USMART: An Unsupervised Semantic Mining Activity Recognition Technique

JUAN YE, GRAEME STEVENSON, and SIMON DOBSON, University of St. Andrews, UK

Recognising high-level human activities from low-level sensor data is a crucial driver for pervasive systems that wish to provide seamless and distraction-free support for users engaged in normal activities. Research in this area has grown alongside advances in sensing and communications, and experiments have yielded sensor traces coupled with ground truth annotations about the underlying environmental conditions and user actions. Traditional machine learning has had some success in recognising human activities; but the need for large volumes of annotated data and the danger of overfitting to specific conditions represent challenges in connection with the building of models applicable to a wide range of users, activities, and environments. We present USMART, a novel unsupervised technique that combines data- and knowledge-driven techniques. USMART uses a general ontology model to represent domain knowledge that can be reused across different environments and users, and we augment a range of learning techniques with ontological semantics to facilitate the unsupervised discovery of patterns in how each user performs daily activities. We evaluate our approach against four real-world third-party datasets featuring different user populations and sensor configurations, and we find that USMART achieves up to 97.5% accuracy in recognising daily activities.

Categories and Subject Descriptors: H.1.2 [**User/Machine Systems**]: Human Factors

General Terms: Design, Algorithms, Experimentation, Measurement, Performance

Additional Key Words and Phrases: Activity recognition, unsupervised learning, ontologies, clustering, string alignment, sequential pattern, segmentation, pervasive computing, smart home

16

ACM Reference Format:
Juan Ye, Graeme Stevenson, and Simon Dobson. 2014. USMART: An unsupervised semantic mining activity recognition technique. ACM Trans. Interact. Intell. Syst. 4, 4, Article 16 (November 2014), 27 pages.
DOI: http://dx.doi.org/10.1145/2662870

1. INTRODUCTION

Studying human behaviours in smart environments is a popular research area, and one that has demonstrated significant implications in human beneficial areas, such as ambient assisted living and elderly care [Petzold et al. 2013]. One of the core topics in this area is activity recognition—that is, recognising human daily activities from their interaction with sensorised objects and environments, which activity-aware applications use as triggers. For example, an ambient kitchen system could provide reminder or feedback services to assist users with their ongoing cooking tasks [Hooper et al.

This work has been supported by the EU FP7 project "SAPERE—Self-Aware Pervasive Service Ecosystems" under contract no. 256873.
The reviewing of this article was managed by the special issue associate editors Andreas Bulling, Ulf Blanke, Desney Tan, Jun Rekimoto, and Gregory Abowd.
Authors' address: J. Ye, G. Stevenson, and S. Dobson, School of Computer Science, University of St Andrews, North Haugh, St Andrews, Fife, UK, KY16 9SX; emails: {juan.ye, graeme.stevenson, simon.dobson}@st-andrews.ac.uk.

DOI: http://dx.doi.org/10.1145/2662870

2012]. The ability to correctly identify users' activities can help to foster and encourage interactions between users and a smart environment.

Pervasive sensing technology designs have progressed to small-size, easy-to-plug-in, long-lived sensors that implicitly collect ambient and user contextual data and thus recognise user activities. For example, object-attached sensors including RFID, pressure mats, binary-state sensors [Van Kasteren et al. 2008], state-change sensors [Tapia et al. 2004a], and motion sensors [Rashidi et al. 2011] have been designed and deployed on a wide range of objects and as well as in laboratory and real-world environments. These sensors exhibit the advantages of unobtrusive monitoring of user interaction with everyday objects in a privacy-preserving manner.

The user–object interaction renders a great opportunity of inferring user activities and further deriving their intentions. In most of the existing approaches [Krishnan and Cook 2014; Palmes et al. 2010; Patterson et al. 2005; Tapia et al. 2004a; Wang et al. 2007], activity representation takes the form of a probabilistic distribution over a set (or a sequence) of objects that are used in this activity—in other words, what objects are used in a certain activity and how significant each of these objects is contributing to identifying this activity [Wang et al. 2007]. The major challenge identified in these techniques is the difficulty in "either manually specifying or automatically identifying all the possible objects that may be used to perform one activity" [Tapia et al. 2006]. One user does not necessarily use the same set of objects each time the same activity is carried out [Brdiczka et al. 2007; Zhai and Bellotti 2005]. More than this, different users may perform the same activity differently. It is challenging to build a generic activity model that can be reused across users. The knowledge-driven techniques usually suffer from the overhead of knowledge engineering effort in specification building and the risk of overfitting the knowledge for individual users [Chen et al. 2012b]. Data-driven techniques need a large amount of high-quality training data to establish the model, which is time- and effort consuming to collect and annotate, especially in an unconstrained real-world setting.

To address this challenging issue, we propose the Unsupervised Semantic Mining Activity Recognition Technique (USMART), which can analyse the semantics underlying user–object interactions and use that semantics to discover meaningful fragments of interactions and to perform activity recognition on them. To the best of our knowledge, it is the first approach that is presented as a systematic way of deeply integrating knowledge- and data-driven techniques in that learning techniques including clustering, sequential mining, and string alignment are augmented with semantics. This technique exhibits the combined strengths of generality, unsupervised learning, and automatic segmentation of streaming sensor traces. Figure 1 presents the key processes of knowledge acquisition and activity recognition in USMART.

This article makes the following contributions. Firstly, we propose a general ontological model for representing human activities and domain concepts in a smart home environment, including objects, locations, and sensors. The ontological model is generic enough to be shared and reused in different home settings.

Secondly, we develop a semantics-drive online sensor data segmentation algorithm that is solely based on the temporal, spatial, and object semantics of sensor events. A prerequisite step in most current activity recognition approaches is to segment a sensor trace that is composed of a sequence of raw sensor events [Gu et al. 2011], such as partitioning a sensor trace using a 1-minute time window. However, this method is impractical in real-world scenarios, as the duration of activities typically varies anywhere from 30 seconds to more than 30 minutes. Such segmentation often gathers the sensor traces that are spread over the boundary of two activities into one segment, which adds extra noise to the inference process. Additionally, for a long-duration activity like cooking, sensor traces within a short time window are often insufficient to arrive at an accurate classification. Our approach partitions the sensor traces based on

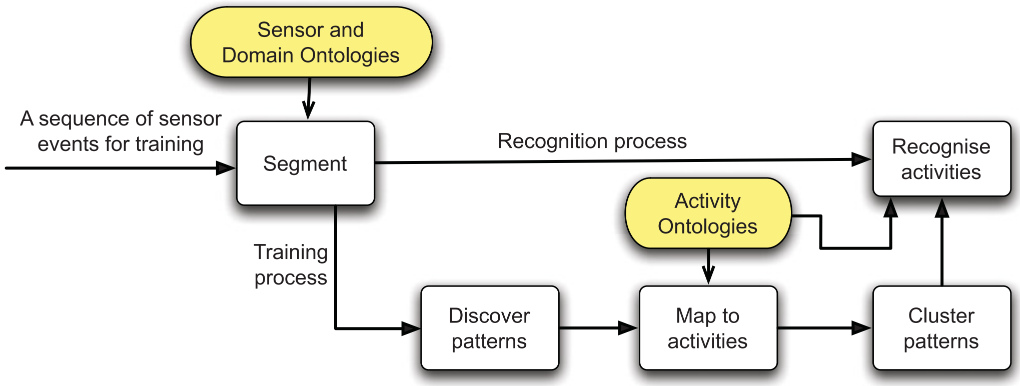


Fig. 1. An overview of USMART working processes.

the semantic similarity between two adjacent sensor events to dynamically segment real-time sensor streams.

Thirdly, we develop an unsupervised learning algorithm to discover sequential activity patterns of each individual user, which complements the general commonsense activity knowledge defined in the preceding ontologies. Finally, we evaluate our approach on four third-party datasets collected in real home settings for a reasonably long period where residents perform their usual daily routine without being given an explicit instruction on how to perform the activity. These unconstrained datasets are closer to practical everyday usage, in contrast to the other presegmented and well-annotated datasets collected in an instrumented laboratory setting where human subjects are asked to perform a set of scripted activities one at a time [Bao and Intille 2004; Singla and Cook 2010]. The goal of this evaluation is to study the generality and effectiveness of our approach on such more realistic datasets.

The preceding contributions can have a significant implication in facilitating designing interactive systems. Compared to pure data-driven techniques, USMART will be better suited for characteristic, event-driven sensor features that are often acquired in interactive systems, since it facilitates expressing and utilising domain knowledge on rich relationships between the features. The segmentation capability can lead to a smoother interaction design by not only identifying a user's current action but also by detecting transitions between the user's tasks and diversion of attention. Such process-like activity recognition will potentially result in more proactive service provision, which helps to strengthen interactive relationships between users and environments.

The rest of the article is organised as follows. Section 2 reviews the recent techniques in activity recognition and compares our technique with the techniques based on user–object interaction technologies. Section 3 presents a general ontological model in a smart home environment, based on which we introduce a technique for evaluating semantic similarity between sensor events in Section 4. Following this, we also discuss the mechanisms for segmenting sensor traces. Section 5 presents how to discover patterns of sequential sensor events when a user performs a certain activity, how to cluster the patterns into a set of concise patterns for each activity, and how to recognise activities from these patterns. The technique is evaluated using data from four real-world smart home environments in Section 6. The goal of the evaluation is to demonstrate the generality of the ontological model and present the performance of our technique in segmenting sensor traces and recognising activities. We discuss the utility of our approach with future research directions in Section 7 and conclude the article in Section 8.

2. RELATED WORK

This section reviews the state of the art in activity recognition and sensor segmentation. Activity recognition techniques have been studied extensively for decades in the areas of computer vision and pervasive computing. They can be roughly classified into two groups: knowledge- and data-driven approaches [Chen et al. 2012a; Ye et al. 2012]. In knowledge-driven approaches, ontologies are one of the most popular [Chen et al. 2012b; Gu et al. 2004; Riboni and Bettini 2009]. Chen et al. [2012b] present an ontological model to represent smart home activities and relevant context. The approach is motivated by the observations that users perform activities of daily living (ADLs) in certain routines that can be described in commonsense knowledge via relationships between the environment, domain contexts, and activities. Such knowledge is valuable in creating ADL models, avoiding the need of large-scale dataset collection and training. Our approach shares the same motivation; however, the difference is that we intend to use the more certain and concise knowledge rather than give a complete specification for each activity, which will be explored in learning techniques. Thus, we can reduce the amount of knowledge engineering effort as well as expert bias. Okeyo et al. [2011] present an interesting approach of evolving an activity model by discovering the new activity patterns from logs of activity data. Our approach discovers patterns based on the semantic matching between sensor events and each activity's profile, and it does not require the labelled activity log.

Riboni and Bettini [2009] build an ontology model for activities and use it to validate the results inferred from the statistical techniques. For example, if the sensors report the user's current location as "LivingRoom" and a statistical technique infers the current possible activities as {(BrushTeeth 0.6), (Reading 0.5)}, then the system will derive the activity "Reading" as the inference result with the consideration of the spatial constraint on these activities. Even though this work shares the same research direction with us in terms of combining knowledge and learning techniques, our approach is towards a deeper and more systematic integration of semantics within learning techniques.

In data-driven approaches, machine learning [Kasteren et al. 2011; Patterson et al. 2005] and data mining [Gu et al. 2009; Rashidi et al. 2011] techniques have been explored. Hidden Markov models (HMMs) [Bui et al. 2002; Fine et al. 1998; Kasteren et al. 2011] and dynamic Bayesian networks (DBNs) [Patterson et al. 2005; Wang et al. 2007] are some of the most widely used, which have the strength in encoding the probabilistic sequence of sensor events. An HMM usually is composed of a finite set of hidden states (e.g., activities) and observations (e.g., sensor events) [Kasteren et al. 2011]. To recognise complicated activities, researchers have developed derivative HMMs such as abstract [Bui et al. 2002] and hierarchical HMMs [Fine et al. 1998].

Patterson et al. [2005] explore extending object interaction–based activity recognition in a realistic setting. The authors aim towards recognising finer-grained activities as well as interleaved and interrupted activities. They study the similarity between objects, based on the lexical hierarchy deduced from an Internet shopping site. The statistical relations between objects and activities are learned in an HMM and a DBN. Our approach also considers the similarity between objects, and we use the WordNet [Miller 1995], which is considered to be more standard and has been widely used. For example, Tapia et al. [2006] uses WordNet to measure the functional similarities between objects, with which to adjust the probabilistic model. Within the model, the probabilities on two similar objects will be smoothed into more similar values. The motivation of their work is that "it is difficult to either manually or automatically identify all the possible objects that may be used to perform an activity, or to accurately calculate the probability with which they will be used." We share the same motivation and use sequential mining techniques to capture the patterns.

Rashidi et al. [2011] introduce an unsupervised method of discovering and tracking activities in a smart environment. They introduce a new mining method to discover patterns of activities and to group the patterns into activity definitions. A boosted version of an HMM is created to represent the activities and their variations, and to recognise real-time activities. This approach focuses on discovering the patterns while neither presents how to map the discovered patterns to the activities nor evaluate the performance in recognising activities from these patterns. Different from this approach, we use the semantics to map the sensor sequences to activities and provide a complete approach in recognising activities from the discovered patterns.

One major limitation for most data-driven approaches is the need for a large amount of labelled training data to establish models and estimate parameters; this usually requires a lot of human effort to label the collected sensor data. To reduce the reliance on training data, some researchers have applied Web mining techniques to extract the commonsense knowledge between activities and objects via mining online documents—that is, what objects are used to perform a daily activity and how significant each object is contributed to identifying this activity [Palmes et al. 2010; Zheng et al. 2009]. The success of the use of the relations between objects and activities motivates our approach to further explore semantic relations in object sensor events, which forms the foundation of USMART.

Zheng et al. [2009] propose an algorithm for cross-domain activity recognition that transfers the labelled data from a source domain to a target domain so that the activity model in the source domain can help to complete the similar activity model in the target domain. The similarity is measured not only on the objects involved in the activities but also on their underlying physical actions. One example in Zheng et al. [2009] is that the activity "Washing-laundry" is similar to "Hand-washing dishes" on the action of "Hand washing." They use the Web search and apply the information retrieval techniques to build the similarity function that produces different probabilistic weights of actions and objects on activities of interest. These weights will be further used to train a multiclass weighted support vector machine to support activity recognition.

Similarly, Gu et al. [2010] and Palmes et al. [2010] mine the Web to extract the most relevant objects involved in each activity. The object's relevance weight is calculated using normalised *term frequency–inverse document frequency* scores, which is a common weighting scheme in information retrieval communities to determine the relative degree of importance of a term to a document in a corpus. Using the relevance weight, an unsupervised algorithm is developed to segment the sensor trace and recognise the corresponding activities. Our approach also supports segmentation of sensor traces while only being based on the semantics between sensor events, making it independent of activities of interest and thus more general.

In terms of segmentation, sliding window protocols are a popular technique. Static sliding windows are used to retrieve sensor features [Bao and Intille 2004; Huỳnh et al. 2007]; however, this approach can truncate or overlap sensor events for an activity instant. Dynamic sliding window methods enable varied sizes of sliding windows based on different features, such as activity duration [Tapia et al. 2004b; McKeever et al. 2010; Okeyo et al. 2014], change of the sensor state [Laguna et al. 2011], or change of the location context of consecutive sensor data [Hong and Nugent 2009]. Krishnan and Cook [2014] explore both static and dynamic sliding window, with the incorporation of the time decay and mutual information based weighting of sensor events within a window (e.g., the change of two sensors occurring consecutively in the entire sensor stream). Rashidi and Cook [2010] extend the tilted time window approach to discover activity sequential patterns over time—that is, using temporally parameterised support counts to find frequent patterns over streaming sensor data. Compared to these segmentation techniques, our approach focuses on evaluating semantics between adjacent

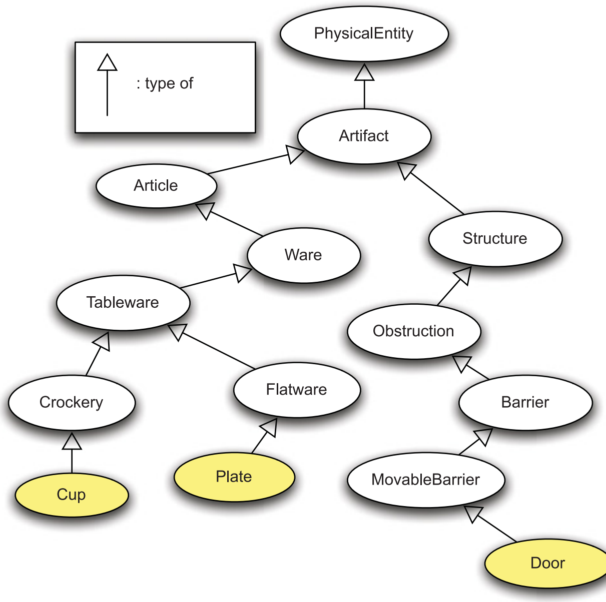


Fig. 2. A hierarchy of objects from WordNet. (Note: For brevity, we omit intermediate terms.)

sensor events in a continuous sensor trace. Different from the segmentation-focused techniques such as Krishnan and Cook [2014] and Okeyo et al. [2014], our segmentation method does not aim to detect exact activity boundaries but to cluster consecutive sensor events that map to a part or the whole of one activity instance.

3. ONTOLOGICAL MODEL

Central to our technique is a general ontological model that consists of four components: `Object`, `Location`, `Sensor`, and `Activity`. Among them, the object ontologies (OOs) and location ontologies (LOs) are general (also called *domain ontologies* in this article) in that they are defined independently of an application domain or a particular environment. The sensor ontologies (SOs) and activity ontologies (AOs) that are built using both ontologies represent concepts specific to environments and activities of interest. Founded on the domain ontology, the activity instances can be related to the sensor instances, which forms the basis for activity recognition.

3.1. Domain Ontology

The OO describes the `type-of` relationships between household objects, as shown in Figure 2. For example, a *cup* is a type of *crockery*, labelled as $Cup \sqsubseteq Crockery$. To make the OO as general as possible, we extract it from WordNet [Miller 1995], which is a hierarchically lexical system of words.

The LO describes `type-of` relationships between locations in terms of their functions rather than the conventionally spatial containment relationship as a usual way. For example, a *bedroom* is a type of *sleeping area*, labelled as $Bedroom \sqsubseteq SleepingArea$. There are two reasons for this design choice: (1) the containment relationship will be varied to different spatial layouts in different home settings, and (2) users perform activities in function-relevant locations. For example, a user is sleeping in a bedroom, and it does not matter whether it is a master bedroom or a guest bedroom, or if it is downstairs or upstairs. Based on this principle, we present the LO in Figure 3, which

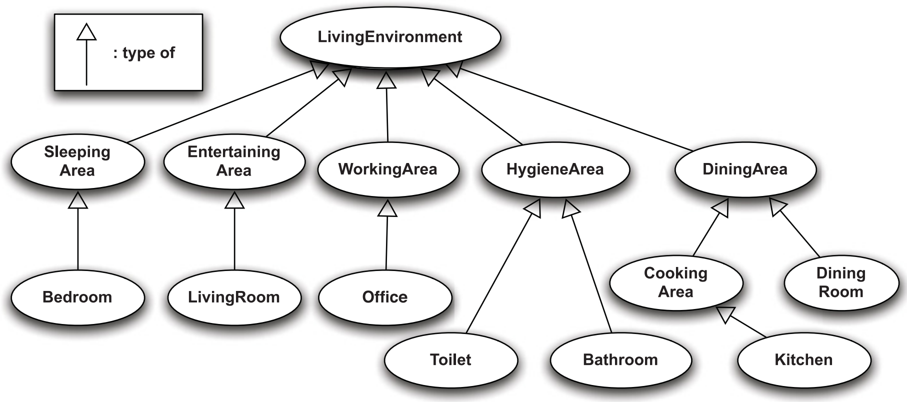


Fig. 3. A hierarchy of location concepts in a home setting.

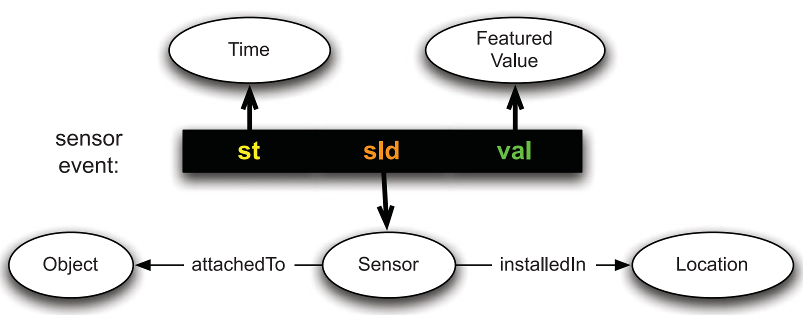


Fig. 4. Semantics of a sensor event.

represents the hierarchy of rooms common to most home settings. In Section 6, we project four home settings, including three apartments and one two-story house, onto this location hierarchy, which shows that the LO is general enough to accommodate all four. The LO is extensible with new rooms and new `type-of` relations. For example, a living room can be considered as a sleeping area if the user usually takes a nap on the couch therein.

3.2. Sensor Ontology

The SO represents sensors and sensor events. An object sensor event is usually represented in a tuple $se = (st, sId, val)$ (in Figure 4), where st is the time when the object is accessed by the user, sId is the id of the sensor that is attached to the object, and val is the reported value. A concrete example is `(25-Feb-2008 09:36:43, 5, 1)`, indicating that the sensor whose id is 5 reports a value of 1 at the time of 09:36:43 on 25th Feb 2008.

The sensor id refers to a sensor that is linked to an object installed in a certain location. The sensor value is optional in most types of object sensors, unless it represents an object state (e.g., open or closed, on or off). Using the sensor id and value, we can interpret the preceding sensor example as "the door of the toilet is open."

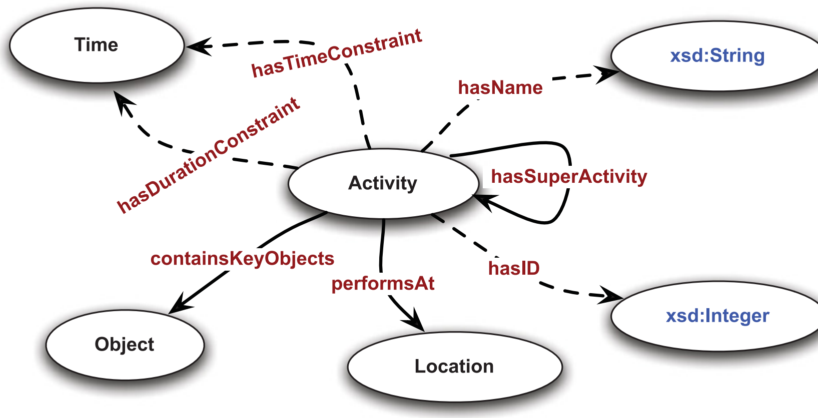


Fig. 5. The activity model. The dashed arrow represents data properties, and the filled arrow represents object properties.

3.3. Activity Ontology

The AO represents the structural properties of activities in Figure 5. Each activity instance is described with its *id*, *name*, and *super* activity. Besides, we constrain each activity with *time*, *object*, and *location* conditions.

Generally, conditions can be classified as *sufficient* or *necessary*. A sufficient condition is the most common type that has been used in knowledge-driven activity recognition techniques. It means that an activity is regarded as occurring if its sufficient condition is satisfied. An example rule for the activity “sleep” is taken from the work of Gu et al. [2005]:

$$\begin{aligned} &person\_in\_bedroom \wedge posture\_is\_liedown \wedge \\ &light\_in\_bedroom\_low \wedge door\_in\_bedroom\_closed \\ &\Rightarrow user\_is\_sleeping \end{aligned}$$

However, sufficient conditions are usually prone to expert bias and specific to particular environment or users. Without careful design, sufficient conditions do not always hold. In the preceding example, a user could sleep while leaving the bedroom door open or leaving the light on.

In contrast, a necessary condition restricts what must hold when a user is performing the activity. It means that if an activity is occurring, then its necessary condition must hold. Alternatively, if the condition does not hold, then it is impossible for the user to be performing the activity. In our example, the necessary condition on “Sleep” could be that a user is lying down, which implies that if the user is not lying down, he is definitely not sleeping. Therefore, compared to the sufficient condition, the necessary condition is easier to specify and often more certain in that we only consider the commonsense knowledge that apply to most users.

In the following, we show how to specify necessary conditions on activities. We specify two types of *time* conditions: *occurring* time, when the activity usually occurs, and *duration*, how long this activity typically lasts. For example in Table I, we place the constraint that the activity “Prepare breakfast” should occur in the morning (6am to 12am), and that the activity “Take shower” should last more than 5 minutes. The *object* and *location* conditions specify what object the user must access to perform this activity and where this activity must occur. Note that the necessary conditions that we specify here should be general knowledge in the sense that most people will execute the activity

Table I. Examples of Necessary Conditions Defined on Activities

Activity	Time	Duration	Key Objects	Location
Prepare breakfast	$[6h, 12h]$	$[2min, 60min]$	KitchenUtensil, Tableware	Kitchen
Prepare dinner	$[16h, 22h]$	$[5min, \infty]$	KitchenUtensil, Tableware	Kitchen
Get drink	—	—	Crockery	House
Take shower	—	$[5min, \infty]$	Bathtub, ShowerRoomDoor	Bathroom
Use toilet	—	—	ToiletFlush	Toilet
Leave house	—	—	FrontDoor	Entrance
Sleep	—	—	Bed	Bedroom

in this way. The sufficient condition on how an individual user actually performs an activity, including the other possible objects with which a user might interact, and the sequences of a user interacting with these objects, is left to the learning technique to discover, which will be described in Section 5.

The object and location conditions are represented as a collection of concepts in the domain ontologies OO and LO introduced in Section 3.1. Thanks to the existence of the OO and LO hierarchies, we can specify them using *more general* concepts rather than list each specific concept.

To enable the unsupervised learning of sufficient conditions for each activity, we need to map the activities to the sensors based on their corresponding objects and locations so that we can recognise activities from the input sensor events. In our approach, this mapping is executed automatically via the ontological reasoning, the principle of which is described in Definition 1—that is, if the sensor's corresponding object is a more specific concept of any required object and the sensor's installed location is a more specific concept of any required location, then the sensor is considered as a *key* sensor to the activity. Using this process, each activity will have a collection of key sensors; this indicates that the activity can be inferred from a sequence of sensor traces only if any of the key sensors are observed in the sequence.

Definition 1. For each activity in AO, its object and location constraint is represented in a collection of objects C_O and locations C_L. A sensor s is *mapped* to an activity if there exists an object $o \in C_O$ and a location $l \in C_L$ such that $O(s) \sqsubseteq o$ and $L(s) \sqsubseteq l$, where $O(s)$ returns the object that the sensor s is attached to, and $L(s)$ returns the location where s is installed.

As we assume that the AO should be general and is defined independently from the sensors deployed in a particular environment, so it is possible that there exists a certain activity to which no sensors can be mapped. In this situation, we will iteratively substitute all of its required objects (or locations) for their super objects (or locations) and then repeat the preceding process until we find the sensors for this activity.

4. EXPLORING SEMANTIC SIMILARITY OF SENSOR EVENTS

Based on the semantics defined on a sensor event, this section will explore the semantic similarity between sensor events—that is, a measure to evaluate how similar one sensor event is to another based on their temporal, spatial, and object dimensions described in the preceding section. The existence of such semantic similarity forms the foundation of USMART, with which we can segment sensor traces and thus discover sensor sequential patterns of activities.

Definition 2. Let se_i and se_j be two sensor events. The *semantic similarity* between them is defined as

$$sim(se_i, se_j) = (sim_T(st_i, st_j), sim_S(sId_i, sId_j)), \quad (1)$$

$$sim_S(sId_i, sId_j) = (sim_C(O(sId_i), O(sId_j)) + sim_C(L(sId_i), L(sId_j)))/2, \tag{2}$$

where

—st_i (st_j), sId_i (sId_j), $O(sId_i)$ ($O(sId_j)$), and $L(sId_i)$ ($L(sId_j)$) are the timestamps, sensor ids, objects that the sensors are attached to, and the locations where the sensors are installed, respectively.
—sim_T is the time similarity function that calculates the time distance between the sensor events.
—sim_S is the sensor similarity function that mainly depends on the conceptual similarity function sim_C between their objects and locations.

The conceptual similarity function is the algorithm proposed by Wu and Palmer [1994]. The idea is to find the least common subsumer (LCS) of the two input concepts and compute the path length from the LCS up to the root node. The LCS is the most specific concept that these two concepts share as an ancestor.

Different similarity measures may be employed. For example, the Leacock Chodorow matcher is based on counting the number of links between two concepts in the hierarchy [Leacock et al. 1998]. The principle of this method is similar to what we use here; however, Wu's similarity value [Wu and Palmer 1994] is more intuitive to understand. Some works use the information content of the concepts by exploiting the frequency of concept occurrences in a given text corpus, such as the Resnik matcher [Resnik 1995] and the Jiang Conrath matchers [Jiang and Conrath 1997]. Compared with these, Wu's method only needs a hierarchy of conceptual terms, which can be easily accessed through a standard lexical system like WordNet.

Definition 3. Let c_i and c_j be two concepts organised in one hierarchy. The *conceptual similarity* measure between them is calculated as

$$sim_C(c_i, c_j) = \frac{2 \times N_k}{N_i + N_j + 2 \times N_k}, \tag{3}$$

where N_i (N_j) is the path length between c_i (c_j) and the LCS node of c_i and c_j, and N_k is the path length between the LCS and the root.

When c_i is equal to c_j, their LCS node is itself and the similarity is 1.0. When c_i is semantically far from c_j, their LCS node can be close to the root in the hierarchy, which leaves N_i and N_j large and N_k small, so the similarity is close to 0. Therefore, the larger the similarity measure, the closer the two concepts. For example, in Figure 2, the conceptual similarity between *Cup* and *Plate* is $(2*4)/(2+2+2*4) = 0.67$, whereas the similarity between *Cup* and *Door* is $(2*1)/(5+5+2*1) = 0.17$.

Since the absolute timestamps on sensor events can be represented in numeric values, its similarity function is calculated as

$$sim_T(st_i, st_j) = max\left(0, 1 - \frac{|st_j - st_i|}{T_{max}}\right), \tag{4}$$

where T_{max} is the maximum range of the time under consideration. For example, if we consider daily activities, we set the T_{max} to be 24 hours (equally 86,400 seconds).

Example 1. In this example, we calculate the semantic similarity between the sensor events se_1, se_2, and se_3 in Figure 6.

—se_1 and se_2: Their time similarity is $1 - \frac{3}{86400} = 1$ using Equation (4); their sensor similarity is $(0.67 + 1)/2 = 0.83$ using Definition 2.

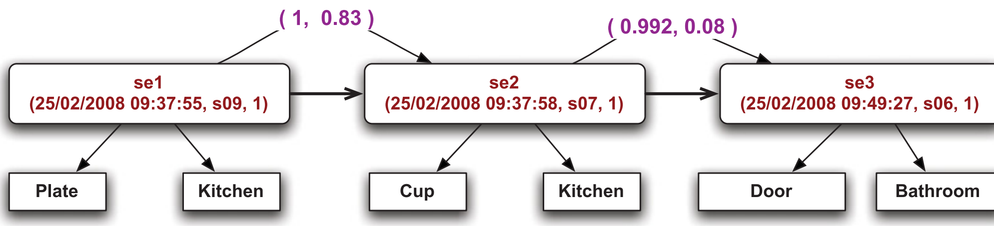


Fig. 6. Calculating semantic similarities on a sequence of sensor events.

—se_2 and se_3: Similarly, their time similarity is $1 - \frac{689}{86400} = 0.992$ and their sensor similarity is $(0.17 + 0)/2 = 0.08$ where the location similarity between these two sensors is 0.

In the preceding example, based on the similarity measures between these three sensor events, we could intuitively derive that the sensor events se_1 and se_2 should be grouped together, for they are more likely to correspond to the same activity, and se_3 should be partitioned from the former two.

Here we introduce Algorithm 1 on *online sensor trace segmentation*—that is, dividing a sequence of raw sensor events into segmentations, each of which is composed of sensor events that will be mapped to one activity. Given two adjacent sensor events in a continuous sensor sequence, if both their time and sensor similarity are above the given thresholds, meaning that these two are close enough, then they are joined together; otherwise, we dissect them. If the input sensor sequence contains simultaneous sensor events se_i and se_j in the input sequence, they can be placed in the sequence as either $\{\ldots, se_i, se_j, \ldots\}$ or $\{\ldots, se_j, se_i, \ldots\}$. The result is an m-sized list of segmented sequences: $L_S = \{\langle se_1, se_2, \ldots, se_{k^1}\rangle, \langle se_{k^1+1}, \ldots, se_{k^2}\rangle, \ldots, \langle se_{k^{m-1}+1}, \ldots, se_n\rangle\}$, where $k^i (1 \leq i \leq m-1)$ is the last index in the ith segmentation.

ALGORITHM 1: Online sensor trace segmentation

Data: $\langle se_1, se_2, \ldots, se_n\rangle$: a sequence of observed sensor events
(θ_T, θ_S): the time and sensor threshold pair
Result: L_S: a list of segmented sequences

```
initialise(fragment);
for i = 1 to n do
    if fragment.isNotEmpty then
        (sim_T, sim_S) = sim(se_{i-1}, se_i);
        if ((sim_T <= θ_T) || (sim_S <= θ_S)) then
            L_S.add(fragment);
            initialise(fragment);
    fragment.add(se_i.sId);
if fragment.isNotEmpty then
    L_S.add(fragment);
return L_S;
```

5. SEQUENTIAL PATTERN MINING

Having segmenting sensor traces, we describe how to identify frequent, repeated, meaningful sequences of sensor events for each activity. As illustrated in Figure 1, on the resultant segments of sequences we label the sequential patterns with activities based on the sensor-activity mapping in Definition 1. Then we apply the k-means clustering technique to cluster the sequences of each activity into k sequential patterns that are intended to be the most frequent and representative patterns for this activity. The activity recognition is then performed using these sequential patterns.

5.1. Discovering Patterns

To perform the sequential pattern mining, we preprocess the acquired segmentations L_S in Algorithm 1 into $L'_S = \{\langle sId_1, sId_2, \ldots, sId_{k^1}\rangle,\ \langle sId_{k^1+1}, \ldots, sId_{k^2}\rangle, \ldots, \langle sId_{k^{m-1}+1}, \ldots, sId_n\rangle\}$, which retains only sensor ids that are observed in the sequence. We use a simple algorithm to discover sequential patterns, namely, $r(s)$ records the occurrence ratio for each unique sequence s in L'_S. At this stage, we aim to capture all possible patterns and map them to the activities. We do not use the classic most-frequent sequential pattern mining techniques [Han et al. 2011], which might drop infrequent patterns that could be significant to infrequently occurring activities.

After we extract all of the unique sequences, we map them to each activity. The principle is that if the sequence contains any of the key sensors of an activity, then this sequence is mapped to the activity. In the end, each activity holds a collection of sequences $A : \{(sq_1, r_1), \ldots, (sq_n, r_n)\}$, where sq_i $(1 \leq i \leq n)$ is a sequence of sensor ids and r_i is the occurrence ratio of the sequence. It is possible for a sequence to be mapped to more than one activity. For example, the activity "prepare breakfast" shares sequences with the activities "get drink" and "prepare dinner."

5.2. Clustering Patterns

After mapping, we apply the k-means clustering algorithm to each activity's sequences. The clustering is based on the semantic similarity of two sequences. Here we employ the string alignment algorithm based on Levenshtein distance, which allows us to replace the cost calculation with the semantic similarity.

5.2.1. Semantics-Enhanced Edit Distance. Levenshtein distance is a common method to quantitatively measure the difference between two string sequences in information theory. The Levenshtein distance computes the cost of converting one string to another string using *deletion*, *insertion*, and *substitution* operations. For example, given two strings "ADF" and "ACEGDG," if we set the costs for each operation to be 1, then their Levenshtein distance is 4—that is, we need four operations of deletion, insertion, and substitution to convert ADF to ACEGDG.

Consider now that each character in a string represents a sensor event. Given the following three sequences,[1] $sq_1 = \langle cup, plate, fridge\rangle$, $sq_2 = \langle pan, freezer\rangle$, and $sq_3 = \langle bathroom\_door\rangle$, the edit distance between sq_1 and sq_2 and the distance between sq_1 and sq_3 are the same (i.e., 3), implying that the sequences sq_2 and sq_3 are equally different from sq_1, which is obviously not true. The reason for this is that the symbols under these sequences are not string symbols but concepts with associated semantic meanings. To address this, we recalculate their distances using Definition 4 that extends the classic formula with the semantic similarity between the sensor events.

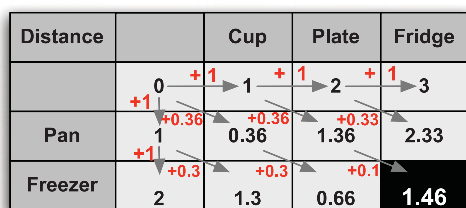
Definition 4. Given two sensor sequences $seq_1 = \langle se^1_1, se^1_2, \ldots, se^1_{n_1}\rangle$ and $seq_2 = \langle se^2_1, se^2_2, \ldots, se^2_{n_2}\rangle$, the *semantics-enhanced edit distance* on the ith and jth sensor events on these two sequences is

$$dist[i,j] = \begin{cases} dist[i-1, j-1], & \text{if } se^1_i = se^2_j \\ min(dist[i-1,j]+1, dist[i,j-1]+1, & \\ \quad dist[i-1,j-1] + (1 - sim(se^1_i, se^2_j)), & \text{otherwise.} \end{cases}$$

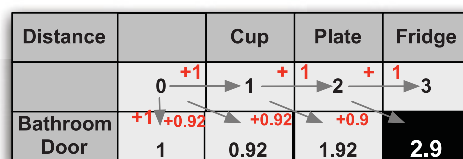
[1]To simplify, we use the objects to replace the sensors in the following representations.

Sensor Similarity	Pan	Plate	Cup	BathroomDoor	Fridge	Freezer
Pan	1	0.64	0.64	0.07	0.67	0.67
Plate		1	0.9	0.08	0.7	0.7
Cup			1	0.08	0.7	0.7
BathroomDoor				1	0.1	0.1
Fridge					1	0.9
Freezer						1

(a) Semantic similarity between sensors



(b) Distance between sq1 and sq2



(c) Distance between sq1 and sq3

Fig. 7. Compare the sequences of sensor events. (a) The semantic similarities between objects in these sensors. (b), (c) Calculations of the semantic distances between these three sensor sequences.

Figure 7(a) shows the semantic similarity between sensors. With the exception of the bathroom door, the other sensors are attached to objects in the kitchen. Here we can also observe that location is a significant factor in distinguishing these sensors. The normalised semantic similarity on the location and object category (see Equation (3)) can strengthen the similarity between sensors in the same location and weaken the similarity between sensors in separate locations.

Figure 7(b) and (c) show how to calculate the distance between sq_1 and sq_2 and that between sq_1 and sq_3, which is significantly different from the originally computed distances. In these two tables, the first data row and column are initialised using the indices of the sequences, and each of the other entries in these two tables is recursively calculated using Definition 4. The calculation starts from the top left of each table and towards the bottom right, whose value represents the total distance between the two sequences of interest. Since the sensors in sq_1 do not share much similarity with the sensors in sq_3, their distance is large and not much different from the original distance of 3. However, the sensors in sq_1 and sq_2 are similar in terms of both location and object categories, so their distance is much smaller.

5.2.2. Clustering Process. The clustering process is described as follows. We randomly choose k patterns as the initial centroids. Then we assign each of the remaining patterns to one of these k groups if its distance to its centroid is closest. The distance measure is the semantics-enhanced edit distance in Definition 4. The centroids are then reselected to choose the pattern with the highest occurrence ratio in each group. These assignment and reselection processes typically go through 100 to 200 iterations, which is considered sufficient for the algorithm to converge.

To reduce the effect of randomly assigning initial centroids, we externally run the preceding algorithm over 100 iterations and choose the result that achieves the minimum cost—the averaged semantic distance of each pattern to its assigned centroid. In terms of choosing the optimal number k, we use the F-test—that is, we set k from 2 to $\sqrt{n}$ (n is the number of patterns in a cluster), and for each k we calculate the percentage of explained variance, which is the ratio of the between-cluster variance over the overall variance. An abrupt change in the percentage of explained variance suggests the corresponding k as an optimal solution. At the current stage, we only employ these classic techniques, because a novel clustering algorithm is not the main objective of our approach. However, we assume that a more effective clustering algorithm could lead to a better activity recognition accuracy. In the future, we will look into more sophisticated clustering algorithms, especially hierarchical connectivity-based clustering.

Completion of the process results in the selection of k patterns for each activity, which represent the most frequent and meaningful sequences. If we consider that the necessary conditions in the AO capture generic knowledge about each activity that is common to most home settings and users, then the patterns learned here that capture diverse and possibly unique ways in which a user might perform an activity can be regarded as their sufficient conditions.

5.3. Activity Recognition

Once the sequential patterns are discovered for each individual activity, we can start recognising activities. Our algorithm works for real-time sensor events. At each time t_i, we evaluate the semantic similarity between the current sensor event se_i and the previous sensor event se_{i-1}. This process is similar to the segmentation algorithm (Algorithm 1). If se_i is close to se_{i-1}, then we join se_i to the tail of the sequence to which se_{i-1} belongs—that is, $\langle \ldots, se_{i-1}, se_i \rangle$. Then we use the new joint sequence to match against the patterns for each activity. The assumption is that the more the evidence accumulates, the more accurate the prediction. If se_i is semantically different from se_{i-1}, we initialise a new sequence to host se_i, which is also matched against the patterns. We rank the matching scores between the acquired sensor fragment and sequential patterns mined in the learning techniques and use the time constraints in AO to filter the ranked activities—in other words, the inference result is the activity whose patterns best match the fragment and whose time constraints are satisfied.

As presented in Section 3.3, we have the *occurring* time and *duration* constraints for activities. The occurring time constraint of an activity is considered *satisfied* if the hour extracted from the timestamp of the current sensor event falls within its range. We only evaluate the duration constraint at the end of each segmentation, where we get the duration of this segmentation by subtracting the timestamp of its last sensor event with the timestamp of its first sensor event. The duration constraint of an activity is considered *satisfied* if the calculated duration falls within its range.

6. EXPERIMENT AND EVALUATION

USMART is built on the assumption that there exist distinguishable semantics between sensor events for different activities. We validate this assumption and evaluate the generality and performance of USMART in activity recognition on datasets that involve different types of object sensors and are collected in different environments and on different subjects.

6.1. Selection of Datasets

To evaluate the effectiveness of our algorithm, we need a dataset that satisfies the following requirements: (1) it contains heterogeneous types of object sensors; (2) it involves single subjects to reduce noise from multiple participants, which has been demonstrated as an issue in our later experiments; and (3) subjects perform multiple repeated activities in a noninterrupted continuous manner rather than performing the same individual activity repetitively in a series of set experiments. Bearing these requirements in mind, we have examined the recent opportunistic challenge dataset [Roggen et al. 2010], which publishes the acceleration and gesture data when subjects were performing breakfast activities in a sensorised laboratory. This dataset is not suitable in our experiment because it does not satisfy the preceding requirements, and at the current stage our technique is unable to explore semantics beyond the object sensors. We have looked through the publicly available datasets, including all datasets listed in the “Home Datasets.”[2] Among them, the datasets that are publicly available and satisfy the requirements are those collected by the University of Amsterdam [Van Kasteren et al. 2008], by the Washington State University [Cook 2012], and by MIT [Logan et al. 2007].

The dataset collected by the University of Amsterdam [Van Kasteren et al. 2008] contains three datasets (called House A, B, and C in the following) that are collected in three different houses with different subjects. Each of the Houses A, B, and C was instrumented with wireless sensor networks to observe the daily activities of the inhabitants. In House A, the sensor network is composed of 14 state-change sensors on household objects like doors, cupboards, or toilet flush [Van Kasteren et al. 2008]. In Houses B and C, each network node was equipped with heterogeneous sensors: passive infrared to detect motion in a specific area; pressure mats to measure whether someone is sitting on a couch or lying in bed; switches to monitor whether doors and cupboards are open or closed; mercury contacts to detect the movement of objects, such as drawers; and water flow sensors to detect the flush of toilet. All of these sensors output binary readings (0 or 1), indicating whether or not a sensor fires. In our experiment, we generate the change point representation (i.e., a reading changes from 0 to 1 or vice versa) from the raw sensor representation of the datasets.

The monitored activities include leaving the house, taking a shower, using the toilet, going to bed, brushing teeth (only in Houses B and C), preparing breakfast, preparing dinner, and getting a drink.[3] These activities were reported by the inhabitants using either a handwritten activity diary or a Bluetooth headset installed with speech recognition software.

The dataset collected by the Washington State University mainly consists of motion sensors, which are placed in the environment rather than attached to particular objects, so the use of semantic similarity between these sensors could be limited. In addition, this dataset shares the features of House A, B, and C datasets.

Another dataset under consideration is the PlaceLab Couple dataset [Logan et al. 2007]. To the best of our knowledge, this dataset is by far the most complex and largest dataset collected in a real-world environment and is publicly available. The PlaceLab dataset contains more than 900 sensor inputs, among which 707 are object sensors, including the wireless infrared motion sensors, “stick-on” object motion sensors, switch sensors, and RFIDs. The dataset was gathered over a period of 15 days that a married couple (who were unaffiliated with the PlaceLab research) lived in the PlaceLab, which generated 455,852 object sensor events in total. During this period, the couple followed

[2]List of home datasets: http://boxlab.wikispaces.com/List+of+Home+Datasets.

[3]In the following figures and tables, we will use the abbreviated names for each of these activities: *Leave*, *Shower*, *Sleep*, *Teeth*, *Breakfast*, *Dinner*, and *Drink*.

Table II. Defining Necessary Conditions on Activities in the PlaceLab Dataset

Activity	Key Objects	Location
Phone	Phone	House
Computer	Computer, Electronics	LivingRoom, Office
TV	Television, RemoteControl	LivingRoom
Hygiene	Faucet, Toilet, Cabinet	Bathroom
Meal	Food, KitchenUtensil	Kitchen
Dress	Closet	Bedroom

a normal life routine. The activities of the subjects were recorded with an audiovisual recording infrastructure instrumented in the apartment. The video was annotated as a ground truth representing the activities that were taking place over the period of study. So far, only the activities of the male subject have been annotated. Even though this dataset involves two subjects, such a large number of sensors will be useful to demonstrate the generality of our ontological model and extensibility of our approach.

The annotated human activities mainly consist of using the phone, using a computer, watching TV, performing hygiene maintenance, having a meal, and getting dressed or undressed. Their necessary conditions are specified in Table II. In the PlaceLab dataset, we do not consider any time constraints for two reasons: (1) we only consider activities during a certain period of a day, and (2) two subjects could perform different activities at the same time, resulting in the interleaved sensor data, and the duration constraint will not hold.

We note that the following experiments are based on the sameOOs and LOs. Houses A, B, and C share the same AOs. For example, even though the four environments have different spatial layouts, we map the locations based on their functionality rather than spatial containment relations. For instance, House C has two toilets downstairs and upstairs, and we map both to the toilet in Figure 3 rather than distinguishing them. For the sensors in each environment, we define them by linking to the preceding OOs and LOs. We extract the OO from WordNet and manually check the noun terms that are closest to object instances tagged in all of these environments. The extracted OO contains 276 instances.

In summary, we believe that the variety of these datasets makes them well suited for evaluating that our technique is general enough to support knowledge transfer across different home domains and users.

6.2. Evaluation Methodology

As the segmentation is based on semantic distance between consecutive sensor events and does not involve any training, we run the segmentation algorithm on the whole sensor data sequence once and measure the precision and recall on the event basis. We consider a segmentation successful if the time at the Detected Segmentation ds (either the start or end time) is within the range of the time at the Recorded Activity ra (again, either start or end time)—that is, $ds \in [ra - tt, ra + tt]$, where tt is the Time Tolerance. Precision here indicates the ratio of the times that the detected transitions between segmentations are actual transitions in the recorded activities, and recall indicates the ratio of the times that the actual transitions in the recorded activities are successfully detected. We also measure the *exact* match, meaning that the detected segmentation $[ds\_start, ds\_end]$ overlaps with the time interval of the recorded activities $[ra\_start, ra\_end]$—that is, $ds\_start \in [ra\_start - tt, ra\_start + tt]$ and $ds\_end \in [ra\_end - tt, ra\_end + tt]$. Note that the time tolerance is a parameter to measure the match of time intervals, which is not part of segmentation and recognition algorithms.

The evaluation of recognition is conducted using the leave-one-day-out technique, where each full day of sensor data is used for testing once and the remaining days of sensor data are used for training to build the user sequential behaviour model. Both the inference result and the recorded activities are represented in a sequence of event tuples `(activity, [startTime, endTime])`. To make comparison easier, we split the time interval of each tuple into 1-minute chunks—that is, `(activity, [startTime, startTime + 1 min))`, `(activity, [startTime + 1 m, startTime + 2 min))`,...,`(activity, [startTime + m min, endTim])`. We compare the inferred and recorded activities at each 1-minute chunk. We do not consider the null type of activities in our evaluation simply because they are time periods with no associated activity annotation in the ground truth, and thus it is impossible to define necessary conditions on them.

The accuracy of recognising activities is evaluated using two parameters: *precision* and *recall*. Precision is the ratio of the times that an activity is correctly recognised to the times that it is inferred. Recall is the ratio of the times that an activity is correctly inferred to the times that it actually occurs. As indicated in Van Kasteren et al. [2008], the datasets are imbalanced in that some types of activities appear more frequently than other types. When measuring the averaged performance across all activities, we consider both the timeslice and class accuracies suggested in Van Kasteren et al. [2008]: the timeslice accuracy is the averaged accuracy over the 1-minute chunks, and the class accuracy is the averaged timeslice accuracy per activity class—that is, $class\_precision = \frac{\sum_{a=1}^{A} precision(a)}{A}$ and $class\_recall = \frac{\sum_{a=1}^{A} recall(a)}{A}$, where A is the total number of the activity classes, and $precision(a)$ and $recall(a)$ are the precision and recall of the averaged timesliced accuracy on the activity class a.

6.3. Choosing Thresholds

For segmentation and activity recognition, we need to choose the *maximum time* and *semantic* thresholds. For the maximum time threshold, we consider 10 minutes as a reasonable gap between activities—in other words, if there are no sensor events reported within 10 minutes, then the user might transit to another activity. Most often, we segment the sensor traces based on their semantic similarity. Here we set the time threshold mainly because the semantic similar (or the same) activities could occur continuously but with a certain gap between the occurrences. These activities usually occur frequently but do not necessarily last long. For example, the user could go to the toilet twice continuously with the gap of 1 hour. Using only the semantic threshold, we would be unable to separate these two events. We have evaluated the effect of different maximum time thresholds on the recognition accuracy. The result shows that the thresholds do not have a major effect on the overall recognition accuracy but slightly affect the "Toilet" activity—that is, in the House A dataset, compared to not setting the time threshold, the recall on this activity has increased from 58% to 80% with the time threshold as 10 minutes.

For the semantic thresholds, we first calculate the similarity measures between sensors in activities. In Section 3.3, we show that each activity can be mapped with a set of sensors. Then for any two activities A, A', we compute the semantic similarity measures between their mapped sensors S, S' and choose the smallest nonzero value as the minimum similarity measure (MSM) between A and A'. The MSM $msm(A, A')$ is calculated as follows:

$$msm(A, A') = min(sim_S(s_i, s_j)), \forall s_i \in S, s_j \in S', sim_S(s_i, s_j) > 0.$$

The semantic threshold is the maximum value among the MSMs between any two activities. We have listed the measures between activities in House A in Table III. For

Table III. Minimum Similarity Measure between Activities in House A with the Selected Threshold Emboldened

Activities	Leave	Toilet	Shower	Sleep	Breakfast	Dinner	Drink
Leave		0	0	0.417	0	0	0.083
Toilet			**0.625**	0	0	0	0
Shower				0	0	0	0
Sleep					0	0	0.083
Breakfast						0.562	0.5
Dinner							0.5

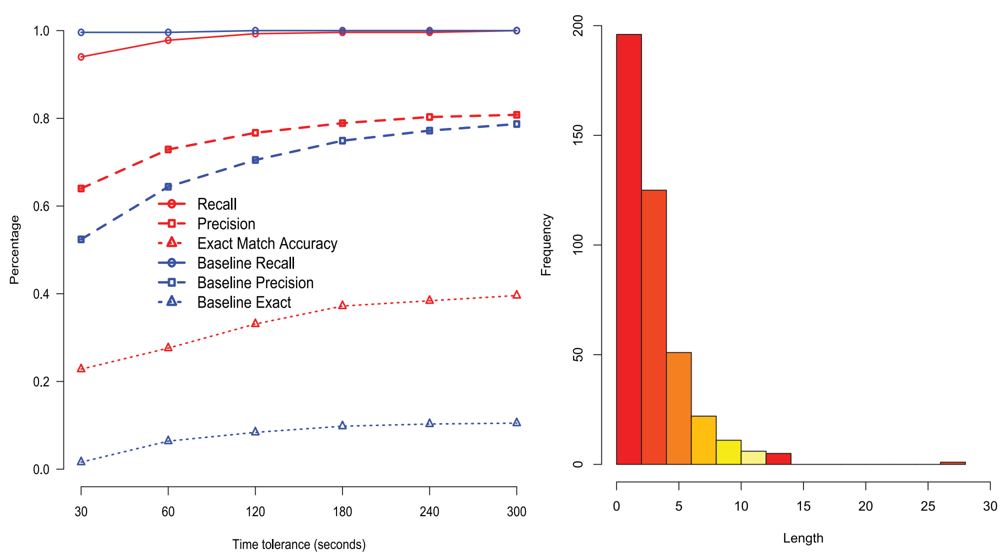


Fig. 8. Evaluating segmentation on the House A dataset with regard to the precision and recall of segmentation (left) and the histogram on lengths of the segments (right).

example, the MSM between "Breakfast" and "Dinner" is 0.562, and the MSM between "Breakfast" and "Toilet" is 0. The maximum among all of these MSMs is 0.625, which is then selected as the semantic threshold for this dataset.

6.4. Segmenting the Datasets

Figures 8, 9, and 10 present the precision and recall of segmentation on the House A, B, and C datasets. For example, the precision, recall, and exact match accuracy of the segmentation algorithm on the House A dataset are 64%, 99.6%, and 22.8% within the time tolerance of 30 seconds. We compare with the baseline approach (in green lines) where each single sensor event is assumed as a segmentation. In terms of the baseline results, we expect high recall (i.e., the change of activities should be observed at change of sensor states) and low precision (i.e., performing an activity usually involves several interactions with more than one sensor). The detected segmentations also exhibit such expectations, as the number of the sensor events that are collected in one segment is not great, indicating that the whole progress of one activity has been partitioned into several segments. Figures 8, 9, and 10 show the histograms on lengths of segmentations in the House A, B and C datasets, where the length is the number of events contained in one segmentation. The ratio of one-length segmentations among all segmentations in Houses A, B, and C are 30% (150 out of 503), 50% (380 out of 762), and 36% (330 out of

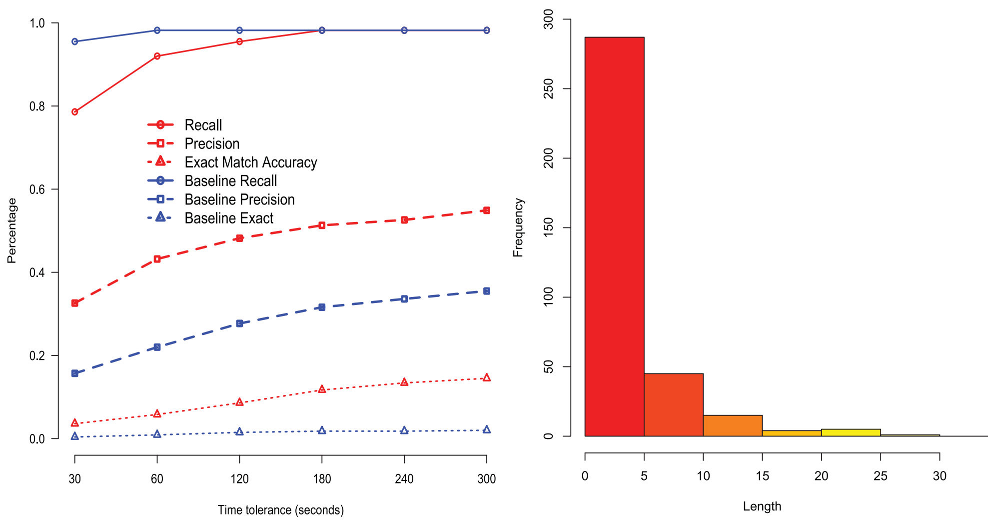


Fig. 9. Evaluating segmentation on the House B dataset with regard to the precision and recall of segmentation (left) and the histogram on lengths of the segments (right).

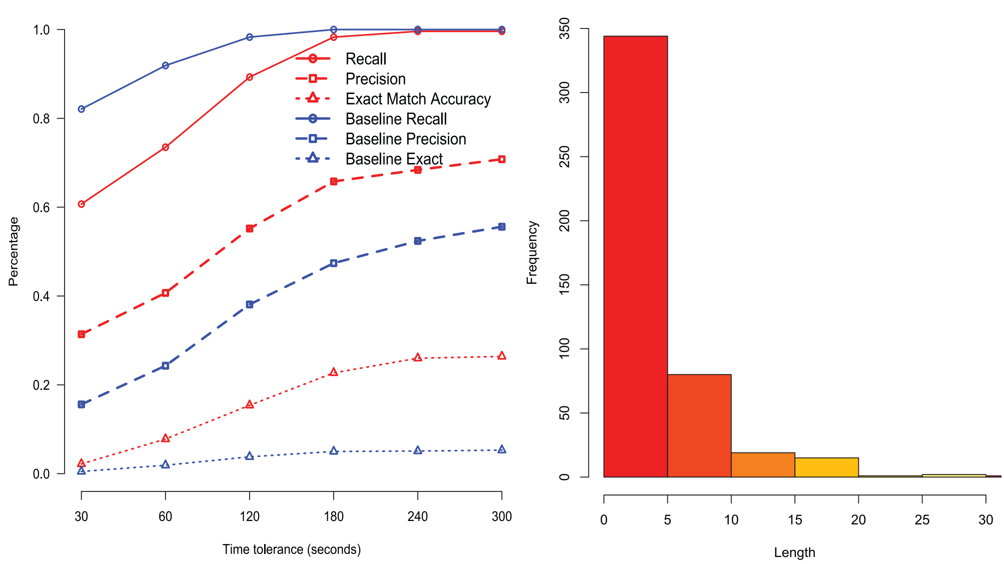


Fig. 10. Evaluating segmentation on the House C dataset with regard to the precision and recall of segmentation (left) and the histogram on lengths of the segments (right).

910), respectively. In addition, we have obtained a better balanced size of segmentations in Houses A and C compared to House B.

Looking at the data for House B, we find that this dataset is noisy. Its diary data records that the inhabitant was preparing dinner from 20:21:50 to 20:39:39 on 24th July. The recorded sensor events during this period include (20:22:10, bathroom door), (20:22:23, stove), and (20:23:46, toilet flush). We assume that either there is another user visiting the house or there is a sensor error. Our algorithm partitions each of

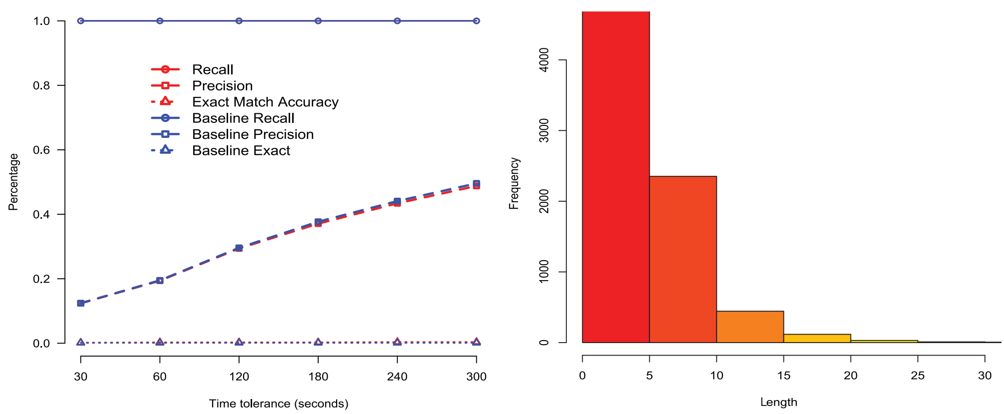


Fig. 11. Evaluating segmentation on the Placelab dataset with regard to the precision and recall of segmentation (left) and the histogram on lengths of the segments (right).

these sensor events as a separate segmentation, which results in the low precision and accuracy in exact match. A similar noise problem is even worse in the PlaceLab dataset, as shown in Figure 11, where both precisions and recalls of our segmentation and baseline approaches are identical. As we have mentioned that there are two inhabitants living in the environment configured with the nonidentity-distinguishable sensors, the sensor sequences are constantly interleaved with semantically completely different sensor events. The histogram shows that 55% of the segmented sequences are one length, so the precision and recall of segmentation is almost identical to the baseline.

Overall, our segmentation technique performs better than the baseline approach on all four datasets, achieving high precision without sacrificing much recall. It validates our assumption that there exists semantic similarity between sensor events and that such similarity can be used to automatically partition sensor events into coherent fragments, each of which corresponds to the same activity. However, the limited ability to detect exact activity boundaries is a common property for most of the segmentation techniques [Krishnan and Cook 2014]. To the best of our knowledge, we are the only ones demonstrating the segmentation performance in this way.

6.5. Recognising Activities

After segmentation, we discover sequential patterns from the sensor fragments, map the patterns to each activity based on their structural semantics, and cluster the patterns to arrive a concise set of frequent and meaningful patterns. We perform activity recognition by matching the incoming sensor events on these patterns and choose the activity whose pattern best matches the input sensor sequence and whose time condition is satisfied.

The recognition accuracies on the House A, B, and C datasets are presented in Figures 12, 13, and 14. The activities “Leave” and “Sleep” achieve the highest accuracies, because they have the most distinguishable object and location semantics that no other activities share with them—that is, the “Leave” involves the access to the front door, and “Sleep” involves the access to the bed in the bedroom. Both of these objects are not normally associated with any other activities. We have achieved high precision and recall when recognising both activities in all three datasets.

Our algorithm performs best on the House A dataset. However, the results highlight that it poorly distinguishes two activities that are semantically close and occur closely together—in other words, the gap between their occurrence is not beyond the time

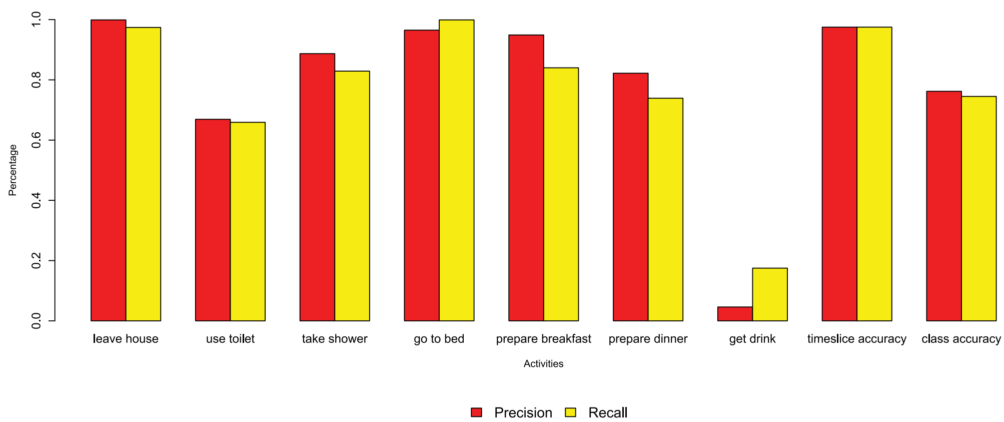


Fig. 12. Precision and recall of activity recognition on the House A dataset.

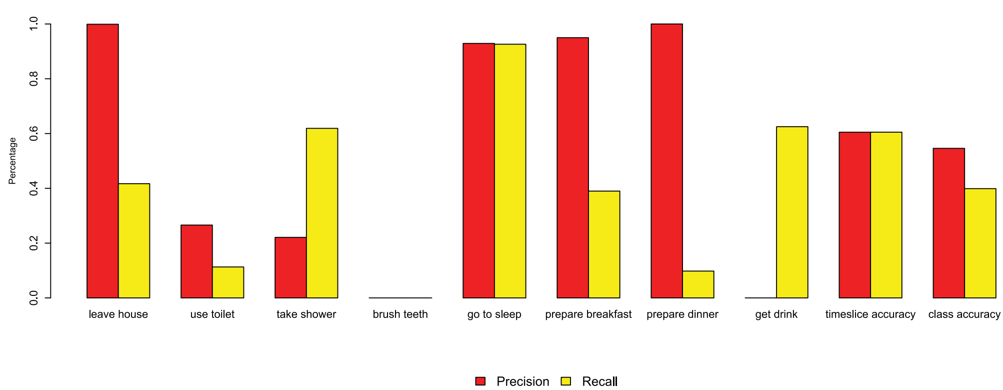


Fig. 13. Precision and recall of activity recognition on the House B dataset.

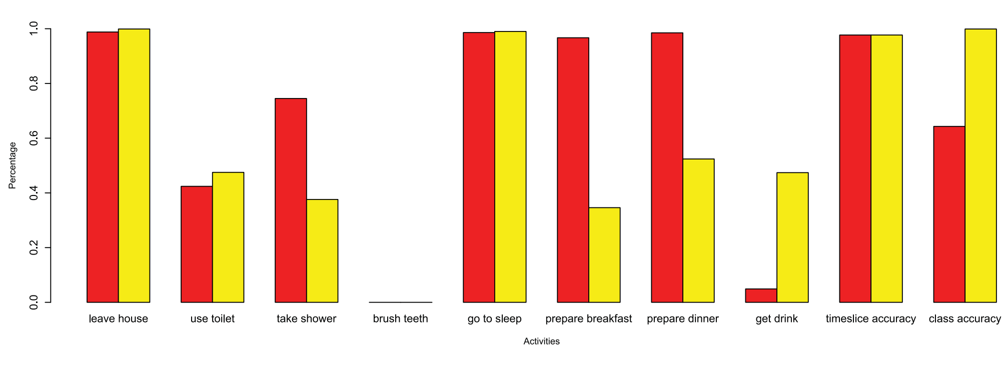


Fig. 14. Precision and recall of activity recognition on the House C dataset.

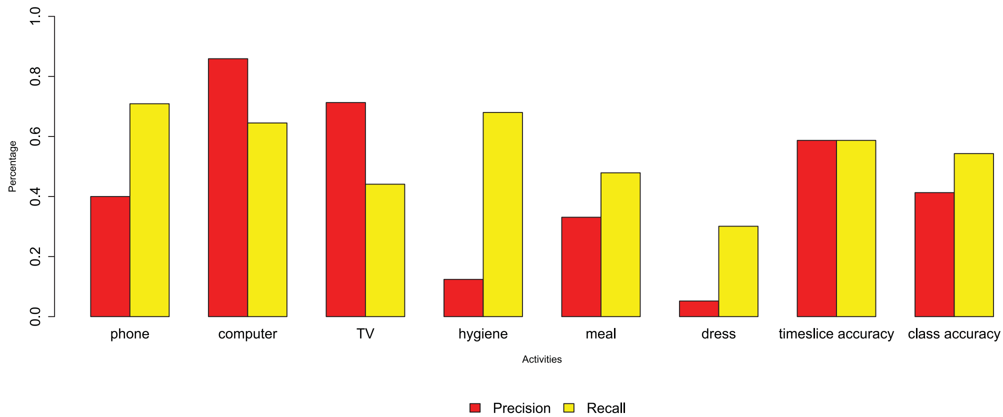


Fig. 15. Precision and recall of activity recognition on the Placelab dataset.

threshold. For example, we cannot distinguish “Shower” immediately after “Toilet,” even though “Toilet” involves a very distinguishable object toilet flush. The problem is that the toilet flush sensor usually fires between the interactions with the toilet door object, and the toilet door is evaluated to be close to the bathroom door. Thus, if these two activities are performed in a close order, we are less likely to segment the sensor sequence correctly. According to the duration constraint, we infer “Shower” as a final result. The same problem happens between “Drink” and the other two kitchen activities of “Breakfast” and “Dinner.”

This issue is also apparent on Houses B and C. In terms of the House B dataset, we barely detected the “Teeth” activity because no sensors are linked to uniquely related objects, such as the toothbrush or toothpaste. We only set its location constraint as “hygiene area,” which means that the activity can be associated with any sensors that are installed in this area. The lack of distinguishable objects results in the low recognition rate and adds noise to the inference of other similar activities (as in House C).

We compare our results with the benchmark results published in Kasteren et al. [2011], where the authors applied naive Bayes, hidden (semi-)Markov models and conditional random fields on these three datasets. On the change of point representations as we use, the best class F-measurements[4] that they achieve on the Houses A, B, and C are 0.72, 0.47, and 0.56, which are close to our results of 0.74, 0.46, and 0.58. We also compare our results with the result published by McKeever et al. [2010], who extend the Dempster-Shafer evidential theory for activity recognition and perform the evaluation on the House A dataset. The comparison shows that our technique achieves higher F-measurements on almost all activities except the “Toilet” and “Drink” activities. By combining the domain knowledge with learning techniques, USMART as an unsupervised learning technique can achieve comparable recognition accuracies to these pure learning and knowledge techniques. We refer to these results only as a reference of the improvement rather than a proper comparison due to the differences in segmentation strategies, knowledge engineering, and the use of labelled training data.

The recognition result on the PlaceLab dataset is shown in Figure 15 and is consistent with the segmentation findings. The interleaved activities performed by different users make it difficult to extract sequence patterns with semantically consistent sensor events. This is more apparent on personal activities like hygiene and getting dressed or

[4]F-measurement is a balanced measure of precision and recall together.

undressed but less apparent on entertainment activities like watching TV and working on computers. That is, while the male subject is using the bathroom, the female subject could be performing other activities and interact with other sensors in the apartment. As they are more likely to watch TV together, less interfering sensor events will be triggered. We would expect to achieve better results if the sensor installation were capable of identifying interactions with each individual.

7. DISCUSSION

In this section, we discuss the utility of the USMART technique and the practical issues of using it from the perspective of interactive systems.

7.1. Formality and Generality

USMART is built on top of a general ontological model that represent domain concepts, sensor events, and human activities in a smart home environment. The generality and feasibility of the ontologies have been demonstrated across four different real-world smart home datasets with different user subjects, home layouts, and sensor deployments. The OOs are derived from an open knowledge base (e.g., WordNet), and the LOs are defined on the function-based relationship, which is more general than the usually used spatial containment relationship. The same set of the LOs and OOs form as a conceptual foundation to describe the sensors and specify activities in these four datasets.

Within this ontological model, we specify the certain and concise necessary conditions on the daily activities—capturing the commonsense knowledge about how people perform certain activities, which helps to reduce the amount of knowledge engineering effort required by developers and reduce the bias introduced by expert knowledge compared to the pure knowledge-driven techniques. The experiments on the House A, B, and C datasets have demonstrated that the activity profiles are general enough to be applied across different users without much reengineering effort.

7.2. Online Sensor Data Segmentation

In USMART, we explore semantics of sensor events from the temporal, spatial, and object dimensions rather than simply treating them as meaningless symbols. Based on the semantics, USMART supports online segmenting a continuous sequence of sensor events without the need of any training data to learn the correlations between sensor events [Krishnan and Cook 2014] or the use of any activity-level knowledge [Tapia et al. 2004b; McKeever et al. 2010; Okeyo et al. 2014]. This algorithm is useful on the unconstrained real-world datasets, which are collected when users interact with sensors in a natural way rather than being given an explicit instruction. It can automatically segment sensor sequences into meaningful partitions, which will not only benefit activity recognition but also enable a more subtle and adaptive interaction design.

7.3. Knowledge-Assisted Unsupervised Learning

USMART is an unsupervised learning algorithm that leverages the ontological knowledge in the application of statistical techniques. In the training process, we automatically map the segmented fragments to activities by reasoning on the semantics between sensors and activities in the preceding ontologies. A sequential mining and clustering technique is applied to explore the most frequent sequential patterns for each activity. These discovered sequential patterns reflect the individual ways in which each user performs a certain activity, including which other objects are accessed and in which order a user interacts with these objects. The activity recognition process seeks to find the sequential pattern that matches best to real-time sensor segment through a semantics-augmented string alignment technique. The recognition accuracy of the recognition algorithm matches to the performance of the state-of-the-art supervised techniques, which validates our assumption and demonstrates the viability of our approach—that is, the semantics underlying object sensor events can help to segment sensor traces and support activity recognition.

7.4. Application and Implication to Wider Interactive Systems

Currently, USMART is centred around user–object interaction—in other words, monitoring user interactions with sensorised everyday objects, analysing the semantics underlying their interactions, and thus inferring daily activities. Compared to data-driven techniques discussed in Section 2 that perform better on numeric sensor data, USMART is more amenable for high-level characteristic sensor data, which are often captured in interactive systems via featured sensors such as Kinect. USMART provides better facilities to allow interactive system designers to express and finely tune domain knowledge.

This technique has the potential to be employed in a wider interactive system as a more subtle, unobtrusive, smooth way of detecting and capturing user intention and tasks into semantic-integral units and of tracking their transition and progression. For example, in public display systems, with the help of a proper spatial ontology, the principle of USMART can be similarly employed to analyse the semantics underlying different spatial cues including location, proximity, direction of movement, and speed to derive user attention on the display and further infer user interest in the displayed content. By monitoring their interest, we can derive the implicit change in their intention to provide more adaptive and customised feedback to users, compared to the feedback that is solely based on the current activity. The advantages of USMART in automatic segmentation and unsupervised learning facilitate this feature in that it reduces the overhead in collecting and labelling a comprehensive dataset as well as in training. These features are crucial in designing interactive systems that aim towards public users in an open environment, where it is less likely to collect sufficient training data on how each potential user will interact with the system.

7.5. Limitations and Future Work

There are several directions to improve the performance of USMART. First of all, we consider that the time similarity function sim_T can exist in two forms: fine- and coarse grained. If there exists a hierarchy of temporal concepts similar to the object or location concepts, then its coarse-grained form is calculated as Definition 3. In this work, we take the fine-grained form and note designing high-level temporal ontologies to characterise temporal features in activity patterns as a future research direction.

As indicated in Section 6, USMART is sensitive to sensor noise, for it only compares a pair of consecutive sensor events. We intend to extend the segmentation methodology to consider more events at a time. To do so, we will design a more flexible approach to dynamically adjust and balance the time and semantic thresholds to collect more events that are similar enough but may contain sporadic sensor noise. In addition, we intend to employ the evidence aggregation techniques like voting or Dempster-Shafer theory to reach a more consistent conclusion.

The current version of USMART does not perform well in distinguishing semantically similar activities that are occurring closely together. One of our future works is to exploit finer ways to specify necessary conditions on activities and to use a utility function to combine the time, location, and object constraints in a more intelligent manner. We will also look into how to apply a more robust technique [Von Luxburg 2010], or Fuzzy logic and connectivity-based clustering techniques to semantics-based sequence matching and clustering. With the help of these techniques, we expect to capture finer-grained and more distinguishable activity patterns.

Another research direction towards this issue is to incorporate the other contextual cues to distinguish these semantically similar activities. For example, the electricity and water flow monitoring sensors might help to distinguish the activities of taking a shower and using the toilet. We intend to extend the ontological model to express the semantics of these types of sensors and define a more generic semantic similarity function to take into account the correlations between these sensors and object sensors.

8. CONCLUSION

This article has presented USMART as a novel contribution to user–object interaction–based activity recognition. Without relying on annotated training data, this technique leverages the semantics of sensor events in a general ontological model, supports automatically segmenting a continuous sensor stream, and reaches the best accuracies of activity recognition from a range of supervised techniques. It paves the future research direction on applying an open knowledge source to enhance the effectiveness of data-driven activity recognition.

REFERENCES

Ling Bao and Stephen S. Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*. Lecture Notes in Computer Science, Vol. 3001. Springer, 1–17. DOI:http://dx.doi.org/10.1007/978-3-540-24646-6_1

Oliver Brdiczka, James L. Crowley, and Patrick Reignier. 2007. Learning situation models for providing context-aware services. In *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction: Ambient Interaction (UAHCI'07)*. Springer-Verlag, Berlin, Heidelberg, 23–32. http://dl.acm.org/citation.cfm?id=1763296.1763300

Hung H. Bui, Svetha Venkatesh, and Geoff West. 2002. Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model. In *Hidden Markov Models: Applications in Computer Vision*. World Scientific Publishing, River Edge, NJ, 177–196.

Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. 2012a. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42, 6, 790–808. DOI:http://dx.doi.org/10.1109/TSMCC.2012.2198883

Liming Chen, Chris D. Nugent, and Hui Wang. 2012b. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering* 24, 961–974. DOI:http://dx.doi.org/10.1109/TKDE.2011.51

Diane Cook. 2012. Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems* 27, 1, 32–38. DOI:http://dx.doi.org/10.1109/MIS.2010.112

Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning* 32, 1, 41–62.

Tao Gu, Shaxun Chen, Xianping Tao, and Jian Lu. 2010. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data and Knowledge Engineering* 69, 6, 533–544. DOI:http://dx.doi.org/10.1016/j.datak.2010.01.004

Tao Gu, Hung Keng Pung, and Da Qing Zhang. 2005. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications* 28, 1, 1–18.

Tao Gu, Liang Wang, Zhanqing Wu, Xianping Tao, and Jian Lu. 2011. A pattern mining approach to sensor-based human activity recognition. *IEEE Transactions on Knowledge and Data Engineering* 23, 9, 1359–1372. DOI:http://dx.doi.org/10.1109/TKDE.2010.184

Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. 2004. An ontology-based context model in intelligent environments. In *Proceedings of the International Conference on Communication Network and Distributed Systems (CNDS'04)*. 270–275.

Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. 2009. epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'09)*. 1–9.

Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

Xin Hong and Chris D. Nugent. 2009. Partitioning time series sensor data for activity recognition. In *Proceedings of the 9th International Conference on Information Technology and Applications in Biomedicine (ITAB'09)*. 1–4. DOI:http://dx.doi.org/10.1109/ITAB.2009.5394306

Clare J. Hooper, Anne Preston, Madeline Balaam, Paul Seedhouse, Daniel Jackson, Cuong Pham, Cassim Ladha, Karim Ladha, Thomas Plötz, and Patrick Olivier. 2012. The French kitchen: Task-based learning in an instrumented kitchen. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12)*. ACM, New York, NY, 193–202. DOI:http://dx.doi.org/10.1145/2370216.2370246

Tâm Huỳnh, Ulf Blanke, and Bernt Schiele. 2007. Scalable recognition of daily activities with wearable sensors. In *Proceedings of the 3rd International Conference on Location and Context Awareness (LoCA'07)*. 50–67. http://dl.acm.org/citation.cfm?id=1777235.1777241

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference Research on Computational Linguistics*. 9008+. http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1997cmp.lg....9008J

Tim L. M. Kasteren, Gwenn Englebienne, and Ben J. A. Kröse. 2011. Human activity recognition from wireless sensor network data: Benchmark and software. In *Activity Recognition in Pervasive Intelligent Environments.* Atlantis Ambient and Pervasive Intelligence, Vol. 4. Springer, 165–186.

Narayanan C. Krishnan and Diane J. Cook. 2014. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing* 10, 138–154. DOI:http://dx.doi.org/10.1016/j.pmcj.2012.07.003

Javier Ortiz Laguna, Angel García Olaya, and Daniel Borrajo. 2011. A dynamic sliding window approach for activity recognition. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization (UMAP'11)*. Springer-Verlag, Berlin, Heidelberg, 219–230. http://dl.acm.org/citation.cfm?id=2021855.2021875

Claudia Leacock, George A. Miller, and Martin Chodorow. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* 24, 1, 147–165. http://dl.acm.org/citation.cfm?id=972719.972726

Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille. 2007. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp'07)*. Springer-Verlag, Berlin, Heidelberg, 483–500. http://dl.acm.org/citation.cfm?id=1771592.1771620

Susan McKeever, Juan Ye, Lorcan Coyle, and Simon Dobson. 2010. Activity reasoning using temporal evidence theory. *Ambient Intelligence and Smart Environment* 2, 3, 253–269.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11, 39–41. DOI:http://dx.doi.org/10.1145/219717.219748

George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. 2011. Ontology-based learning framework for activity assistance in an adaptive smart home. In *Activity Recognition in Pervasive Intelligent Environments.* Atlantis Ambient and Pervasive Intelligence, Vol. 4. Springer, 237–263. DOI:http://dx.doi.org/10.2991/978-94-91216-05-3_11

George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. 2014. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing* 10, 155–172. DOI:http://dx.doi.org/10.1016/j.pmcj.2012.11.004

Paulito Palmes, Hung Keng Pung, Tao Gu, Wenwei Xue, and Shaxun Chen. 2010. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive and Mobile Computing* 6, 1, 43–57. DOI:http://dx.doi.org/10.1016/j.pmcj.2009.10.004

Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC'05)*. IEEE, Los Alamitos, CA, 44–51. DOI:http://dx.doi.org/10.1109/ISWC.2005.22

Martin Petzold, Francesco Barbabella, Jan Bobeth, Dagmar Kern, Christopher Mayer, and Martin Morandell. 2013. Towards an ambient assisted living user interaction taxonomy. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems (CHI EA'13)*. ACM, New York, NY, 49–54. DOI:http://dx.doi.org/10.1145/2468356.2468366

Parisa Rashidi and Diane J. Cook. 2010. Mining sensor streams for discovering human activity patterns over time. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10)*. IEEE, Los Alamitos, CA, 431–440. DOI:http://dx.doi.org/10.1109/ICDM.2010.40

Parisa Rashidi, Diane J. Cook, Lawrence B. Holder, and Maureen Schmitter-Edgecombe. 2011. Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering* 23, 4, 527–539.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol. 1*. Morgan Kaufmann, San Francisco, CA, 448–453.

Daniele Riboni and Claudio Bettini. 2009. Context-aware activity recognition through a combination of ontological and statistical reasoning. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC'09)*. Springer-Verlag, Berlin, Heidelberg, 39–53.

Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Killian Forster, Gerhard Troster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and Jose del R. Millan. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *Proceedings of the 7th International Conference on Networked Sensing Systems (INSS'10)*. 233–240.

Geetika Singla and Diane J. Cook. 2010. Recognizing independent and joint activities among multiple residents in smart environments. *Ambient Intelligent and Humanized Computing* 1, 1, 57–83.

Emmanuel Munguia Tapia, Tanzeem Choudhury, and Matthai Philipose. 2006. Building reliable activity models using hierarchical shrinkage and mined ontology. In *Pervasive Computing*. Lecture Notes in Computer Science, Vol. 2968, Springer, 17–32.

Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004a. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing.* Lecture Notes in Computer Science, Vol. 3001. Springer, 158–175. DOI:http://dx.doi.org/10.1007/978-3-540-24646-6_10

Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004b. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing.* Lecture Notes in Computer Science, Vol. 3001. Springer, 158–175. DOI:http://dx.doi.org/10.1007/978-3-540-24646-6_10

Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp'08)*. ACM, New York, NY, 1–9. DOI:http://dx.doi.org/10.1145/1409635.1409637

Ulrike Von Luxburg. 2010. Clustering stability: An overview. *Foundations and Trends in Machine Learning* 3, 2, 235–274.

Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. 2007. Common sense based joint training of human activity recognizers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*. Morgan Kaufmann, San Francisco, CA, 2237–2242. http://dl.acm.org/citation.cfm?id=1625275.1625636

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*. Association for Computational Linguistics, Stroudsburg, PA, 133–138. DOI:http://dx.doi.org/10.3115/981732.981751

Juan Ye, Simon Dobson, and Susan McKeever. 2012. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing* 8, 1, 36–66.

Shumin Zhai and Victoria Bellotti. 2005. Introduction to sensing-based interaction. *ACM Transactions on Computer-Human Interaction* 12, 1, 1–2. DOI:http://dx.doi.org/10.1145/1057237.1057238

Vincent Wenchen Zheng, Derek Hao Hu, and Qiang Yang. 2009. Cross-domain activity recognition. In *Proceedings of the 11th International Conference on Ubiquitous Computing (Ubicomp'09)*. ACM, New York, NY, 61–70. DOI:http://dx.doi.org/10.1145/1620545.1620554

Received September 2013; revised June 2014; accepted June 2014