step, we used the TDMiner tool\(^1\), which is a temporal data mining tool developed in java, based on a frequent episode framework. We used the TDMiner tool to count the episode frequency, based on the Apriori-style level-wise procedure, using the fast non-overlapped count algorithm mentioned earlier. The TDMiner tool returns frequent episodes of length 1, 2, .., \( N \) (\( N \) is specified by the user), with the corresponding frequencies. These results are stored in separate files.

The following example shows how episodes are extracted from an event sequence. Consider the following event sequence:

\[
\{(e_1, 10), (e_2, 20), (e_1, 32), (e_3, 35), (e_1, 50), (e_2, 70)\}
\]

Let \( \text{min-sup} = 2 \) be the minimum support specified by the user. The episodes of length 1 that will be extracted are \( e_1 \) and \( e_2 \), given their respective frequencies, \( 3 \geq 2 \), and \( 2 \geq 2 \). In the same way, there are a total of four occurrences of the episode \( e_1e_2 \) of length 2. We list them here:

1. \{(e_1, 10)(e_2, 20)\}
2. \{(e_1, 10)(e_2, 70)\}
3. \{(e_1, 32)(e_2, 70)\}
4. \{(e_1, 50)(e_2, 70)\}

The remaining episodes of length 2 and those of greater size are extracted in the same way. The frequency of an episode is computed in each sequence in the database.

\(^1\)http://neural-code.cs.vt.edu/index.html
Step 2:  
This step uses the episodes extracted in step 1 to build the FEWM matrix. We implemented an application that parses the generated episodes, extracts their frequencies, and builds the FEWM matrix. The columns of this matrix are the episodes generated in step 1, and the rows correspond to the sequences in the sequence database. The greater the episode size, the better the explanation provided by the episode as to the person’s behavior, and the more helpful it is in discriminating the person’s behavior from that of others. Therefore, in our approach we are interested in long episodes, in order to perfectly characterize the person’s behavior. In generating episodes, the Apriori algorithm takes into account the different possibilities for the order relation between events. The reason for considering all these possibilities is that the same task performed by a user in the ubiquitous environment can take several forms (episodes). For example, several sequences of events correspond to the action “add sugar to a cup of coffee”. It could be (take spoon, take sugar, pour the sugar into the cup of coffee), or (take sugar, take spoon, pour the sugar into the cup of coffee). This allows us to study the different behaviors of users when performing tasks.

Step 3:  
This step constitutes the core of our algorithm. The episode weight computation as such is performed in this step. After episode extraction, we have a set of episodes with their corresponding frequencies in each event sequence. Episodes associated with a high frequency are not necessarily more significant than episodes with a low frequency in the event sequence. It is thus important to identify the significance of each frequent episode extracted in step 1.

Given the variability of human behavior, the significance of an episode is based on its significance with respect to the event sequence as well as the class to which it belongs. The significance of an episode with respect to the event sequence, denoted by \( \text{SES}(E, S) \), can be obtained by dividing the frequency of the episode, denoted by \( F_{E,S} \), by the total sum of frequencies of all episodes contained in the event sequence. In the same way, the significance of an episode with respect to the class, denoted by \( \text{SEC}(E, C) \), is obtained by summing up the \( \text{SES}(E, S) \) in every event sequence. The following formulas show the \( \text{SES}(E, S) \) and \( \text{SEC}(E, C) \) computations.

\[
\text{SES}(E, S) = \frac{F_{E,S}}{\sum_{F_{E,S}} F_{E,S}} \quad (1)
\]

where \( FE \) denotes a frequent episode, and \( F_{E,S} \) denotes the frequency of the episode \( E \) contained in the event sequence \( S \).

\[
\text{SEC}(E, C) = \sum_{S \in C} \text{SES}(E, S) \quad (2)
\]

In our case the class \( C \) corresponds to a person. Given the increased number of sensors disseminated in the environment, and the difference in the person’s behavior, a frequent episode with a high \( \text{SEC} \) in one class is not necessarily more important than an episode with a low \( \text{SEC} \) in another class. Therefore, we should normalize the significance of an episode with respect to a class in order to obtain the normalized \( \text{SEC} \) in a class, denoted by \( \text{NSEC}(E, C) \). We use the following formula to compute the \( \text{NSEC} \), where \( \text{MaxSEC}(C) \) and \( \text{MinSEC}(C) \) correspond respectively to the maximum and minimum of the \( \text{SEC} \) of the frequent episodes in the class \( C \). We add 1 to the numerator and denominator to avoid the case where \( \text{MaxSEC}(C) - \text{MinSEC}(C) = 0 \).

\[
\text{NSEC}(E, C) = \frac{\text{SEC}(E, C) - \text{MinSEC}(C) + 1}{\text{MaxSEC}(C) - \text{MinSEC}(C) + 1} \quad (3)
\]

Once \( \text{NSEC} \) is calculated for each frequent episode, we update the frequency of each episode by adding the \( \text{NSEC} \) of each episode to its frequency. The new frequency obtained is called the weight of the frequent episode, denoted by \( WFE(E) \). The weight of each frequent episode is calculated by the following formula, which balances the episode frequency over the class to which the episode belongs:

\[
WFE(E) = F_{E,S} + \text{NSEC}(E, C) \quad (4)
\]

The pseudocode for the implementation of the new scoring function employed to attribute weights to episodes is given in Algorithm 2.

Algorithm 2 Pseudocode of the employed scoring function

**Initialization**: Frequent episode frequency matrix

1: for \( i = 1, ..., N_e \) (for each class) {
2: for \( j = 1, ..., N_s \) (for each sequence in the class) {
3: for \( k = 1, ..., N_e \) (for each frequent episode) {
4: - Compute the \( \text{NSEC}(E, C) \) (compute the significance of the episode with respect to the class \( C \)) using formula 3.
5: - Compute the weight using formula 4.
6: - Update the FEWM Matrix.
7: - Return the FEWM Matrix.
8: end
9: end
10: end

The result of this step is the completed FEWM matrix, which will be provided to a classification algorithm in step 5. As noted earlier, there is only one study reported in the literature that assigns weights for frequent patterns to classify sequences [5]. However, this study employs a very simple scoring function inspired by the work of [26], that computes the weight by dividing the pattern length minus 1 by the number of patterns that describe the corresponding class. This scoring function does not take into account patterns of length 1. Moreover, it cannot determine the significant patterns in a given sequence. In contrast to this approach, our approach can be applied to all episodes, can determine the significant episodes, and gives good classification results.

An example of the content of the FEWM matrix is shown in Table 2.

Table 2: Example of the FEWM matrix

<table>
<thead>
<tr>
<th>Ep 1</th>
<th>Ep 2</th>
<th>Ep 3</th>
<th>Ep 4</th>
<th>Ep 5</th>
<th>Ep 6</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.6665</td>
<td>21.0001</td>
<td>1.0004</td>
<td>1.0004</td>
<td>2.3336</td>
<td>0.0001</td>
<td>U1</td>
</tr>
<tr>
<td>10.0004</td>
<td>12.6665</td>
<td>1.2225</td>
<td>5.0066</td>
<td>3.0333</td>
<td>3.3333</td>
<td>U2</td>
</tr>
<tr>
<td>9.0000</td>
<td>12.1066</td>
<td>3.3190</td>
<td>3.0333</td>
<td>5.0048</td>
<td>3.0048</td>
<td>U3</td>
</tr>
</tbody>
</table>

Step 5:

In this step we apply a classification algorithm such as the decision tree procedure to the FEWM matrix. Note that other supervised learning algorithms can be applied at this stage to perform the classification.
4. EXPERIMENTS

In this section, we present the datasets obtained from experiments conducted at the Domus laboratory [11] and the Testbed smart home [12] to validate our approach.

4.1 Description of the Domus smart home

The Domus smart home is a one-bedroom apartment on the University of Sherbrooke campus. It includes one bedroom, one bathroom, a kitchen, a dining room, and a living room. (See Figure 3.)

In this study, we considered the sensors which are already mounted in the DOMUS apartment. Six zones are defined to cover the different apartment areas, as shown in Figure 3. The number of installed sensors varies depending on the zone of interest. The Domus smart home apartment is equipped with the following sensor categories:

- Infrared (IR) movement detectors: These cover a zone or a part of a zone; for example, in the dining room and living room there is only one IR detector that covers the entire zone, whereas three others are installed in the kitchen, covering the oven, the sink, and the toaster. These sensors provide the user’s location in a zone.
- Pressure detector in the form of tactile carpeting: This sensor, placed in the entrance hall, detects the user moving between the bedroom and living room. There are two paths connecting these two zones: through the kitchen or the entrance hall.
- Light switches: These sensors send an event every time the occupant turns the lights on or off.
- Door contacts: These sensors are placed on the doors, and send an event related to the door state (open or closed).
- Closet contacts: The same as door contacts, these are placed on the cupboards and fridge. They provide an event when their state (open or closed) is changed.
- Flow meters: These provide the states of the taps and the flush toilet. Two are mounted on the cold and hot water taps of the kitchen sink, one on the washbasin cold water tap and another in the flush toilet. They send an event when the tap is opened or closed or the flush toilet is used.

4.2 Experiments Scenario

Six adults (Master’s and Ph.D. students at the University of Sherbrooke) participated in these experiments, which evaluated early morning routines (grooming, breakfast), corresponding to basic tasks of everyday living. Two series of experiments were performed in the Domus smart home apartment. In the first experiments (series 1), the user was asked to perform the early morning routine as he would normally do it at home. In the second set (series 2) he was asked to repeat the same routine, with the introduction of a constraint. The constraint involved learning a tea recipe which takes at most 10 minutes. In series 1, the user came 10 times to the laboratory over two consecutive weeks. After 2 weeks’ break, the user was asked to come for 5 days in one week to perform series 2. In both series, the user was free to use any equipment available in the apartment, and to determine the order of the activities composing his or her routine. Each experiment lasted about 45 minutes. Data were recorded for each user in XML format, using our system recorder framework developed in the Domus laboratory.

5. EXPERIMENTAL RESULTS

We conducted our experiments under the Weka framework (version 3.7.0) [30]. The purpose of this experiment was to observe the general performance of our algorithm and its accuracy in practical environments. The dataset is composed of 55 sequences corresponding to series 1 (5 sequences were damaged and therefore excluded from the experiments), and 30 sequences corresponding to series 2. The length of sequences varies between 100 and 470 events for series 1 and between 210 and 680 events for series 2. The number of sensors installed in the Domus smart home apartment is 36, corresponding to 72 sensor states (ON, OFF, OPEN, CLOSED, ..., etc), in addition to one sensor (pressure detector) with a single state, for a total of 73 sensor states. We used decision trees as our classification algorithm. In fact, the decision tree procedure is one of the most widely used techniques in event sequence classification [25], and its classification time is very low. We used different episode lengths (3, 4, and 5) in order to ensure the efficiency of our algorithm and a reliable evaluation of our approach. The number of episodes used in our experiments depends on the events contained in each sequence. Table 3 shows the minimum and maximum number of episodes extracted in each dataset. In order to enrich our evaluation, we also used another dataset obtained from the Testbed smart home at Washington State University [12]. The dataset corresponds to data gathered from the Testbed smart home, where two participants performed daily living activities such as those involved in breakfast and grooming. More than 160 sensor states are involved when performing these activities. The dataset is composed of 62 sequences gathered during the breakfast activity, and 84 sequences during the grooming activity. The lengths of the sequences vary between 25 and 491 and between 36 and
Table 3: Minimum and maximum number of episodes extracted in each dataset (k = Kilo)

<table>
<thead>
<tr>
<th>DataSet</th>
<th>Episode Length</th>
<th>N=3</th>
<th>N=4</th>
<th>N=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domus Series 1</td>
<td></td>
<td>[0.18 k, 16.533 k]</td>
<td>[0.264 k, 69.501 k]</td>
<td>[0.337 k, 4.646 k]</td>
</tr>
<tr>
<td>Domus Series 2</td>
<td>[1.647 k, 83.874 k]</td>
<td>[0.04 k, 5.035 k]</td>
<td>[0.307 k, 26.929 k]</td>
<td></td>
</tr>
<tr>
<td>Testbed Breakfast</td>
<td>[0.04 k, 1.985 k]</td>
<td>[0.036 k, 9.679 k]</td>
<td>[0.119 k, 57.031 k]</td>
<td></td>
</tr>
<tr>
<td>Testbed Grooming</td>
<td>[0.014 k, 1.332 k]</td>
<td>[0.023 k, 3.034 k]</td>
<td>[0.031 k, 55.24 k]</td>
<td></td>
</tr>
</tbody>
</table>

432 events per sequence for breakfast and grooming, respectively.

Additional tests using the frequency-based (FB) approach [10] and the SubDue framework [20] were performed for comparison purposes, using the same datasets. In the FB approach, the frequency of an episode corresponds to the number of times the episode occurs in the event sequence. There is no scoring function applied to the episode frequency in the FB approach. In our experiments, we used a 10-fold cross-validation strategy to perform the classification.

5.1 SubDue framework

For the SubDue framework, we implemented a program that generates graphs from the event sequences. The resulting graphs are the input of the SubDue framework. For example, for the following event sequence composed of three sensors,

- 2009-05-06 08:40:10 S1 ON
- 2009-05-06 08:42:43 S2 OFF
- 2009-05-06 08:45:33 S3 OPEN

the following graph representation is generated:

![Example of graph representation of event sequence data](image)

As we can see from figures 6(a), 6(b) and 5, the identification accuracy of our approach is generally between 93 and 100, which is a high accuracy score compared to the FB approach and the SubDue framework. This is further confirmation for the effectiveness of our approach.

A comparison of the classification results obtained for each dataset using the three approaches is shown in figures 7(a), 7(b), 7(c) and 7(d). The SubDue framework performs poorly in a multi-class problem (Domus dataset with 6 persons), and relatively well in a bi-class problem (Testbed dataset with 2 persons).

From figures 7(a), 7(b), 7(c) and 7(d), we observe that episodes of greater size yield relatively higher accuracy than those of smaller size. In practice, the larger the episode, the better it reflects the task being performed, and the more understanding is provided. The results obtained in our experiments confirm this observation.

6. DISCUSSION

Our approach is aimed at identifying persons in a ubiquitous environment. Given the variability of human behavior, characterizing a person from sensor states contained in a sequence is a challenging and intricate task. By using the frequent episode principle and exploiting its inherent benefits, namely episode discovery and temporal constraints, identification can be accomplished by considering the frequent episodes extracted from event sequences. However, using frequent episodes alone, without deep analysis and study, does not necessarily guarantee good identification, as shown in figure 6(b). In this context, therefore, additional processing and analysis of frequent episodes are necessary and important.

In our approach, the frequent episodes extracted are supported by an additional piece of information: the weight of the episode. This information can be used to prioritize episodes based on their importance, which can help in improving the accuracy of person identification. Furthermore, incorporating domain-specific knowledge into the episode discovery process can further enhance the performance of our approach.
Table 4: Classification accuracy using our approach, FB approach and SubDue framework

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy (%)</th>
<th>Episode Length</th>
<th>Frequency Based approach</th>
<th>SubDue framework</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N=3</td>
<td>N=4</td>
<td>N=5</td>
</tr>
<tr>
<td>Domus Series 1</td>
<td>94.11</td>
<td>100</td>
<td>98.11</td>
<td>84.16</td>
</tr>
<tr>
<td>Domus Series 2</td>
<td>96.55</td>
<td>93.10</td>
<td>96.29</td>
<td>37.93</td>
</tr>
<tr>
<td>Testbed Breakfast</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>62.50</td>
</tr>
<tr>
<td>Testbed Grooming</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>47.50</td>
</tr>
</tbody>
</table>

Figure 6: Classification accuracy in our approach and FB approach

(a) Our approach
(b) FB approach

Figure 7: Comparison of the classification accuracy in our approach, FB approach and SubDue framework

(a) Domus Series 1
(b) Domus Series 2
(c) Testbed Breakfast
(d) Testbed Grooming
class, the assignment of weights allows us to balance the episode frequency by increasing the lowest frequency, making this episode more important. The more important the episode, the higher its weight, and the more discriminative it will be. As shown in Figure 6(a), the assignment of weights significantly improves the classification results.

The other benefit of weighting is the detection of interesting (significant) episodes. By significant episodes, we mean those that can best characterize the person and have the most discriminative power. This direction, which was not exploited in the present study, is the subject of our ongoing work. Indeed, the assignment of weights to episodes poses two challenges: detecting significant episodes, and reducing the dimensionality of the space using only these significant episodes. The development of real-time applications with a low dimensionality space thus constitutes a major contribution to the ubiquitous environment and KDD communities.

Our approach has broad social, economic and academic impacts. It can be used in ubiquitous environments such as smart homes and smart hospitals to characterize the inhabitants and adapt a variety of tasks accordingly. Our approach can also be used to characterize persons and products in supermarkets and study the purchase trends for each product. Finally, our approach contributes to the literature by introducing new algorithms and methods that can be used by the scientific community for teaching and/or research purposes.

7. CONCLUSION

In this paper, we presented a new efficient algorithm for person identification in ubiquitous environments, based on frequent pattern mining. Specifically, we analyzed data gathered from two ubiquitous environments: the Domus and Testbed smart homes. In order to identify persons, we first extracted frequent episodes from the datasets. Next, we assigned weights to these episodes. Finally, we applied a classification algorithm to classify the frequent episodes. We concluded that episode weighting is more appropriate than the episode frequency-based approach, and achieves high classification accuracy.

We illustrated the effectiveness and suitability of our approach on multiple datasets extracted from two smart homes, Domus and Testbed. The experimental results show that our approach is able to identify persons with a significantly higher accuracy than the frequency-based approach or the SubDue framework. Moreover, we tested our approach on different episode lengths, and observed that episodes of greater size yield relatively better accuracy than those of smaller size. In addition, our algorithm is based on sequential pattern mining, which can be used in various applications involving large amounts of sequential data, such as protein classification, temporal sequences, and financial sequences. We used our approach to detect significant episodes in event sequences. Indeed, the weights assigned to frequent episodes are used to distinguish significant episodes from non-significant ones. Episodes with greater weights are more significant than those with lower weights. Therefore, a person’s behavior can be characterized using significant episodes that can distinguish him or her from others.

8. KNOWLEDGEMENT

The authors wish to thank Virginie Charette, a professor at the Department of Mathematics of the University of Sherbrooke for her help.

9. REFERENCES


