COM: A Method for Mining and Monitoring Human Activity Patterns in Home-based Health Monitoring Systems

Parisa Rashidi, University of Florida
Diane J. Cook, Washington State University

The increasing aging population in the coming decades will result in many complications for society and in particular for the health care system due to the shortage of health care professionals and health care facilities. To remedy this problem, researchers have pursued developing remote monitoring systems and assisted living technologies by utilizing recent advances in sensor and networking technology, as well as in the data mining and machine learning fields. In this paper, we report on our fully automated approach for discovering and monitoring patterns of daily activities. Discovering and tracking patterns of daily activities can provide unprecedented opportunities for health monitoring and assisted living applications, especially for older adults and individuals with mental disabilities. Previous approaches usually rely on pre-selected activities or labeled data to track and monitor daily activities. In this paper, we present a fully automated approach by discovering natural activity patterns and their variations in real life data. We will show how our activity discovery component can be integrated with an activity recognition component to track and monitor various daily activity patterns. We also provide an activity visualization component to allow caregivers to visually observe and examine the activity patterns using a user-friendly interface. We validate our algorithms using real life data obtained from two apartments during a three month period.

Categories and Subject Descriptors: H.2.8 [**Information Systems**]: DATABASE—*Data mining*; I.2.6 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Learning*; H.4.m [**Info. Systems**]: INFORMATION SYSTEMS APPLICATIONS—*Miscellaneous*

General Terms: DESIGN, EXPERIMENTATION, ALGORITHMS

Additional Key Words and Phrases: Assisted Living Technology, Smart Environments, Health Monitoring, Sequence Mining

1. INTRODUCTION

The growing aging population in the coming decades will result in many complications for families, society and the government. According to the latest available statistics from 2009, the older population of the United States numbers 39.6 million (about 12.9% of the U.S. population), and this number is expected to grow to 19% of the population by 2030 [AOA 2011]. The increasing aging population will result in many new challenges and problems, such as the shortage of health care professionals and care facilities, an increase in age related diseases and rising health care costs. An estimated 9% of adults age 65+ and 50% of adults age 85+ need assistance with everyday activities due to physical and cognitive decline. For instance, Alzheimer's disease is a major concern for the older population. An estimated 5.4 million Americans had Alzheimer's disease in 2011, and by 2040 approximately 11 million people will be suffering from cognitive decline due to Alzheimer's [Hebert et al. 2001].

Author's addresses: P. Rashidi, Computer and Information Science and Engineering Department, University of Florida, prashidi@cise.ufl.edu; D. J. Cook, Electrical Engineering and Computer Science Department, Washington State University, cook@eecs.wsu.edu.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

As a result of the increasing aging population and its related mental and physical complications, these is a rising interest in assisted care and health monitoring technologies based on ubiquitous computing and ambient intelligence in order to lower health care costs and to avoid unnecessary institutionalization. An example of assisted care systems is a remote health monitoring and intervention system. These systems are able to monitor and track the daily activities of older adults. Activities of Daily Living, or for short ADL [Reisberg et al. 2001], is a term that is used widely by geriatric specialists to refer to basic self-care activities that determine the independence of an older adult in daily life. The basic set of ADL activities include personal hygiene, dressing, self feeding, and taking medicine. The ability to perform ADLs independently and completely on a regular basis provides a measurement of the functional status of a person and the ability to live independently at home [McDowell and Newell 1996]. In a recent report by Rialle et al. [Rialle et al. 2008], family caregivers of Alzheimer disease's patients were surveyed about assisted care technologies. Most caregivers ranked activity identification and tracking at the top of their list of needs.

In response to this need, researchers are building remote ADL monitoring and intervention systems based on ambient intelligence technology. Not only can such systems identify and track daily activities, but they also can provide timely prompts if the resident has difficulty completing a task or remembering the activity steps. Such systems allow elderly and people with special needs to live more independently in the comfort of their own home, while reducing the burden of caregivers by avoiding constant monitoring. The widespread use of such systems in the future will also significantly reduce the health care costs by avoiding unnecessary institutionalization.

In this paper, we use smart home technology [Rashidi and Cook 2009; Cook et al. 2003; Doctor et al. 2005; Abowd and Mynatt 2004; Helal et al. 2005] for monitoring the activities of older adults. A smart home can be considered as any regular home which has been augmented with various types of sensors such as infrared motion sensors for tracking the movement of residents at home. Such sensors can provide rich context information about the environment and its inhabitants. Smart homes can be used for various purposes, such as security, energy conservation, home automation, and of course for assisted care of older adults [Gopalratnam and Cook 2007; Rashidi and Cook 2008; Yiping et al. 2006; Wren and Munguia-Tapia 2006].

The main component of an ADL monitoring system is an "activity recognition" module. The activity recognition system recognizes resident's activities from low level sensor data. A number of supervised activity recognition methods have been proposed in the literature [Mozer et al. 1998], [Brdiczka et al. 2005], [Philipose et al. 2004; Zhao et al. 2010], [Maurer et al. 2006], [Liao et al. 2005], [Inomata et al. 2009]. Supervised methods have the drawback that they require labeled data for training the recognition algorithm. Annotating human activity data from low level sensors is a very time consuming and laborious task and can limit the scalability of the system. Recently a few unsupervised methods have been proposed to tackle the data annotation problem [Gu et al. 2009], [Pei et al. 2007], [Huynh and Schiele 2006]. Though these methods address the data annotation problems, they consider a simplified version of the problem by ignoring the real life properties of human activity, such as its sequential nature, the existence of disrupted patterns and variations of the same pattern.

In this paper, we propose an unsupervised method for discovering discontinuous and varied-order activity patterns in real world settings. Our method is called COM, standing for **C**ontinuous, varied **O**rder, **M**ulti Threshold activity discovery method. COM not only discovers discontinuous (disrupted) activity patterns and their variations, but it also handles other real world complications, such as vastly different frequencies in different regions of the home. COM is able to find a higher percentage of the frequent patterns, thus achieving a higher recognition rate. It eliminates the need for configuring most of the parameters by the user, resulting in a more automated approach. We also provide a pattern visualizer component to visualize the activity patterns and their variations. The visualization component helps the end users and caregivers to better identify the variations of the patterns and to detect the abnormal or suspicious cases.

2. RELATED WORK

2.1. Smart Environment

Relying on advances in sensor technology, as well as in data mining and machine learning fields, nowadays many researchers are working on smart environments [Cook and Das 2004]. Smart environments provide a way to respond to the needs of the residents in a context aware manner. A smart environment is equipped with different types of sensors which allow the system to collect data on inhabitants activities and environmental situations. Some of the efforts for realizing such smart environments have been demonstrated in actual physical testbeds such as our CASAS project [Rashidi and Cook 2009], MavHome project [Cook et al. 2003], Gator Tech Smart House [Helal et al. 2005], iDorm [Doctor et al. 2005], and Aware Home [Abowd and Mynatt 2004].

Activity data in a smart environment can be captured via different mediums depending on the target task. Activity data at smart home can be collected using ambient sensors such as infrared motion sensors to track the motion of residents around home [Rashidi and Cook 2009]. Additional ambient sensors such as temperature sensors, pressure sensors, contact switch sensors, water sensors and smart powermeters can provide other type of context information. For recognizing residents' interaction with the key objects, some researchers have used object sensors such as RFID tags [Tapia et al. 2004a; Philipose et al. 2004]. Besides ambient sensors, surveillance cameras and other types of image capturing devices such as thermographic camera have been used for activity recognition [Stauffer and Grimson 2000]. Our algorithms are designed to handle data provided from any of these classes of sensors. However, in our testbeds we utilize sensors that do not require frequent battery charging and do not raise privacy concerns among users [B.K. Hensel and Courtney 2006].

2.2. Activity Recognition

One of the most important components of an automatic monitoring system based on smart home technology is "activity recognition". An activity recognition algorithm recognizes resident activity patterns from low level sensor data. It is usually given a sequence of sensor readings. In training phase, each sensor event might be also accompanied by its corresponding activity label. Table I shows several "labeled" example sensor readings from our testbeds.

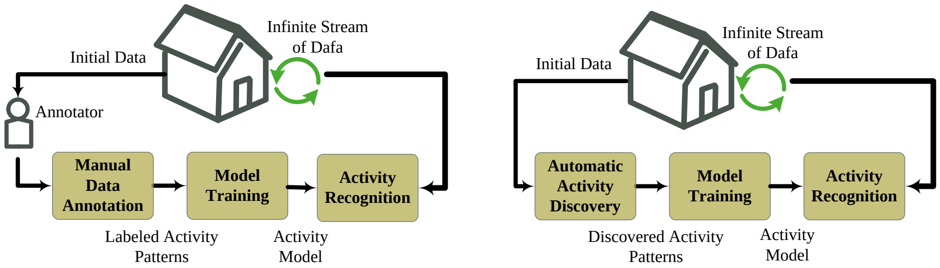
Table I: Example labeled sensor data.

Timestamp	Sensor ID	Label
7/17/2011 09:52:25	$M004$	Personal Hygiene
7/17/2011 09:56:55	$M030$	Personal Hygiene
7/17/2011 14:12:20	$M015$	None

As depicted in Table I, a sensor event can be part of a labeled activity, e.g. the first two sensor events. It also can have no activity labels, such as the third sensor event. The goal of an activity recognition algorithm is to predict the label of a sensor event or a sequence of sensor events, e.g. "Personal Hygiene" for $M004 - M030$ sequence.

The majority of activity recognition algorithms are supervised, i.e. they rely on labeled data for training [VG et al. 2008]. Some researchers have published results of experiments in which the participants are required to manually note each activity they perform at the time they perform it [Liao et al. 2005; Tapia et al. 2004a; Philipose et al. 2004]. In other cases, the experimenters told the participants which specific activities should be performed, so the correct activity labels were identified before the sensor data was even collected [Cook and Schmitter-Edgecombe 2009; Gu et al. 2009; Maurer et al. 2006]. Occasionally, the experimenter manually inspected the raw sensor data to annotate it with a corresponding activity label [Wren and Munguia-Tapia 2006]. Researchers have also used methods such as experience sampling [Tapia et al. 2004b] where subjects carry a personal digital assistant (PDA) as self-report devices.

The supervised activity recognition methods range from simple methods such as naive Bayes [Brdiczka et al. 2005] based on sensor events independence assumption, to more recent and sophisticated methods such as conditional random fields [Philipose et al. 2004; Zhao et al. 2010] which model the sensor events as probabilistic sequences. Other notable supervised methods include decision trees [Maurer et al. 2006], neural networks [Mozer et al. 1998], Markov models [Liao et al. 2005], and dynamic Bayes networks [Inomata et al. 2009]. Despite their prevalence, supervised methods do not scale up well in real world. First, the assumption of consistent pre-defined activities does not hold in reality. Due to physical, mental, cultural, and lifestyle differences [Wray and Laird 2003] not all individual perform the same set of tasks. Even for the same pre-defined activity, different individuals might perform it in vastly different ways, making the reliance on a list of pre-defined activities impractical due to the inter-subject variability. Thus data needs to be annotated for each individual person and for each task. Annotating activity data is a very time consuming and laborious task. Therefore unsupervised approaches seem to be more suitable for activity recognition in a normal day-to-day setting. Figure 1 shows the differences between supervised and unsupervised methods more clearly.



(a) Supervised model.

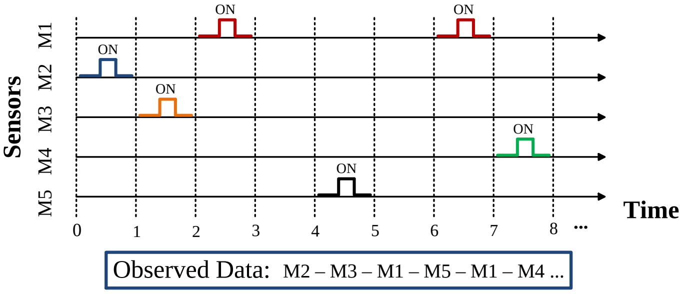
(b) Unsupervised model.

Fig. 1: Activity recognition models.

Recently, a few unsupervised methods have been proposed to tackle the data annotation problem, such as the frequent sensor mining method [Gu et al. 2009], simultaneous frequent-periodic pattern mining method [Rashidi and Cook 2009], episode discovery [Heierman and Cook 2003], activity modeling based on low dimensional Eigenspaces [Schiele 2006], multi dimensional motif discovery [Vahdatpour et al. 2009; Zhao et al. 2010], mixed discriminative and generative methods [Huynh and Schiele 2006], probabilistic models [Barger et al. 2005; Dimitrov et al. 2010] and retrieving activities' definitions using Web mining [Perkowitz et al. 2004; Palmes et al. 2010].

ID	Items
1	$\{A, B, F, G, H\}$
2	$\{D, F, X\}$
3	$\{A, C, X, Y\}$

(a) Transaction data.



(b) Sensor data as a single stream.

Fig. 2: Transaction data vs. sensor data.

Though these methods address the data annotation problem, they consider a simplified version of the problem by ignoring the real world nature of data such as its sequential form, possible disruptions (a phone call in the middle of meal preparation), or variation of the same pattern. For example, Hayes, et al. [Hayes et al. 2007] note that variation in the overall activity level at home is correlated with mild cognitive impairment. This highlights the fact that it is important for a caregiver to be able to monitor all the activities and their variations. Unlike conventional sequential mining methods [Agrawal and Srikant 1995; Zaki 2001; Pei et al. 2004; Yang et al. 2006; Chiu et al. 2004] which deal with transactional data, activity sequence patterns need to be discovered from a “single” stream of sensor data with no clear boundaries between consecutive activities. As a result, the length of the patterns can be highly variable. These problems turn activity mining problem into a challenging problem where sequence patterns can be disrupted by irrelevant events, they can have many variations in terms of step ordering or duration, and their support might greatly vary in different areas of the space.

2.3. Sequence Data Mining

Sequential pattern mining has been studied for more than a decade [Agrawal and Srikant 1995] and many methods have been proposed for finding sequential patterns in data [Agrawal and Srikant 1995; Pei et al. 2001; Wang and Han 2004; Masseglia et al. 1998]. All of these approaches assume that data is in a transactional format. However, input data in a smart environment is not usually in transactional format. Figure 2 depicts the difference between transaction data and sensor data. In case of transaction data, each single transaction is associated with a set of items and is identified by a transaction ID, making it clearly separated from the next transaction. Sensor data, on the other hand, has no boundaries separating different activities from each other. Instead it is just a single stream of sensor events with no clear boundaries between different activities.

Approaches proposed by the sensor mining community [Papadimitriou et al. 2003; Loo et al. 2005] try to turn sensor data into a transactional dataset using techniques such as the Apriori technique [Agrawal and Srikant 1995] to group frequent events together. Another method is to simply use fixed or varied clock ticks [Loo et al. 2005]. In our scenario, using such simple techniques does not allow us to deal with complex activity patterns which can be discontinuous, varied order, and of arbitrary length. In our solution, we group together the co-occurring events into varied-order discontinuous activity patterns.

3. MODEL

Our input data is a sequence of sensor events e in the form $e = \langle t, s\rangle$ where t denotes a timestamp, and s denotes a sensor ID. An example showing several sensor events was depicted in Table I. Each sensor ID is associated with its room name (e.g. kitchen) which we will refer to as a location tag L. We define a pattern instance as a sequence of n sensor events $\langle e_1, e_2, ..e_n\rangle$. A pattern itself represents the collection of all of its instances. An example of an activity pattern such as meal preparation may appear as $\langle M005, M003, M001\rangle$ where $M005$, $M003$ and $M001$ refer to sensors in the kitchen. One variation of such a pattern is $\langle M003, M002, M001\rangle$. Note that we use the terms activity and pattern interchangeably. As mentioned before, in contrast to the supervised approaches which reply on labeled data [van Kasteren et al. 2008; Chan et al. 2008; Vail et al. 2007; Tapia et al. 2004b], we assume that our data is not annotated and the activity boundaries are not specified.

Our objective is to develop a method which can automatically discover real life patterns of resident's activities, even if the patterns are somehow discontinuous or have different event orders across their instances. For example, consider the "meal preparation" activity. Most people will not perform this activity in exactly the same way each time, rather some of the steps or their order might be changed (varied order pattern). In addition, the activity might be interrupted by irrelevant events such as answering the phone (discontinuous pattern).

Our method improves upon our previous method called DVSM [Rashidi et al. 2011] which is able to find discontinuous and varied order patterns in the data. Similar to previous unsupervised methods, DVSM works best when using data collected under controlled conditions and in artificial settings. In contrast, DVSM faces difficulty mining real life data. For example, it has difficulty discovering activities where heterogeneous sensors are used (such as motion sensors in conjunction with contact switch sensors). Our new method is named COM which stands for **C**ontinuous, varied **O**rder, **M**ulti Threshold activity discovery method. COM not only discovers discontinuous patterns and their variations, but is also able to better handle real life data by dealing with different frequencies/sensors problem. It is able to find a higher percentage of the frequent patterns, thus achieving a higher recognition accuracy rate. Also by pruning the irrelevant patterns based on mutual information, it only retains the most relevant variations of a pattern, thereby reducing the number of irrelevant variations. It eliminates the need for configuring most of the parameters by user, such as the percentage of the top frequent sensor events which should be used to discover the activity patterns, or the support threshold for frequent events, or the number of activities. We also provide a pattern visualizer component to visualize the activity patterns and their variations in order to help the users/care-givers to better monitor the variations of the patterns and to help them detect the abnormal or suspicious cases.

The architecture of the system can be seen in Figure 3, including its mining, clustering, recognition and visualization components. Note that after discovering activity patterns, we then cluster discovered patterns to provide a more concise and compact representation. The clusters are then used to track and recognize the resident's activities. In the following section, we will provide a more detailed description of each one of the mining, clustering, recognition and visualization components.

3.1. Discovering Patterns

Our mining algorithm is able to discover activity patterns and their variations from sensor data, even if the patterns exhibit some sort of discontinuity. For example, as depicted in Figure 4, pattern $\langle a, b\rangle$ can be discovered from instances $\{b, x, c, a\}$, $\{a, b, q\}$, and $\{a, u, b\}$, despite the fact that the events are discontinuous and have varied orders.

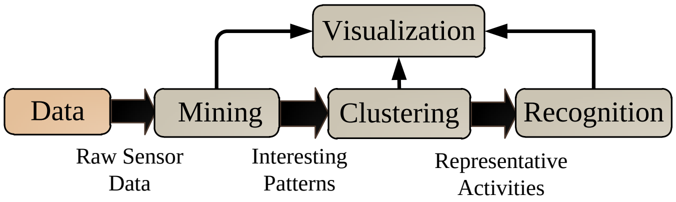


Fig. 3: Main components of COM.

We denote a general pattern P as the pattern that comprises of all of its n variations, where each variation is denoted by P_i. It should be noted that a continuous pattern is discovered by our algorithm as a special case of discontinuous patterns with a discontinuity of zero. It also should be noted that our approach is different than frequent itemset mining, as we take into account the order of events.

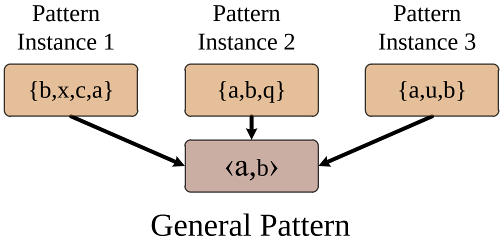


Fig. 4: A general pattern can consist of several variations.

3.1.1. Mining Method Details. First we create a reduced dataset D_r from the input data $\mathcal{D}$. The reduced dataset only contains frequent sensor events used for pattern construction. In the DVSM algorithm, the user has to specify what percentage of the top frequent events (α) should be used. Then by using a global support threshold on the frequency of sensor events, the reduced dataset is created. Taking such an approach will result in ignoring the problem of “rare sensors”. In our studies, we found out that different regions of a home, as well as do different types of sensors, exhibit different sensor frequencies. In our new solution, we do not require the user to identify α, rather we automatically compute several support thresholds for different regions of the home and for different types of sensors. We will explain this problem in more detail in subsequent sections.

After constructing D_r, COM slides a window of size 2 across D_r to find patterns of length 2. After this first iteration, the whole dataset does not need to be scanned again. Instead, COM extends the patterns discovered in the previous iteration by their prefix and suffix events. It then matches the extended pattern against the already-discovered patterns in the same iteration, to check if the extended pattern is a variation of a previous pattern, or if it is a new pattern [Rashidi and Cook 2009]. To facilitate comparisons, we save general patterns along with their discovered variations in a hash table.

At the end of each iteration, we prune infrequent, highly discontinuous or irrelevant variations of a general pattern. We also prune the infrequent, highly discontinuous or non-maximal general patterns. We identify a general pattern/ variation to be interesting if it has a minimum compression value according to the Minimum Description Length (MDL) principle [Rissanen 1978]. The minimum description length principle advocates that the pattern which best describes a dataset is the one which maximally compresses the dataset by replacing instances of the pattern by pointers to the pattern definition. Figure 5 illustrates this compression concept in our algorithm better.

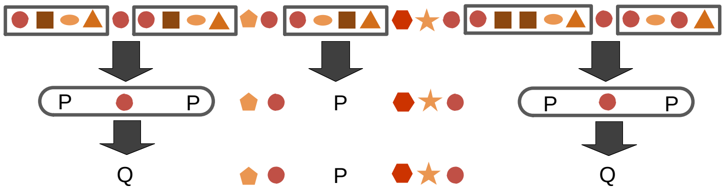


Fig. 5: A sequence pattern (P) is identified and used to compress the dataset. A new best pattern (Q) is found in the next iteration of the algorithm.

However, since we allow discontinuities to occur, each instance of the pattern needs to be encoded not only with a pointer to the pattern definition, but also with a continuity factor, Γ. The discontinuity of a pattern instance is defined as the number of bits required to express the gaps in definition of the pattern. It can be defined in a top-down manner, starting from the component events of a pattern instance [Rashidi et al. 2011]. The continuity between each two events of a pattern instance is defined in terms of the average number of the infrequent events separating the two events. The more the separation between the two events, the less will be the continuity. For a pattern variation, the continuity is based on the average continuity of its instances. For a general pattern the continuity is defined as the average continuity of its variations.

By including the continuity concept in the definition of the patterns, the compression value c for a general pattern P is defined as in Equation (1). Similarly Equation (2) defines the compression value c^v for a variation P_i. Here L denotes description length according to MDL principle as the number of bits required to encode an entity. The length of compressed dataset using pattern definition is reflected in term $L(\mathcal{D}_r|P)$.

$$c(P) = \frac{L(\mathcal{D}_r) * \Gamma_P}{L(P) + L(\mathcal{D}_r|P)} \tag{1}$$

$$c^v(P_i) = \frac{(L(\mathcal{D}_r|P) + L(P)) * \Gamma_{P_i}}{L(\mathcal{D}_r|P_i) + L(P_i)} \tag{2}$$

We transform the compression values to be in range of $[0..1]$ via the so-called softmax scaling technique [Pyle 2005], as in Equation (3).

$$s(c(P)) = \frac{1}{1 + \exp(c(P))} \tag{3}$$

Patterns with high compression values are flagged as “interesting”, while patterns with low compression values are discarded. In other words, we are looking for patterns which are longer, more frequent and more continuous (less gaps within patterns). The same method is applied for pruning variations of a general pattern.

In addition, by computing the mutual information [Guyon and Elisseeff 2003] between the general pattern and each of its sensor events, we are able to find the set of core sensors for each general pattern. Finding the set of core sensors allows us to prune the “irrelevant” variations of a pattern which do not contain the core sensors.

Equation (4) computes the mutual information between the general pattern and each of its sensor events. Here $Pr(x)$ refers to the probability of a random variable x.

$$MI(s, P) = Pr(s, P) * \log \frac{Pr(s, P)}{Pr(s)Pr(P)} \tag{4}$$

At the end of each iteration, we also prune redundant "non-maximal" general patterns; i.e., those patterns that are totally contained in another larger pattern. This multi-stage pruning process considerably reduces the number of discovered patterns. We continue extending the patterns by prefix and suffix until no more interesting patterns are found. This allows us to find variable length patterns. A post-processing step records attributes of the patterns, such as event duration and start time.

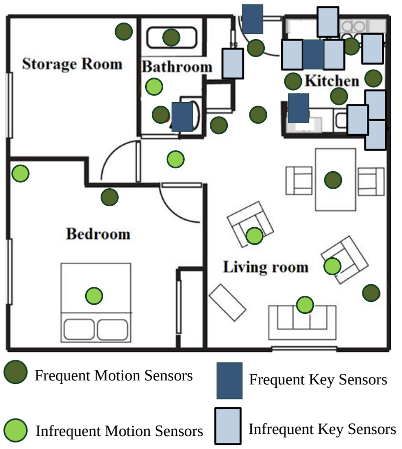
3.1.2. Multiple Support Thresholds. In mining real life activity patterns, the frequency of sensor events can vary across different regions of the home or other space. If the such differences are not taken into account, then the patterns that occur in less frequently used areas of the space might not be discovered. For example, if the resident spends most of his/her time in the living-room during the day and only goes to the bedroom for sleeping, then the sensors will be triggered more frequently in the living-room area than in the bedroom. Therefore when looking for frequent patterns, the sensor events in the bedroom might be ignored and consequently the sleep pattern might not be discovered. The same problem happens with different types of sensors. Usually, motion sensors are triggered much more frequently than other type of sensors such as cabinet sensors. This problem is known as the "rare item" problem in market basket analysis and is usually addressed by providing multiple minimum support values [Liu et al. 1999]. For example, a web page might have a lower chance of being visited by visitors, but we still might be interested in finding click patterns in such pages.

In our solution, each different region of home is identified by a location tag L which represents the functional areas such as bedroom, bathroom, etc. The sensors are also categorized depending on their type. In our experiments, we categorized the sensors into motion sensors and interaction-based sensors. The motion sensors are used to track the motion of a person around a home, e.g. infrared motion sensors. Interaction-based sensors, as we will call them "key sensors", are the non-motion tracking sensors such as cabinet sensors or RFID tags on items. Based on observing and analyzing sensor frequencies in multiple smart homes, we found that a motion sensor might have a higher chance of being triggered than a key sensor in some regions. Hence we will derive separate minimum sensor supports for different sensor categories.

The type and the location tag of all sensors is passed on to COM as an initial configuration file. For each region and category, the minimum acceptable frequency is automatically derived as the average frequency for that specific region and category. More specifically, we denote the minimum acceptable frequency support of the key sensors for each region as $\bar{f}_k$. It is computed as the average frequency of key sensors in that region. Similarly, $\bar{f}_m$ denotes the minimum acceptable frequency support for motion sensors in a specific region. It is computed as the average frequency of motion sensors in that specific region. Figure 6 shows a depiction of selected sensors and the minimum acceptable frequency supports for one of the smart homes used in our experiments. If a region contains only motion sensors and no key sensors, we denote its $\bar{f}_k$ as N/A. One can clearly see that the motion sensors are activated more frequently in the living room than in any other area of the home, such that the frequency support of motion sensors in the kitchen is half of the frequency support in the living room. Also, we can see that the motion sensors in the kitchen are activated 3 times more than the key sensors such as cabinet doors or the refrigerator door. Of course, these results might

Location	$\overline{f}_k$	$\overline{f}_m$
Kitchen	0.01	0.03
Entrance	0.02	0.02
Bathroom	0.02	0.02
Bedroom	N/A	0.03
Living room	N/A	0.06

(a) frequencies.



(b) Frequent sensor on home map.

Fig. 6: Frequent sensors are selected based on the minimum acceptable regional support. The frequencies were normalized in the range of [0..1].

be different in case of a different smart home and a different resident. For example, in a second smart home, we might observe that the sensors in the workspace are activated much more frequently than the sensors in the living room. Nonetheless, our algorithms allow us to automatically compute multiple support thresholds in different situations, thus tailoring our solution to different spaces.

3.2. Clustering

After discovering activity patterns, we cluster them to get an even more compressed representation. Though we group similar pattern variations in the mining stage, still their grouping is solely based on the structural similarity in terms of common sensor events. Two similar patterns with slightly different trajectories might activate different sets of sensors. So they will be considered as separate patterns even if they exhibit high similarity in start time, duration and occurrence locations. To remedy this problem, we use a clustering algorithm. The clustering algorithm groups patterns in terms of their structure, start time, duration and regional similarity. Clustering also addresses the problem of discovering too many similar patterns where the sheer number of patterns makes it difficult to analyze the true underlying major salient ideas.

Our clustering algorithm is similar to conventional hierarchal agglomerative clustering techniques [Tan et al. 2005], although it does not form the complete hierarchy. The algorithm is depicted in Algorithm 1.

Agglomerative clustering techniques build a hierarchy from the individual elements by progressively merging clusters, until all data ends up in one cluster. Here, we do not continue the hierarchal clustering up to the point of reaching a single cluster, rather the clustering continues until the similarity between the two closest clusters drops below a threshold ζ. This gives us a set of clusters at the highest level of the hierarchy. After forming the clusters, the cluster centroids at the highest level are used to track and recognize resident's activities. Using such a clustering method, the user no longer has to provide the number of clusters in advance. We use a group-average link method [Tan et al. 2005] to compute the proximity matrix based on the similarity measure defined in Equation (5).

ALGORITHM 1: Clustering Method

```
Input: $\mathcal{P}$
Output: $\mathcal{C}$
// Each pattern is considered as a cluster at first
$\mathcal{C} = \mathcal{P}$ ;
Compute Proximity Matrix, $m$ ;
repeat
    $sim = \max(m[p,q]) \quad \forall p, q \in \mathcal{C}$ ;
    if $sim > \zeta$ then
        merge $p, q$ ;
    end
    Update $m$ ;
until $sim > \zeta$;
```

$$\Upsilon(i,j) = \Upsilon_t[i,j] + \Upsilon_d[i,j] + \Upsilon_{\mathcal{L}}[i,j] + \Upsilon_S[i,j] \tag{5}$$

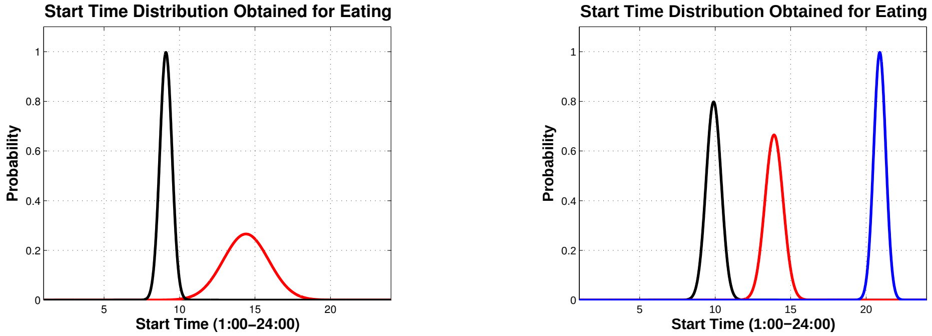
In Equation (5), Υ_t refers to the start time similarity (if the two activities happen at similar times, e.g., both around noon), Υ_d refers to duration similarity (if the two activities have similar durations), $\Upsilon_{\mathcal{L}}$ refers to regional similarity (if the two activities happen in similar locations, e.g., both in the kitchen), and Υ_S refers to structure similarity (if the two activities have similar structure in terms of sensors). We normalize $\Upsilon(i,j)$ to fall within the range $[0..1]$. For simplicity, we have chosen the similarity values to have equal effects; however, it is possible to define $\Upsilon(i,j)$ as a weighted average.

In our model, start time is represented in terms of a mixture normal distribution with means $\Theta = \langle \theta_1..\theta_r \rangle$. The use of multiple distributions allows us to better capture the variability in start time of activities. Two example start time mixture distributions can be seen in Figure 7 for the "eating" activity. We can observe that by using a normal mixture model, we are able to capture breakfast, lunch and dinner times as the regular meals for the inhabitants. We represent start time θ in an angular form Φ measured in radians instead of a linear representation. This allows for time differences to be represented correctly (2:00 am will be closer to 12:00 am than to 5:00 am). The similarity between the two start time distributions is calculated using Equation (6).

Including temporal information about activities can be quite crucial for the activity recognition step. For example, in order to distinguish between hand-washing activity and dish-washing activity in the same kitchen sink, one needs to rely upon temporal information. The reason is because almost all contextual information such as sensor activation would be the same for the two activities. However, by taking into account the fact that dish-washing usually takes longer than hand-washing activity (reflected in duration distributions), we are able to properly distinguish these two activities. Likewise, start time information can help us distinguish between activities such as having dinner versus having lunch.

$$\Upsilon_t[i,j] = \max_{\substack{\theta_1 \in \Theta_i \\ \theta_2 \in \Theta_j}} \left(1 - \frac{|\Phi_{\theta_2} - \Phi_{\theta_1}|}{2\pi}\right) \tag{6}$$

Duration mapping is calculated as in Equation (7). Duration is also represented in the form of a mixture normal distribution with means $\Gamma = \langle \gamma_1..\gamma_r \rangle$.



(a) Breakfast and lunch.

(b) Breakfast, lunch and dinner.

Fig. 7: Example start Time distribution for "eating" activity.

$$\Upsilon_d[i,j] = \max_{\substack{\gamma_1 \in \Gamma_i \\ \gamma_2 \in \Gamma_j}} \left(1 - \frac{|\gamma_2 - \gamma_1|}{\max(\gamma_2, \gamma_1)}\right) \tag{7}$$

The regional and structural similarities are calculated as in Equations (8) and (9) using Jaccard similarity measure [Tan et al. 2005]. In Equation (8), $\mathcal{E}$ refers to the set of sensors for a pattern.

$$\Upsilon_S[i,j] = \frac{|\ \mathcal{E}_i \bigcap \mathcal{E}_j\ |}{|\ \mathcal{E}_i \bigcup \mathcal{E}_j\ |} \tag{8}$$

$$\Upsilon_{\mathcal{L}}[i,j] = \frac{|\ \mathcal{L}_i \bigcap \mathcal{L}_j\ |}{|\ \mathcal{L}_i \bigcup \mathcal{L}_j\ |} \tag{9}$$

3.3. Recognition

The activity recognition algorithm in our model is based on using a Hidden Markov Model (HMM) [Rabiner 1990]. The HMM is constructed automatically from the cluster centroids. An HMM is a statistical model of a dynamic system, which models the system using a finite set of hidden states and observable states. Each of the states is associated with a multidimensional probability distribution. Transitions between hidden states are governed by transition probabilities. Transitions from hidden states to observable states are declared as emission probabilities. In our model, each hidden state corresponds to a discovered activity, while the observable states correspond to the fired sensor events (see Figure 8).

We can specify an HMM using three probability distributions: the distribution over initial states $\Pi = \{\pi_k\}$, the state transition probability distribution $A = \{a_{kl}\}$, with $a_{kl} = p(y_t = l | y_{t-1} = k)$ representing the probability of transitioning from state k to state l; and the observation or emission distribution $B = \{b_{il}\}$, with $b_{il} = p(x_t = i | y_t = l)$ indicating the probability that the state l would generate observation $x_t = i$. We can find the most likely sequence of hidden states given the observation in Equation (10) and by using the Viterbi algorithm [Viterbi 1967].

$$argmax_{x_1 \ldots x_t} P(y_1, ..., y_t, y_{t+1} | x_{1:t+1}) \tag{10}$$

We compute the transition probabilities and the observation probabilities automatically from data during the data mining and clustering stages.

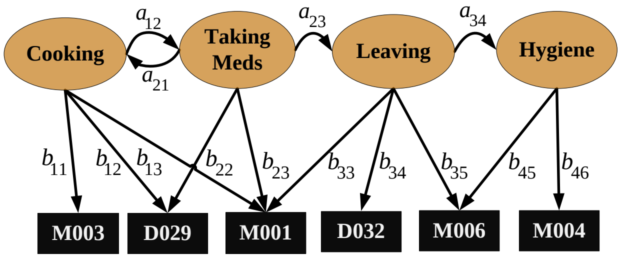
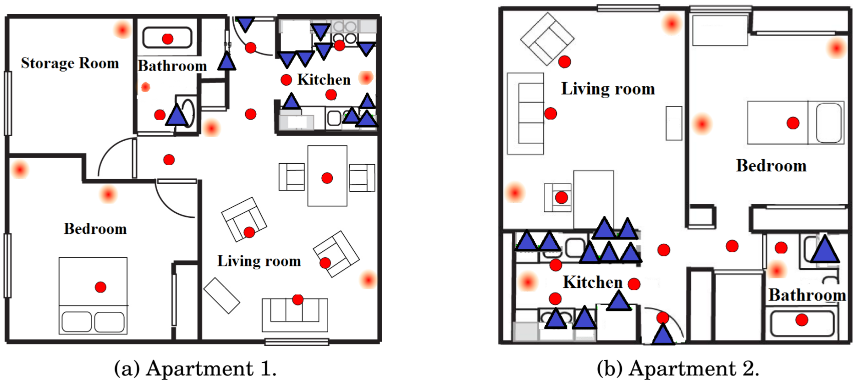


Fig. 8: A Hidden Markov Model. The ovals represent hidden states (i.e., activities) and the rectangles represent observable states (i.e. sensor activations). Here a_{ij} represents transition probability between activities and b_{kl} represents the emission probability of the observable state given current hidden state.

4. EXPERIMENTS

We evaluated the performance of our COM algorithm using the data that was collected from two different smart apartments. The layout of the apartments including sensor placement and location tags are shown in Figure 9. We will refer to apartments in Figures 9a and 9b as Apartments 1 and 2. The data was collected during an approximately three month period. Each apartment is equipped with motion sensors and magnetic door sensors which monitor the open/closed status of doors and cabinets.



(a) Apartment 1.

(b) Apartment 2.

Fig. 9: Sensor map and location tags for each apartment. On the map, circles show motion sensors while triangles show switch contact sensors. The hollow-shaped motion sensors are the area motion sensor.

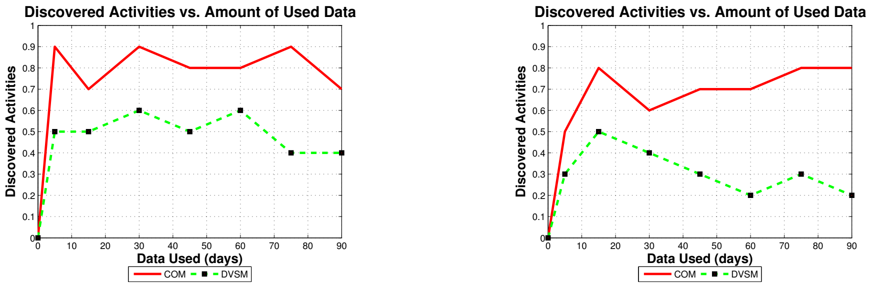
To be able to evaluate the results of our algorithms, each of the datasets was annotated with ADL activities of interest for the corresponding resident and apartment. Those activities are typical activities used in the evaluation of assisted living systems. A total of 10 activities were noted for each apartment. Those activities included bathing, bed-toilet transition, eating, leave/enter home, meal preparation(cooking), personal hygiene, sleeping in bed, sleeping not in bed (relaxing) and taking medicine. The first and second datasets include 3384 and 2602 annotated activity instances, respectively.

All the activities are performed in a natural setting, i.e. data is collected without any interruption in the daily life of residents. The resident in the first apartment is a

92 years old man, while the resident in the second apartment is a female older adult around 80 years old. Both residents live alone independently. All the sensor data is captured and stored in a SQL database, using a publish/subscribe protocol middleware. To maintain privacy we remove identifying information and encrypt collected data before it is transmitted over the network.

We ran our algorithm for each one of the apartments. One major improvement of our COM algorithm is to use multiple support thresholds for different regions of homes, as well as for different types of sensors. A single support threshold might not pose a problem in scripted experiments where all activities are performed with the same frequency (e.g. 20 times as in our DVSM experiments [Rashidi et al. 2011]). However, in real life this assumption results in discarding many patterns. To show how using a single threshold affects the accuracy of pattern discovery, we performed a number of experiments, once using the COM algorithm and once using DVSM.

The data mining step was able to discover a considerable number of pre-defined activities of interest. In apartment 1, it discovered 8 out of 10 activities, including bathing, leave/enter home, meal preparation(cooking), personal hygiene, sleeping in bed, sleeping not in bed (relaxing) and taking medicine. In apartment 2, it was able to discover 7 out of 10 activities including bathing, bed-toilet transition, eating, enter home, leave/enter home, meal preparation(cooking), personal hygiene, sleeping in bed, and sleeping not in bed (relaxing). Some of the patterns that have not been discovered are indeed quite difficult to spot and also in some cases less frequent. For example the housekeeping activity happens every 2-4 weeks and is not associated with any specific sensor. Also some of the similar patterns are merged together, as they use the same set of sensors, such as eating and relaxing activities. It should be noted that some of the activities are discovered multiple times in form of different patterns, as the activity might be performed in a different motion trajectory using different sensors. Figures 10a and 10b show the number of distinct discovered activities by both COM and DVSM algorithms in apartments 1 and 2. One can clearly see that COM is able to discover a higher number of distinct activities. These figures also show the amount of data for achieving a certain level of accuracy.



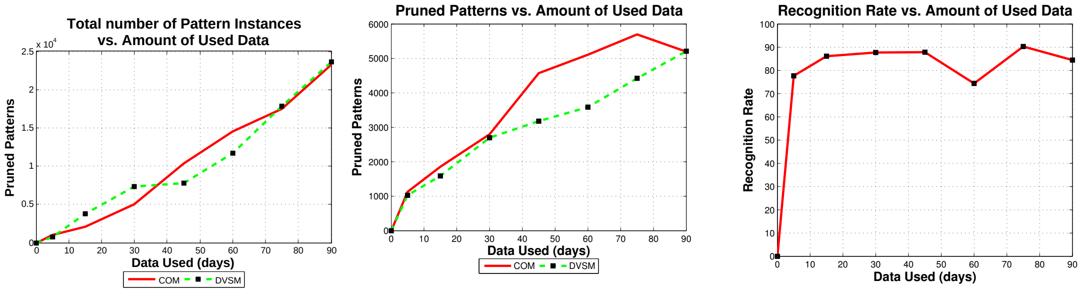
(a) Apartment 1.

(b) Apartment 2.

Fig. 10: Percentage of distinct activities discovered vs. the amount of data.

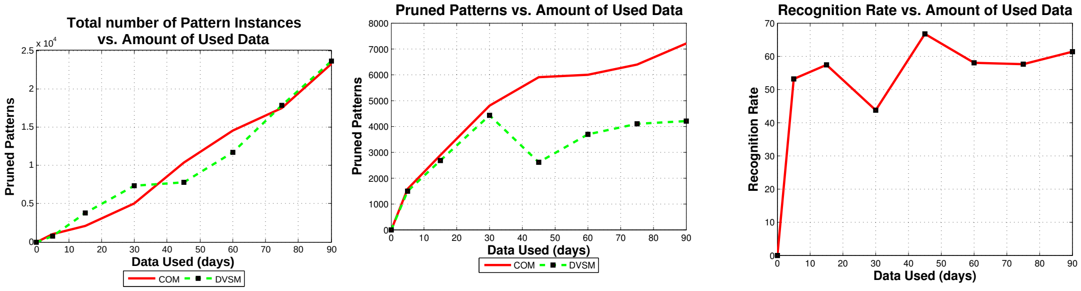
Using externally provided annotations to verify our results, we also computed the percentage of non-annotated discovered activities, i.e. the percentage of activities that actually have no annotation, but have been discovered by our algorithm. For apartment 1 the percentage of non-annotated activities with respect to the number of total distinct activities was 7.0% and for apartment 2 it was 2.0%. The low percentage of discovered activities that are not annotated shows that most of the discovered patterns are indeed well aligned with those patterns identified by the human annotator as interesting in the first place.

Figures 11b and 12b show the total number of discovered patterns *instances* for both the COM and DVSM algorithms, as well as the total number of pruned instances. It can be seen that though sometimes COM generates more pattern instances and in general more patterns, it also prunes more pattern instances due to its improved pruning capabilities, while still discovering more distinct patterns (activities).



(a) Total pattern instances. (b) Pruned pattern instances. (c) HMM Recognition.

Fig. 11: Results for apartment 1.



(a) Total pattern instances. (b) Pruned pattern instances. (c) HMM Recognition.

Fig. 12: Results for apartment 2.

In the next step, we clustered the discovered activities together using our agglomerative clustering method. The similarity threshold ζ of 0.75 was found to be a suitable value based on several runs of our experiments. By using the externally provided labels, we were also able to measure the purity of the clustered patterns with respect to the consistency of their variations using Equation (11). Equation (11) is known in literature as Jaccard similarity is a form of simple matching coefficient and is used to determine the similarity between clusters and actual classes. Here $|v_{11}|$ refers to the number of variations that have the same label as their general pattern, and $|v_{01}|$ and $|v_{10}|$ refer to the number of variations that have a different label other than their general pattern's label. We call this number the variation consistency.

Table II: Confusion matrix for apartment 1.

	Hg.	Lv.	Ck.	Rx.	Md.	Et	Hs.	Sp.	Bh.	Tl.
Hg.	**91.6**%	0.0%	0.0%	7.2%	0.0%	0.0%	0.0%	0.0%	0.9%	0.0%
Lv.	0.0%	**88.3**%	0.0%	10.5%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
Ck.	0.1%	0.0%	**97.8**%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Rx.	0.0%	1.8%	2.2%	**95.9**%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Md.	0.0%	0.2%	55.6%	0.5%	**43.5**%	0.0%	0.0%	0.0%	0.0%	0.0%
Et	0.0%	0.0%	1.3%	98.6%	0.0%	**0.0**%	0.0%	0.0%	0.0%	0.0%
Hs.	0.0%	0.0%	98.3%	1.6%	0.0%	0.0%	**0.0**%	0.0%	0.0%	0.0%
Sp.	0.3%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	**99.3**%	0.0%	0.0%
Bh.	26.4%	0.0%	0.0%	0.5%	0.0%	0.0%	0.0%	0.0%	**72.9**%	0.0%
Tl.	76.6%	0.0%	0.0%	19.8%	0.0%	0.0%	0.0%	2.9%	0.4%	**0.0%**

The activity names are abbreviated as Hg. = Hygiene, Lv. = Leave Home, Ck. = Cook, Rx. = Relax, Md. = Taking Medication, Et. = Eat, Hs. = Housekeeping, Sp. = Sleep, Bh. = Bathing, and Tl. = Bed to Toilet.

$$purity(P_i) = \frac{|v_{11}|}{|v_{11}| + |v_{01}| + |v_{10}|} \tag{11}$$

The variation consistency for clustered patterns in apartment 1 was 0.90 in our experiments, while for apartment 2 it was 0.76. By closely looking at the data and examining it, it was revealed that the patterns in the second dataset are much more irregular. Therefore most similar patterns are combined together, such as taking medication and meal preparation which usually happen at approximately the same time and the same location (in this case kitchen).

Next, using the discovered patterns an HMM was automatically constructed to track and recognize the activities. Figures 11c and 12c show the recognition results for the two apartments. Tables II and III show the confusion matrices for all the activities. It can be seen from those tables that due to higher regularity in apartment 1, it is easier to track and recognize activities. It should be noted that some similar activities might be mistaken with one another, such as relaxing and eating which happen at similar times and similar locations. In addition, some activities such as the housekeeping activity are not recognized because they were not discovered in the first place. As mentioned before, housekeeping happens quite rarely compared to other activities; and even if it happens to be in the data, because of its erratic nature and occurring all over the home, it is not very easy to discover. Therefore, in the activity recognition step, it is mistaken with other activities such as cooking. However, one still can observe that despite the fact that the real life activities can be somehow hectic and irregular, still our algorithm is able to track and monitor a considerable number of the patterns.

The output of our preliminary algorithms are designed to be in XML format. Though an excellent choice for data representation and portability, XML might not be the best choice for a natural user-friendly representation, especially in case of activity patterns with features such as temporal features and trajectory path. We have designed a preliminary visualizer to visualize the discovered activity patterns on the home map, as depicted in Figure 13a. It also displays important statistical information such as start time or duration. Users can go back and forth between patterns and their variations using navigation buttons. Also it's possible to see all the patterns or all the variations of a pattern at the same time. In this case, each pattern (or variation) will be visual-

Table III: Confusion matrix for apartment 2.

	Hg.	**Lv.**	**Ck.**	**Rx.**	**Md.**	**Et.**	**Hs.**	**Sp.**	**Bh.**	**Tl.**
Hg.	**83.9**%	0.0%	3.4%	11.8%	0.0%	0.0%	0.0%	0.0%	0.6%	0.0%
Lv.	0.0%	**93.2**%	6.0%	0.7%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Ck.	0.4%	0.6%	**94.8**%	4.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Rx.	0.1%	0.6%	0.2%	**98.9**%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Md.	0.9%	0.4%	95.7%	2.9%	**0.0**%	0.0%	0.0%	0.0%	0.0%	0.0%
Et.	1.3%	0.0%	4.6%	93.9%	0.0%	**0.0**%	0.0%	0.0%	0.0%	0.0%
Hs.	0.8%	0.8%	93.1%	5.1%	0.0%	0.0%	**0.0**%	0.0%	0.0%	0.0%
Sp.	39.4%	0.0%	42.5%	17.9%	0.0%	0.0%	0.0%	**0.0**%	0.0%	0.0%
Bh.	23.5%	0.1%	0.7%	2.3%	0.0%	0.0%	0.0%	0.0%	**73.2**%	0.0%
Tl.	83.0%	0.0%	3.2%	13.1%	0.0%	0.0%	0.0%	0.0%	0.5%	**0.0**%

The activity names are abbreviated as Hg. = Hygiene, Lv. = Leave Home, Ck. = Cook, Rx. = Relax, Md. = Taking Medication, Et. = Eat, Hs. = Housekeeping, Sp. = Sleep, Bh. = Bathing, and Tl. = Bed to Toilet.

ized using a unique color-code as depicted in "Map Guide". Our visualization tool also generates graphic charts which highlight certain features of the data, as depicted in Figure 13b. The charts include information such as activity occurrences at multiple temporal granularity and activity density in terms of color coded number of sensor events per hour. Using such a simple visualizer allows users not to deal with the sensor information in a textual format, which might be confusing and hard to understand. Rather it allows the users to see the patterns in a natural format and quickly diagnose relations between different variations of a pattern. Several discovered patterns and their variation are shown in Figure 14.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an automatic approach to activity discovery and monitoring for assisted living in a real world setting. Most current approaches use supervised methods for activity recognition. However, due to the required effort and time for annotating activity datasets, supervised methods do not scale up well in practice. Annotating activity data imposes a burden on annotators and residents and often introduces a source of error in the process. Our model is able to automatically discover activity patterns and their variations, even if the patterns exhibit discontinuity or if the patterns' frequencies exhibit difference across different regions in home.

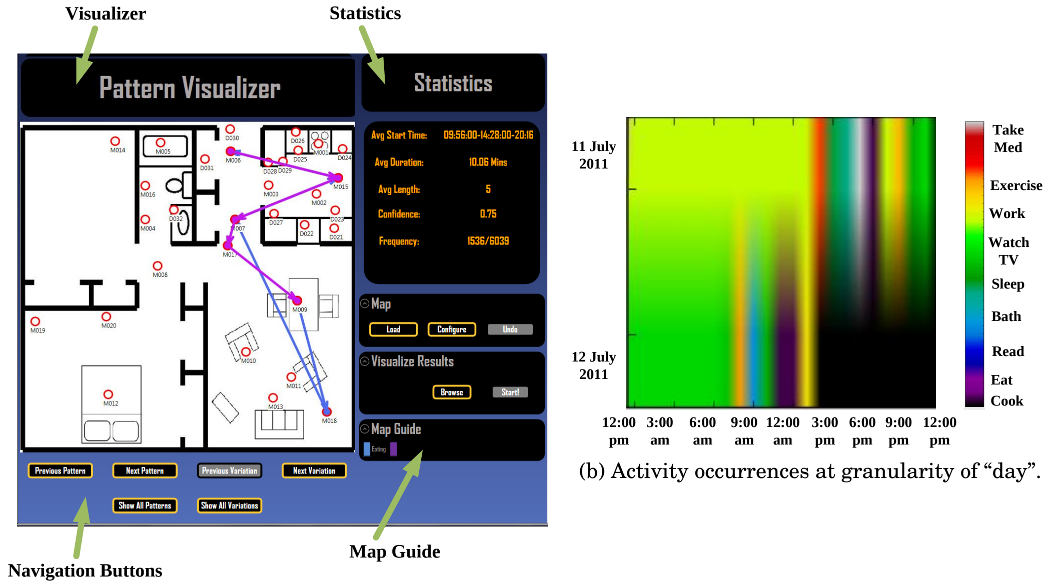
Our sequence mining algorithm has the potential to be applied to wide range of applications such as web click analysis or DNA sequence analysis. Ultimately, we hope that our model can be used in a fully functional system deployed in a real home. In the future, we plan to extend the system using an anomaly detection component to detect anomalies in observed data.

ACKNOWLEDGMENTS

The authors would like to thank Chao Chen and Prafulla Nath Dawadi for providing the visualization diagrams.

REFERENCES

Abowd, G. and Mynatt, E. 2004. *Smart Environments: Technology, Protocols, and Applications*. Wiley, Hoboken, NJ, USA, Chapter Designing for the human experience in smart environments., 153–174.



Avg Start Time: 09:56:00-14:28:00-20:16
Avg Duration: 10.06 Mins
Avg Length: 5
Confidence: 0.75
Frequency: 1536/6039

(a) The pattern visualizer.

(b) Activity occurrences at granularity of "day".

Fig. 13: Snapshots of the visualizer.

AGRAWAL, R. AND SRIKANT, R. 1995. Mining sequential patterns. In *International Conference on Data Engineering*. 3–14.

AOA. 2011. Aging statistics. http://www.aoa.gov.

BARGER, T., BROWN, D., AND ALWAN, M. 2005. Health-status monitoring through analysis of behavioral patterns. *IEEE Transactions on Systems, Man and Cybernetics, Part A 35,* 1, 22 – 27.

B.K. HENSEL, G. D. AND COURTNEY, K. 2006. Defining obtrusiveness in home telehealth technologies: A conceptual framework. *Journal of the American Medical Informatics Association 13*, 428–431.

BRDICZKA, O., MAISONNASSE, J., AND REIGNIER, P. 2005. Automatic detection of interaction groups. In *international conference on Multimodal interfaces*. 32–36.

CHAN, M., ESTÈVE, D., ESCRIBA, C., AND CAMPO, E. 2008. A review of smart homes-present state and future challenges. *Computer Methods and Programs in Biomedicine 91,* 1, 55–81.

CHIU, D.-Y., WU, Y.-H., AND CHEN, A. L. P. 2004. An efficient algorithm for mining frequent sequences by a new strategy without support counting. In *Proceedings of the 20th International Conference on Data Engineering*. ICDE '04. 375–382.

COOK, D. AND DAS, S. 2004. *Smart Environments: Technology, Protocols and Applications*. Wiley-Interscience.

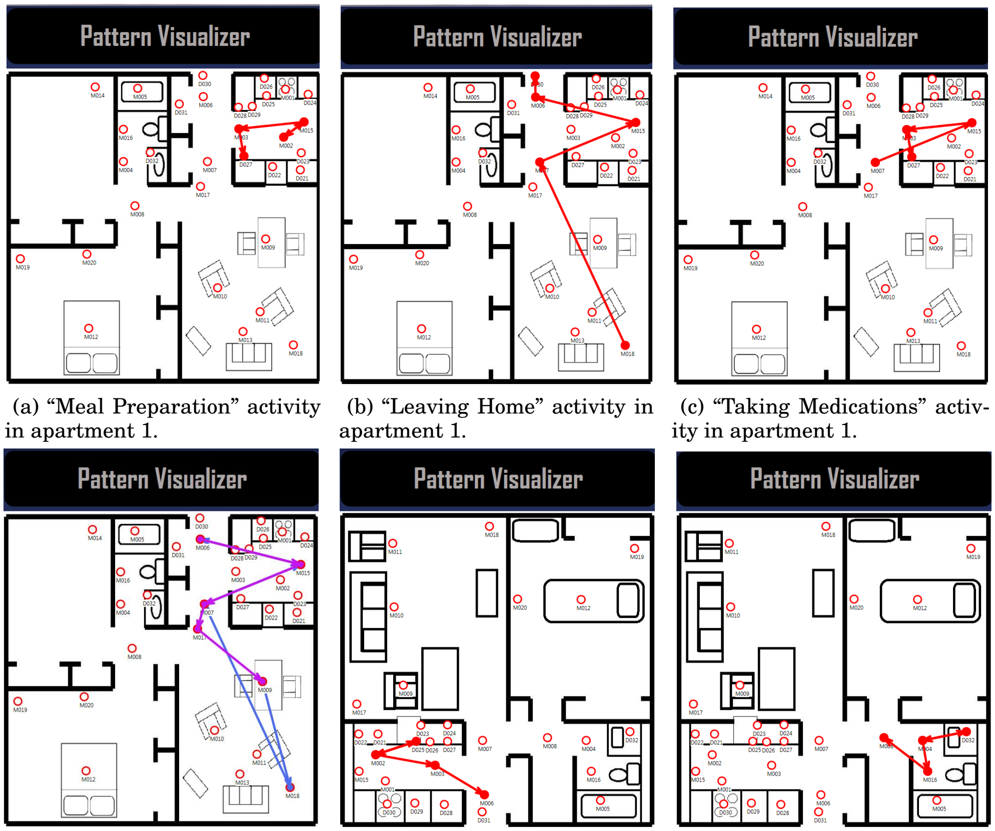
COOK, D. AND SCHMITTER-EDGECOMBE, M. 2009. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine 48,* 05, 480–485.

COOK, D., YOUNGBLOOD, M., HEIERMAN, E.O., I., GOPALRATNAM, K., RAO, S., LITVIN, A., AND KHAWAJA, F. 2003. Mavhome: an agent-based smart home. In *IEEE International Conference on Pervasive Computing and Communications*. 521–524.

DIMITROV, T., PAULI, J., AND NAROSKA, E. 2010. Unsupervised recognition of adls. In *Artificial Intelligence: Theories, Models and Applications*. Lecture Notes in Computer Science Series, vol. 6040. 71–80.

DOCTOR, F., HAGRAS, H., AND CALLAGHAN, V. 2005. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A 35,* 1, 55–65.

GOPALRATNAM, K. AND COOK, D. J. 2007. Online sequential prediction via incremental parsing: The active lezi algorithm. *IEEE Intelligent Systems 22,* 1, 52–58.



(a) “Meal Preparation” activity in apartment 1.

(b) “Leaving Home” activity in apartment 1.

(c) “Taking Medications” activity in apartment 1.

(d) Variations of “Leaving Home” activity in apartment 1.

(e) “Meal Preparation” activity in apartment 2.

(f) “Personal Hygiene” activity in apartment 2.

Fig. 14: Snapshots of the visualizer.

GU, T., WU, Z., TAO, X., PUNG, H., , AND LU, J. 2009. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *International Conference on Pervasive Computing and Communication*.

GUYON, I. AND ELISSEEFF, A. 2003. An introduction to variable and feature selection. *Machine Learning Research 3*, 1157–1182.

HAYES, T. L., PAVEL, M., LARIMER, N., TSAY, I. A., NUTT, J., AND ADAMI, A. G. 2007. Distributed healthcare: Simultaneous assessment of multiple individuals. *IEEE Pervasive Computing 6,* 1, 36–43.

HEBERT, L. E., BECKETT, L. A., SCHERR, P. A., AND EVANS, D. A. 2001. Annual incidence of alzheimer disease in the united states projected to the years 2000 through 2050. *Alzheimer Disease and Associated Disorders 15,* 4, 169–173.

HEIERMAN, III, E. O. AND COOK, D. J. 2003. Improving home automation by discovering regularly occurring device usage patterns. In *International Conference on Data Mining*. 537.

HELAL, S., MANN, W., EL-ZABADANI, H., KING, J., KADDOURA, Y., AND JANSEN, E. 2005. The gator tech smart house: A programmable pervasive space. *Computer 38*, 50–60.

HUYNH, T. AND SCHIELE, B. 2006. Unsupervised discovery of structure in activity data using multiple eigenspaces. Lecture Notes in Computer Science Series, vol. 3987. 151–167.

INOMATA, T., NAYA, F., KUWAHARA, N., HATTORI, F., AND KOGURE, K. 2009. Activity recognition from interactions with objects using dynamic bayesian network. In *International Workshop on Context-Awareness for Self-Managing Systems*. 39–42.

LIAO, L., FOX, D., AND KAUTZ, H. 2005. Location-based activity recognition using relational markov networks. In *International Joint Conference on Artificial Intelligence*. 773–778.

LIU, B., HSU, W., AND MA, Y. 1999. Mining association rules with multiple minimum supports. In *conference on Knowledge discovery and data mining*. 337–341.

LOO, K. K., TONG, I., KAO, B., AND CHEUNG, D. 2005. Online algorithms for mining inter-stream associations from large sensor networks. In *PAKDD*.

MASSEGLIA, F., CATHALA, F., AND PONCELET, P. 1998. The psp approach for mining sequential patterns. In *European Symposium on Principles of Data Mining and Knowledge Discovery*. 176–184.

MAURER, U., SMAILAGIC, A., SIEWIOREK, D. P., AND DEISHER, M. 2006. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks*. 113–116.

MCDOWELL, I. AND NEWELL, C. 1996. *Measuring Health: A Guide to Rating Scales and Questionnaires* 2nd Ed. Oxford University Press, New York.

MOZER, M. C., DODIER, R. H., ANDERSON, M., VIDMAR, L., III, R. F. C., AND MILLER, D. 1998. The neural network house: An overview. In *American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*. 110–114.

PALMES, P., PUNG, H. K., GU, T., XUE, W., AND CHEN, S. 2010. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive Mobile Computing 6*, 43–57.

PAPADIMITRIOU, S., BROCKWELL, A., AND FALOUTSOS, C. 2003. Adaptive, hands-off stream mining. In *conference on Very large data bases*. 560–571.

PEI, J., HAN, J., MORTAZAVI-ASL, B., PINTO, H., CHEN, Q., DAYAL, U., AND HSU, M. 2001. Prefixspan: Mining sequential patterns by prefix-projected growth. In *International Conference on Data Engineering*. 215–224.

PEI, J., HAN, J., MORTAZAVI-ASL, B., WANG, J., PINTO, H., CHEN, Q., DAYAL, U., AND HSU, M.-C. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transaction on Knowledge and Data Engineering 16*, 1424–1440.

PEI, J., HAN, J., AND WANG, W. 2007. Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems 28,* 2, 133–160.

PERKOWITZ, M., PHILIPOSE, M., FISHKIN, K., AND PATTERSON, D. J. 2004. Mining models of human activities from the web. In *international conference on World Wide Web*. 573–582.

PHILIPOSE, M., FISHKIN, K., PERKOWITZ, M., PATTERSON, D., FOX, D., KAUTZ, H., AND HAHNEL, D. 2004. Inferring activities from interactions with objects. *IEEE Pervasive Computing 3,* 4, 50–57.

PYLE, D. 2005. *Do Smart Adaptive Systems Exist?* Springer, Chapter Data Preparation and Preprocessing, 27–53.

RABINER, L. R. 1990. Readings in speech recognition. Morgan Kaufmann Publishers Inc., Chapter A tutorial on hidden Markov models and selected applications in speech recognition, 267–296.

RASHIDI, P. AND COOK, D. J. 2008. Adapting to resident preferences in smart environments. In *AAAI Workshop on Preference Handling*. 78–84.

RASHIDI, P. AND COOK, D. J. 2009. the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on Systems, Man, and Cybernetics journal, Part A 39,* 5, 949–959.

RASHIDI, P., COOK, D. J., HOLDER, L. B., AND SCHMITTER-EDGECOMBE, M. 2011. Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering 23*, 527–539.

REISBERG, B., FINKEL, S., OVERALL, J., SCHMIDT-GOLLAS, N., KANOWSKI, S., LEHFELD, H., HULLA, F., SCLAN, S. G., WILMS, H.-U., HEININGER, K., HINDMARCH, I., STEMMLER, M., POON, L., KLUGER, A., COOLER, C., BERGENER, M., HUGONOT-DIENER, L., ROBERT, P. H., AND ERZIGKEIT, H. 2001. The Alzheimer's disease activities of daily living international scale. *International Psychogeriatrics 13,* 02, 163–181.

RIALLE, V., OLLIVET, C., GUIGUI, C., AND HERVÉ, C. 2008. What do family caregivers of alzheimer's disease patients desire in smart home technologies? contrasted results of a wide survey. *Methods of Information in Medicine 47*, 63–9.

RISSANEN, J. 1978. Modeling by shortest data description. *Automatica 14*, 465–471.

SCHIELE, B. 2006. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *Workshop on Location and Context Awareness*. Springer.

STAUFFER, C. AND GRIMSON, W. 2000. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22,* 8, 747–757.

TAN, P.-N., STEINBACH, M., AND KUMAR, V. 2005. *Introduction to Data Mining*. Adison Wesley.

TAPIA, E. M., INTILLE, S. S., AND LARSON, K. 2004a. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, 158–175.

TAPIA, E. M., INTILLE, S. S., AND LARSON, K. 2004b. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing*, 158–175.

VAHDATPOUR, A., AMINI, N., AND SARRAFZADEH, M. 2009. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *international jont conference on Artifical intelligence*. 1261–1266.

VAIL, D. L., VELOSO, M. M., AND LAFFERTY, J. D. 2007. Conditional random fields for activity recognition. In *conference on Autonomous agents and multiagent systems*. ACM, New York, NY, USA, 1–8.

VAN KASTEREN, T., ENGLEBIENNE, G., AND KROSE, B. 2008. Recognizing activities in multiple contexts using transfer learning. In *AAAI AI in Eldercare Symposium*.

VG, W., O, O., M, C., AND LA, R.-M. 2008. Mild cognitive impairment and everyday function: Evidence of reduced speed in performing instrumental activities of daily living. *American Journal of Geriatric Psychiatry 16,* 5, 416–424.

VITERBI, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory 13,* 2, 260–269.

WANG, J. AND HAN, J. 2004. Bide: Efficient mining of frequent closed sequences. In *International Conference on Data Engineering*. 79.

WRAY, R. AND LAIRD, J. 2003. Variability in human modeling of military simulations. In *Conference of Behaviour Representation in Modeling and Simulation*. 160–169.

WREN, C. AND MUNGUIA-TAPIA, E. 2006. Toward scalable activity recognition for sensor networks. In *Workshop on Location and Context-Awareness*. 218–235.

YANG, Z., KITSUREGAWA, M., AND WANG, Y. 2006. Paid: Mining sequential patterns by passed item deduction in large databases. In *International Database Engineering and Applications Symposium*. 113–120.

YIPING, T., SHUNJING, J., ZHONGYUAN, Y., AND SISI, Y. 2006. Detection elder abnormal activities by using omni-directional vision sensor: Activity data collection and modeling. In *International Joint Conference SICE-ICASE*. 3850–3853.

ZAKI, M. J. 2001. Spade: an efficient algorithm for mining frequent sequences. *Machine Learning 42*, 31–60.

ZHAO, L., WANG, X., SUKTHANKAR, G., AND SUKTHANKAR, R. 2010. Motif discovery and feature selection for crf-based activity recognition. *Pattern Recognition, International Conference on 0*, 3826–3829.

Received November 2011; revised xx 2012; accepted xx 2012