Mining Emerging Patterns for Recognizing Activities of Multiple Users in Pervasive Computing

Tao Gu*, Zhanqing Wu†, Liang Wang†, Xianping Tao† and Jian Lu†
*Department of Mathematics and Computer Science, University of Southern Denmark
Email: gu@imada.sdu.dk
†State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: {wzq,wangliang,txp}@ics.nju.edu.cn, lj@nju.edu.cn

***Abstract*—Understanding and recognizing human activities from sensor readings is an important task in pervasive computing. Existing work on activity recognition mainly focuses on recognizing activities for a single user in a smart home environment. However, in real life, there are often multiple inhabitants live in such an environment. Recognizing activities of not only a single user, but also multiple users is essential to the development of practical context-aware applications in pervasive computing. In this paper, we investigate the fundamental problem of recognizing activities for multiple users from sensor readings in a home environment, and propose a novel pattern mining approach to recognize both single-user and multi-user activities in a unified solution. We exploit Emerging Pattern – a type of knowledge pattern that describes significant changes between classes of data – for constructing our activity models, and propose an Emerging Pattern based Multi-user Activity Recognizer (*epMAR*) to recognize both single-user and multi-user activities. We conduct our empirical studies by collecting real-world activity traces done by two volunteers over a period of two weeks in a smart home environment, and analyze the performance in detail with respect to various activity cases in a multi-user scenario. Our experimental results demonstrate that our epMAR recognizer achieves an average accuracy of 89.72% for all the activity cases.**

I. Introduction

In recent years, human activity recognition, which aims to recognize the actions and goals of one or more agents from a series of observations [1], has become an important research direction in pervasive computing. In this paradigm, various sensors are typically deployed to collect a sequence of observations, and these observations are used to train an appropriate activity model. The trained model can then be used to assign new observations with activity labels. There are many useful applications such as monitoring Activities of Daily Living (ADLs) [2] for the elderly and cognitively impaired persons, and providing proactive assistance [3].

Understanding and recognizing human activities based on sensor readings is interesting since sensors can capture many useful low-level features of human users and their living environments. However, it is challenging because of noisy sensor data and complex pattern of human activities. To recognize various activities accurately and reliably, an appropriate activity model has to be deployed to relate low-level features to high-level concepts. Most existing work focuses on recognizing activities of a single user [1, 8-23]. However, in a living space (i.e., typically a home environment), there are usually multiple inhabitants. In such an environment, a habitant may perform her/his daily activities independently, and often, two or more habitants may perform activities together. For example, in daily life, it is common for family members to prepare meals, do a household cleaning, or watch TV together. We call this problem the multi-user activity recognition problem. Recognizing activities of both a single user and multiple users has practical implications for many real-world applications.

A. Problem Analysis

A deep understanding of this problem is essential to both computer scientists and psychologists. To investigate the different situations of multi-user activities in real life, we conducted an interview survey among students in a local university. In this survey, each interviewee was given a list of ADLs and required to answer a number of questions related to multi-user activities in their daily lives. The list contains all the ADLs researchers commonly studied. The questions include which ADL in your daily life is often performed by two or more family members; for each of multi-user activities, describe the steps of how the activity is performed in detail and the role of each participant, etc. After analyzing the result, we categorize the different cases of ADLs as follows.

1) Single-user sequential ADLs: In this category, a single user performs an activity in a sequential manner (i.e., one activity after another) independently. The ADLs of this category correspond to intrapsychological functions, such as brushing teeth or making coffee. Most of existing work [1, 8-23] focuses on this type of ADLs.

2) Single-user sequential ADLs with simultaneity: In this category, two or more users perform the same activity in a sequential manner both independently and simultaneously. The ADLs of this category also correspond to intrapsychological functions, such as two or more users are eating meal or drinking coffee together.

3) Multi-user ADLs with collaboration: The ADLs of this category involve two or more users working together to complete an activity in a cooperative manner, where each of them performs a partial step of the activity. The ADLs of this category correspond to interpsychological functions [4]. For example, two users are making pasta together, i.e., one user

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

is boiling pasta in the pot while the other user is preparing tomato sauce by cutting tomatoes and onions.

4) Multi-user ADLs with conflict: In this category, two or more users are involved in an activity in a conflicting manner, where users compete against each other for the activity. For example, while a user is taking shower, other users want to use the toilet, or while a user is playing computer game, another user wants to check email.

There also exist more complex situations in each of *Single-user ADLs* and *Multi-user ADLs*, for example, while a user is performing an activity, she/he may want to perform another activity in an *interleaved* (i.e., switching between steps of these two ADLs) or *concurrent* (i.e., performing steps of these two ADLs simultaneously) manner. These situations can be quite complex while concerning both *Single-user* ADLs and *Multi-user ADLs* at the same time, and hence they are out of the scope of this paper.

B. Paper Scope and Our Contributions

In this paper, we investigate the fundamental problem of multi-user activity recognition in pervasive computing environments. We attempt to study this problem from two aspects – both sensor platform and recognition framework. We first develop our sensor platform for multi-user trace collection; we then propose a novel activity model leveraging on a pattern mining technique and design our recognition algorithm to recognize activities of both single user and multiple users in a unified framework.

More specifically, we design our sensor platform capable of capturing user motion, user location, human-to-object interaction, human-to-human interaction, and user identities. Based on this platform, we conduct a real-world activity trace collection done by two volunteers over a period of two weeks in a smart home environment. The trace contains a variety of activity cases mentioned in Section 1.1 for our study. We then propose a novel activity model based on *Emerging Pattern* (EP) – a new type of knowledge pattern that describes significant changes between classes of data. Both single-user and multi-user models are built upon EPs that are mined from training instances. While both models are capable of capturing discriminative patterns, the multi-user model can especially model user interaction. Based on our activity models, we design an activity recognizer named *epMar* to recognize both *Single-user ADLs* and *Multi-user ADLS*. We conduct comprehensive experiments to evaluate our system to demonstrate its effectiveness.

In summary, the paper makes the following contributions.

- *We propose a novel activity model based on Emerging Pattern for both single-user and multi-user, and especially, the multi-user model is capable of capturing user interaction.*
- *We propose a novel activity recognizer to recognize both single-user and multi-user activities in a unified framework. The feedback loop in the recognizer is novel in the way that it is able to adjust the boundary between two adjacent activities, resulting in more accurate recognition.*
- *We design our sensor platform for multiple users, conduct a real-world trace collection consisting of a variety of activity cases, and evaluate our recognizer through comprehensive experiments.*

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we describe our sensor platform. Section 4 gives the background on Emerging Pattern, and then describes the mining of Emerging Patterns. We present our activity models and the *epMAR* recognizer in Section 5. Section 6 reports our empirical studies, and finally, Section 7 concludes the paper.

II. Related Work

Much early work in human activity recognition [8-9] has been done in the computer vision community. They leverage on video cameras, and explore various spatio-temporal analysis to recognize people's actions from video sequences.

In pervasive computing, researchers are interested in recognizing activities based on sensor readings. Typical static classifiers include naïve Bayes used in [10-12], decision trees used in [10, 12, 13], and k-nearest neighbor (k-NN) used in [10]. In temporal classification, state-space models are typically used to enable the inference of hidden states (i.e., activity labels) given the observations. We name a few examples here: Hidden Markov Model (HMM) used in [14-17], Dynamic Bayesian Network (DBN) used in [18] and Conditional Random Field (CRF) used in [21-23]. A different attempt is Time Series based classification, in which an activity is modeled as a sequence of discrete events [19]. Activities are recognized through discovering and matching the Motif which is defined as the subsequences with similar behavior appeared frequently in time-series data.

There is some existing work on recognizing multi-user or group activities in computer vision. Gong et al. [27] developed a dynamically multi-linked HMMs model to interpret group activities. They also compared their methods with Multi-Observation HMM, Parallel HMMs, and Coupled HMM. Nguyen et al. [26] employed hierarchical HMM for modeling the behavior of each person and the joint probabilistic data association filters for data association. Park et al. [24] presented a synergistic track- and body-level analysis framework for multi-person interaction and activity analysis in the context of video surveillance.

However, in pervasive computing, little work has been done in addressing the multi-user activity recognition problem based on sensor readings. Although in [12], the data collection was done by two users (i.e., a couple) in an instrumented home, the dataset contains annotated data for the male user only and the work reported in the paper focuses on this single user. An interesting work is done by Lin et al. [25], where they deployed various kinds of sensors in a home environment and proposed a layered model to learn multiple users' preferences based on sensor readings. However, they only focus on learning of preference models of multiple users,

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

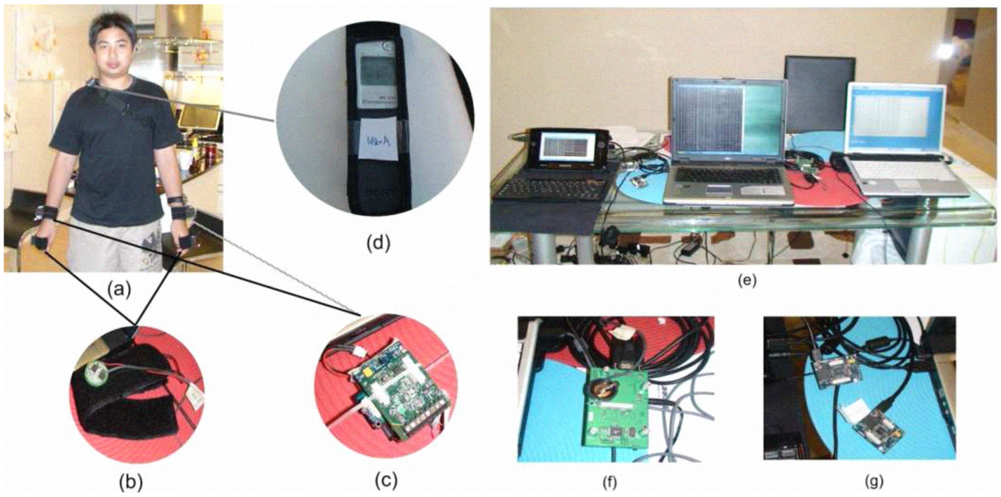

Fig. 1. (a) Wearable sensors consist of 2 RFID wristband readers, 2 iMote2 sets and 1 audio recorder, (b) RFID wristband reader, (c) iMote2 set, (d) audio recorder, (e) servers, (f) Mica2Dot module connected to a server, and (g) iMote2 modules connected to servers.

i.e., relationships among users as well as dependency between services and sensor observations, not on recognizing their activities. In this paper, we aim to recognize both single-user and multi-user activities in a unified framework. Our EPs based model uses a set of multi-attribute tests for each class, while most previous classifiers consider only one test on one attribute at a time. Different from the Time Series based classifier concerning the mining of regularities, we mine the abnormal growth among classes in our approach.

III. Our Sensor Platform

We built our sensor platform from off-the-shelf sensors. It measures user motion (i.e., both hands' movements), user location, human-object interaction (i.e., objects touched and sound), and human-to-human interaction (i.e., voice).

Fig. 1 illustrates our sensor setup. A user wears a Crossbow iMote2 set and an RFID wristband reader on each of his hands. Each iMote2 set (Fig. 1c) consists of an iMote2 IPR2400 processor and radio board and an ITS400 sensor board, capable of measuring his hand movement (3-axis accelerometer). The RFID wristband reader (Fig. 1b, similar to Intel's RFID bracelet [24]) is custom-built and it incorporates a SkyeTek M1-mini RFID reader, a Crossbow Mica2Dot wireless module and a Li-Polymer rechargeable battery. The wristband is able to detect the presence of a tagged object within the range of 6 to 8 cm. A user also wears an audio recorder (Fig. 1d) to record both human voice and environmental sound. The sampling rate of the RFID readers is set to 2 Hz, the sampling rate of 3-axis accelerometer in each iMote2 128 Hz, and the sampling rate of audio recorder 16 KHz. In addition, user location is detected in a simple way that a UHF RFID reader is located in each room to sense the proximity of a user wearing a UHF tag. To determine human identity, the device IDs of each iMote2 set and RFID reader are logged and bound to a specific user. However, it is not possible to determine audio identity precisely as the audio information may come from other users who live in the same space.

When a user performs daily activities, the acceleration readings from each iMote2 set are transmitted wirelessly to a local server (Fig. 1e, left or middle) which runs on a laptop PC with an iMote2 IPR2400 board connected through its USB port (Fig. 1g). When a user handles a tagged object, the wristband scans the tag ID and sends it wirelessly to another server (Fig. 1e, right) that can map the ID to an object name. This server runs on a Linux-based laptop PC with a MIB510CA serial interface board and a Mica2Dot module connected through its serial port (Fig. 1f). In addition, human voice and environmental sound are recorded by the audio recorder. These sensor readings are logged separately, and will be merged into a single text file as the activity trace for a single user.

IV. Mining Emerging Patterns from the Activity Trace

A. Emerging Pattern and Preliminaries

We first provide the background of EP. EP describes significant changes between two classes of data [29]. An EP is a set of items whose frequency changes significantly from one dataset to another.

Suppose that a dataset D consists of many instances. An instance contains a set of items (i.e., an itemset), where an item is an attribute-value pair. The support of an itemset X, $supp_D(X)$, is $count_D(X)/|D|$, where $count_D(X)$ is the number of instances in D containing X.

Definition 1 Given two different classes of datasets D_1 and D_2, the growth rate of an itemset X from D_1 to D_2 is defined as $GrowthRate(X) =$

$$\begin{cases} 0, & if \quad supp_1(X) = 0 \quad and \quad supp_2(X) = 0 \\ \infty, & if \quad supp_1(X) = 0 \quad and \quad supp_2(X) > 0 \\ \frac{supp_2(X)}{supp_1(X)}, & otherwise \end{cases}$$

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

EPs are those itemsets with large growth rates from D_1 to D_2.

Definition 2 Given a growth rate threshold $\rho > 1$, an itemset X is said to be a $\rho - EmergingPattern$ (or simply EP) from a background dataset D_1 to a target dataset D_2 if $GrowthRate(X) \geqslant \rho$.

An EP with high support in its target class and low support in the contrasting class can be seen as a strong signal indicating the class of a test instance containing it.

B. Data Preprocessing

Before we introduce the mining of EPs for activity recognition, we need to compute appropriate sensor features and discretize their readings.

1) Feature Extraction: We convert all the sensor readings (i.e., 3-axis acceleration, audio, location, and tagged objects) to a series of *observation vectors* by concatenating all of the data observed in a fix time interval which is set to one second in our experiments. Our feature extract process generates a 47-dimensional *observation vector* every second. Different types of sensors require different processing to compute various features.

For acceleration data, we compute the features including mean, variance, energy, frequency-domain entropy, and correlation. The DC feature is the mean acceleration value over the window. Variance is used to characterize the stability of the signal. The energy feature captures the data periodicity, and is calculated as the sum of the squared discrete FFT component magnitudes of the signal. Frequency-domain entropy helps to discriminate activities with similar energy values, and is calculated as the normalized information entropy of the discrete FFT component magnitudes of the signal. Correlation is calculated between every two axes of each accelerometer and between all pairwise combinations of axes on different accelerometers. This feature aims to find out the correlation among the different axes of the two accelerometers.

For audio data, we compute both time-domain and frequency-domain features. The time-domain features measure the temporal variation of audio signal, and consist of three features. The first one is the standard deviation of the reading over the window, normalized by the maximum reading in the window. The second one is the dynamic range defined as (max - min) / max, where min and max represent the minimum and maximum readings in the window. The third is Zero-Crossing Rate (ZCR), which measures the frequency content of the signal and is defined as the number of time-domain zero crossings in the window. In the frequency-domain, we compute two features – centroid (the midpoint of the spectral power distribution) and bandwidth (the width of the range of frequencies that the signal occupies).

For RFID reading or location information, we use object name or location name directly as features. For each RFID wristband reader, we choose the first object in a one-second window since a user is unlikely to touch two or more objects in such a short interval. If no RFID reading is observed or in the presence of a corrupted tag ID, the value will be set to NULL.

2) Discretization and Indexing: We then transform these *observation vectors* into *feature vectors*. A *feature vector* consists of many *feature items*, where a *feature item* refers to a feature name-value pair in which a feature can be numeric or nominal. We denote a numeric feature as $numfeature_i$. Suppose its range is $[x, y]$ and an interval $[a, b]$ (or in other forms, $(a, b]$, $[a, b)$, or (a, b)) is contained in $[x, y]$. We call $numfeature_i@[a, b]$ a *numeric feature item*, meaning that the value of $numfeature_i$ is limited inclusively between a and b. We denote a nominal attribute as $nomfeature_j$. Suppose its range is $\{v_1, v_2, ..., v_n\}$, we call $nomfeature_j@v_k$ a *nominal feature item*, meaning the value of $nomfeature_j$ is v_k.

The key step of transformation is to discretize numeric features. We follow the entropy-based discretization method [40], which partitions a range of continuous values into a number of disjoint intervals such that the entropy of the partition is minimal.

The discretization method partitions 44 numeric feature values into a total of 484 disjoint intervals. Then we can directly combine the feature name and its interval into a *numeric feature item*. For the nominal feature, the feature name and its value are combined as a *nominal feature item*. For the $left\_object$ and $right\_object$ features, we merge them into one feature by computing $left\_object \cup right\_object$ without losing any essential objects during the user-object interaction due to user's handedness. In our current sensor setting, we have a total of 574 *feature items*. They are indexed by a simple encoding scheme and will be used as inputs to the EPs mining process described in the next section.

C. Mining Emerging Patterns

We obtain two sets of training dataset, corresponding to the two users, and each observation is assigned with an activity label. For each set of training dataset, we first separate the instances of *Single-user ADLs* from *Multi-user ADLs*. An instance here refers to a union of all the observations that belong to an activity during a continuous period of time. Then we mine EPs of *Single-user ADLs* and *Multi-user ADLs* separately. We discover the EPs by an efficient algorithm described in [30]. The algorithm mines closed patterns and generators simultaneously under one depth-first search scheme. After computation, we get m sets of EPs, one set per activity. We refer EP_{A_i} as the EPs of A_i.

Table I presents an example set of EPs of the *brushing teeth* activity for each user. The meaning of each item in an EP is obvious as we translate each feature number into a feature name-value pair. Obviously, acceleration data, objects such as toothpaste, toothbrush and water, location (i.e., bathroom) and audio data play their roles in these two EP sets.

V. Activity Model and Recognition Algorithm

A. Problem Formulation

We formulate the multi-user activity recognition problem as follows. Given a number of training datasets corresponding

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

TABLE I
A PARTIAL SET OF THE EPS FOR THE *Brushing Teeth* ACTIVITY.

User	EPs	Support(%)	Growth rate
User 1	$\{\ \mathit{accel\_left\_x\_mean}@(-201.9765 \sim 107.0105], \mathit{accel\_correlation\_9}@(-\infty \sim 77890.1185], \mathit{location}@\mathit{bathroom}, \mathit{object}@\mathit{toothpaste}, \mathit{accel\_left\_x\_mean}@(-831.872 \sim 612.6145], \mathit{accel\_right\_z\_entropy}@(-3.2115 \sim 1.511], \mathit{audio\_centroid}@(2750.1775 \sim 3680.728], \mathit{object}@\mathit{water}, \mathit{audio\_zcr}@(0.3315 \sim 0.4815], \mathit{audio\_centroid}@(3680.728 \sim 5090.182]\ \}$	90	∞
User 2	$\{\mathit{object}@\mathit{toothbrush}, \mathit{audio\_centroid}@(2617.5925 \sim \infty], \mathit{accel\_correlation\_4}@(-\infty \sim 354687.569], \mathit{object}@\mathit{water}, \mathit{location}@\mathit{bathroom}, \mathit{audio\_zcr}@(0.42 \sim \infty], \mathit{object}@\mathit{toothpaste}\}$	90	∞

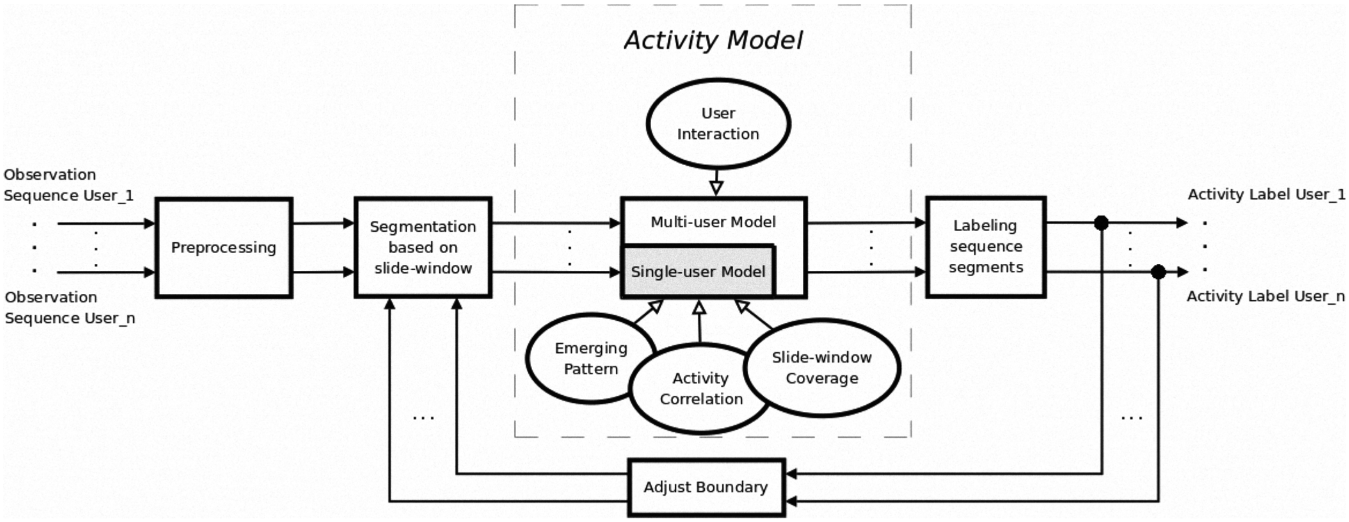

Fig. 2. Overview of the *epMAR* recognizer.

to multiple users, where each dataset consists of a sequence of observations for both *Single-user ADLs* and *Multi-user ADLs*, our objective is to train appropriate activity models and develop an activity recognizer that can assign each new observation in each dataset with the correct activity label. Formally, the training dataset O of an user consists of T observations $O = \{o_1, o_2, ... , o_T\}$ associated with activity labels $\{A_1, A_2, ... , A_m\}$, where there are m activities.

B. Overview

We give an overview of the *epMAR* activity recognition system which is capable of recognizing both *Single-user ADLs* and *Multiple-user ADLs*, as illustrated in Fig. 2. There will be one or more observation sequences corresponding to a single user or multiple users input into the *epMAR* recognizer. These observation sequences are generated by our sensor platform and will be first pre-processed into feature vectors as described in Section 4.2. The *epMAR* recognizer operates in two phases – model training and activity recognition. In the training phase, a training dataset for each of the users will be used to train our activity models. Our activity models consist of both a single-user model and a multi-user model. The single-user model is designed to recognize activities of a single user, whereas the multi-user model is capable of recognizing activities of multiple users. Both models are built upon the concept of EP which is used as a powerful discriminator to differentiate activity classes. In the recognition phase, for each user's sequence (i.e., S_t, $t = 0 \sim T$), we first segment its sequence using a slide-window (i.e., L_{A_i}) to obtain a test instance (i.e., $S_{t \sim t+L_{A_i}}$, $t = 0 \sim T$), and then apply our recognition algorithm to label this sequence segment. The above process will be performed recursively. Since each of these sequence segments corresponds to an activity label, for each pair of consecutive sequence segments, we design an algorithm to detect and adjust the boundary. This algorithm serves as a feedback loop in our system aiming to label sequence segments accurately and overcome the drawback of a slide-window based segmentation method.

In the following sections, we first describe our proposed activity models, and then present our algorithms.

C. Activity Model

1) Single-user Model: The single-user model is designed to recognize activities of a single user. It is composed of three elements: *EP score*, *Slide-window Coverage score* and *Activity-Correlation score*. The details are described as follows:

EP score: This score element provides a measurement on a fraction of EPs contained in a test instance. The more fraction of EPs we detect in a test instance, the higher EP score we obtain. As each activity class is typically associated with more

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

than one set of EPs, we aggregate the strength of each EP to sum up their contributions. We use a simple aggregation method described in [31], and the aggregated EP score of $S_{t\sim t+L_{A_i}}$ for A_i is defined as follows.

$$aggregated\_score(A_i, S_{t\sim t+L_{A_i}}) = \sum_{X\subseteq S_{t\sim t+L_{A_i}}, X\in EP_{A_i}} \frac{growth\_rate(X)}{growth\_rate(X)+1} * supp_{A_i}(X) \tag{1}$$

where $supp_{A_i}(X)$ is the support of X in A_i, and $growth\_rate(X)$ is $supp_{A_i}(X)$ divided by the X's support in non-A_i classes. Then the EP scores of each activity are "normalized" by dividing them using the median of the scores of the training instances of that activity. Finally, the *EP score* is defined as follows.

$$ep\_score(A_i, S_{t\sim t+L_{A_i}}) = \frac{aggregated\_score(A_i, S_{t\sim t+L_{A_i}})}{base\_score(A_i)} \tag{2}$$

where $base\_score(A_i)$ is the median of the values of $aggregated\_score(A_i, S_{t\sim t+L_{A_i}})$ in the training data.

Slide-window Coverage score: This score element is used to measure how many irrelevant observations contained in a test instance for a particular activity. The less irrelevant observations we detect in a test instance, a higher *Slide-window Coverage score* we obtain. We denote $coverage\_score(A_i, S_{t\sim t+L_{A_i}})$ as the *Slide-window Coverage score* of a test instance $S_{t\sim t+L_{A_i}}$ for an activity A_i. This score element is computed based on $relevance(A_i, f_p)$, where f_p is a feature vector contained in L_{A_i}. Recall that a feature vector is a set of feature items, $relevance(A_i, f_p)$ is computed based on $relevance(A_i, item_h)$ for each $item_h \in f_p$ which is defined as follows.

$$relevance(A_i, item_h) = P(item_h|A_i) \quad + \sum_{item_h\in X, X\in EP_{A_i}} supp_{A_i}(X) \tag{3}$$

We then aggregate the value of $relevance(A_i, item_h)$ for all $item_h \in f_p$, and the normalized $relevance(A_i, f_p)$ can be computed as follows.

$$relevance(A_i, f_p) = \frac{unnorm\_relevance(A_i, f_p)}{base\_relevance(A_i)} \tag{4}$$

Finally, the *coverage score* is computed by averaging all the $relevance(A_i, f_p)$ as follows.

$$coverage\_score(A_i, S_{t\sim t+L_{A_i}}) = \frac{1}{L_{A_i}} \sum_{f_p\in L_{A_i}} relevance(A_i, f_p) \tag{5}$$

Activity-Correlation score: This score element is used to measure correlations between activities. Such correlation commonly exists in our daily lives, e.g., a user usually brushes his teeth, followed by washing his face; cleans the dining table after eating a meal.

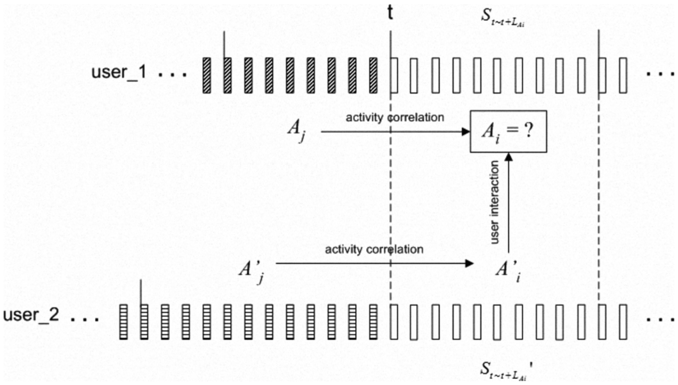

Fig. 3. Multi-user Model.

We use condition probability to model correlations between activities. We define the *activity-correlation score* of A_i as $P(A_i|A_j)$, which is the conditional probability of A_i given A_j. This score can be easily obtained from the training dataset. Note that the initial value of the conditional probability of A_i is set to zero, i.e., $P(A_i|NULL) = 0$.

Finally, we propose our single-user model using a linear combination of the above three elements. Hence, we have the following definition.

Definition 3 Given a time t and a labeled activity A_j which ends at t, for each activity A_i, a test instance $S_{t\sim t+L_{A_i}}$ is obtained from t to $t + L_{A_i}$, the score function of A_i for a single-user model is then defined as follows.

$$\begin{aligned} &single\_user\_score(A_i, A_j, S_{t\sim t+L_{A_i}}) = \\ &c_1 * ep\_score(A_i, S_{t\sim t+L_{A_i}}) \quad + \\ &c_2 * coverage\_score(A_i, S_{t\sim t+L_{A_i}}) + c_3 * P(A_i|A_j) \end{aligned}$$

where c_1, c_2 and c_3 are the coefficients, representing the importance of an individual score element. These coefficients reveal different activity and behavior patterns of a user. For example, a higher c_1 implies that a user always performs her/his activities in a consistent manner. A higher c_2 implies that an activity is performed in a constant duration whereas a lower c_2 implies that the duration variance of the activity among all the instances can be large. If c_3 is high, it implies that there exists strong correlation between activities performed by a user, i.e., activities performed always follow a certain order.

2) Multi-user Model: The multi-user model is capable of recognizing activities of multiple users. It extends our single-user model by taking user interactions into account. The training of this model requires multiple observation sequences corresponding to different users. An example of a two-user scenario is given in Fig. 3 (Note that while we use a two-user scenario in this paper, in theory, it can scale up to more than two users). Two sequences corresponding to user_1 and user_2 respectively are input into the recognizer in which each sequence is handled separately. Fig. 3 illustrates the case of detecting a multi-user activity A_i for a test instance $S_{t\sim t+L_{A_i}}$ of user_1, giving previous labels for both users (i.e., a labeled activity A_j for user_1 and another labeled activity

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

A'_j for user_2). In addition to our single-user model which captures EPs, slide-window coverage and activity correlation, our multi-user model captures user interaction. To model user interaction, we introduce *inter_score* which is defined as follows.

$$inter\_score(A_i, A'_j, S'_{t\sim t+L_{A_i}}) = \max_{A'_i}[P(A_i|A'_i) * P(A'_i|A_i) * confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}})] \quad (6)$$

where $S'_{t\sim t+L_{A_i}}$ is the sequence segment of user_2 during the time period from t to $t+L_{A_i}$, A'_i is the possible activity for $S'_{t\sim t+L_{A_i}}$, and $P(A_i|A'_i)*P(A'_i|A_i)$ is the joint probability of A_i and A'_i occurs while they are conditioning on each other. $confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}})$ measures the certainty of label A'_i, and it is defined as follows.

$$confidence(A'_i, A'_j, S'_{t\sim t+L_{A_i}}) = \frac{single\_user\_score(A'_i, A'_j, S'_{t\sim t+L_{A_i}})}{\sum_k single\_user\_score(A'_k, A'_j, S'_{t\sim t+L_{A_i}})} \quad (7)$$

Finally, we define our multi-user model as follows.

Definition 4 Given a time t, a labeled activity A_j which ends at t for user_1 and a labeled activity A'_j which ends at t for user_2, for each activity A_i of user_1, a test instance $S_{t\sim t+L_{A_i}}$ is obtained from t to $t+L_{A_i}$, $S_{t\sim t+L'_{A_i}}$ is obtained for user_2, the score function of A_i for a multi-user model is then defined as follows.

$$\begin{aligned} &multi\_user\_score(A_i, A_j, A'_j, S_{t\sim t+L_{A_i}}, S'_{t\sim t+L_{A_i}}) = \\ &\quad single\_user\_score(A_i, A_j, S_{t\sim t+L_{A_i}}) \ + \\ &\quad c_4 * inter\_score(A_i, A'_j, S'_{t\sim t+L_{A_i}}) \end{aligned} \quad (8)$$

where c_4 is the coefficient representing the importance of *inter_score*. Similar to other coefficients, a higher c_4 implies there exist more interaction between multiple users when they perform a multi-user activity.

D. Algorithm to Detect and Adjust Boundary

As we summarize in Section 5.2, we obtain a test instance using a slide-window. However, a slide-window is only an approximation of the actual duration of the instance. Any error in a segmentation may affect subsequent segmentations. To overcome this limitation, we propose an algorithm to detect and adjust the boundary of the two label activities. This algorithm is based on the following heuristics. Intuitively, given an activity instance, its feature vectors obtained by preprocessing its observations usually have a higher relevance to this activity than all other activities. Furthermore, the relevance of its feature vectors in the same activity instance does not vary significantly as compared to the relevance of two feature vectors belonging to two different activities. The algorithm is described in Algorithm 1.

Algorithm 1 The Algorithm to Detect and Adjust Boundary **-adjustBoundary**

```
Input: feature vector of length $L_{A_j} + L_{A_i}$: $F = \{f_{t-L_{A_j}}, ..., f_{t+L_{t+L_{A_i}}}\}$,
    where $t$ is the existing boundary,
    labeled activity $A_j$ followd by $A_i$.
Output: the boundary between $A_j$ and $A_i$.
foreach $p$ from $t - L_{A_j}$ to $t + L_{A_i}$ do
    $RW[p] = relevance(A_j, f_p) - relevance(A_i, f_p)$;
end for
foreach $p$ from $t - L_{A_j}$ to $t + L_{A_i}$ do
    $upperSum$ = sum of all $RWs$ from $t - L_{A_j}$ to $p$;
    $lowerSum$ = sum of all $RWs$ from $p$ to $t + L_{A_i}$;
    $GAIN[p]$ = $upperSum$ - $lowerSum$;
end for
    boundary = $p$ such that $GAIN[p]$ is maximun;
return boundary;
```

Algorithm 2 The *epMAR* Algorithm

```
Input: two observation sequences with a length of $T$ for user1
    and user2: $O^1 = \{o^1_1, o^1_2, ..., o^1_T\}$, $O^2 = \{o^2_1, o^2_2, ..., o^2_T\}$;
    $m$ activities $\{A_1, A_2, ..., A_m\}$.
Output: assign the activity label to each observation.
pre-process $O^1$ and $O^2$ to obtain feature vectors $F^1$ and
    $F^2$, where $F^1 = \{f^1_1, f^1_2, ..., f^1_T\}$, $F^2 = \{f^2_1, f^2_2, ..., f^2_T\}$;
$t^1$ = $t^2$ = 1;
while $t^1 \leqslant T$ or $t^2 \leqslant T$
  foreach user $u$ = 1, 2 do
    $A^u_{previous}$ = null; $A^u_{candidate}$ = null;
    foreach activity $A_i$ , $i = 1, 2, ..., m$ do
      get instance $S^1_{t\sim t+L_{A_i}} = \bigcup_{p=t}^{t+L_{A_i}} f^1_p$;
      get instance $S^2_{t\sim t+L_{A_i}} = \bigcup_{p=t}^{t+L_{A_i}} f^2_p$;
      compute
      $score\_with\_inter(A_i, A^u_{previous}, A^{u\_other}_{previous},$
                  $S^u_{t\sim t+L_{A_i}}, S^{u\_other}_{t\sim t+L_{A_i}})$;
    end for
    $A^u_{current} = A_i$ with the highest score;
    if $t^u = 1$ or $A^u_{current} = A^u_{candidate}$
      Assign label $A_{current}$ to $o_t \sim o_{t+L_{A_{current}}}$;
      $t^u = t^u + L_{A^u_{current}}$;
      $A^u_{previous} = A^u_{current}$;
      $A^u_{candidate} = null$;
    else if $A^u_{candidate} \neq A^u_{current}$
      $t^u = adjustBoundary(F^u_{t-L_{A^u_{previous}}, t+L_{A^u_{current}}},$
                  $A^u_{previous}, A^u_{current})$;
      $A^u_{candidate} = A^u_{current}$;
    end if
  end for
end while
```

E. The epMAR Algorithm

The entire process of our *epMAR* recognizer is described in Algorithm 2. To summarize, for each user, given m activities,

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

TABLE II
Single-user ADLs AND *Multi-user ADLs* PERFORMED.

Single-user sequential ADLs	brushing teeth	making coffee	vacuuming
	washing face	toileting	using phone
	brushing hair	ironing	using computer
	making pasta	making tea	reading book/magazine
Single-user sequential ADLs with simultaneity	watching TV w/s.	eating meal w/s.	drinking w/s.
Multi-user ADLs with collaboration	making pasta w/col.	cleaning a dinning table w/col.	making coffee w/col.
Multi-user ADLs with conflict	toileting w/conf.	watching TV w/conf.	using computer w/conf.

Fig. 4. Snapshots showing our smart home, and various activities being performed.

TABLE III
DATASETS BREAKDOWN

Activity category	**Number of instances collected**	
	User 1	**User 2**
Single-user sequential ADLs	120	120
Single-user sequential ADLs with simultaneity	30	30
Multi-user ADLs with collaboration	30	30
Multi-user ADLs with conflict	30	30

we first obtain a test instance $S_{t\sim t+L_{A_j}}$ for A_j using L_{A_j}; and we compute $score(S_{t\sim t+L_{A_j}}, A_j)$ for A_j. The one with the maximum score is assigned to $S_{t\sim t+L_{A_j}}$ with its candidate label A_j. The same process is repeated to compute another candidate label A_i. We then apply the algorithm to detect and adjust the boundary of A_j and A_i. Finally, we re-compute the score for each activity label, and the label with maximum score yields the final label for this instance. The above process then runs recursively from this boundary over the entire trace. The *epMAR* recognizer is flexible in a way that it works for both single-user and multi-user. In the case of a single user, the recognizer takes a single-user observation sequence as input and recognizes activities based on the single-user model. In the case of multiple users, the recognizer takes multiple observation sequences as input and recognizes activities based on the multi-user model. Note that the *epMAR* algorithm applies to more than two users although we show only two users for illustration in Algorithm 2.

VI. EXPERIMENTAL STUDIES

We now move to evaluate our proposed algorithm. To the best of our knowledge, there are no multi-user datasets available in the pervasive computing community. We decided to conduct our own trace collection. In this section, we first describe our evaluation methodology and experimental setup, then present and discuss the evaluation results obtained from a series of experiments.

A. Experimental Setup and Methodology

Trace collection was done by two volunteers over a period of two weeks in a smart home environment, as shown in Fig. 4 (top left). These two volunteers are final-year university students from a local university. We randomly select 21 activities from the list of common ADLs in our survey, corresponding to the four types of activities mentioned in Section 1.1, as summarized in Table II. We tagged over 100 day-to-day objects, e.g., tablespoons, cups, faucets, etc, as partially shown in Fig. 4 (top second). Each day, both users wore our sensor platform respectively, and performed these activities at their choices in their own ways based on their daily practices. The trace was then logged in the servers. All the servers and the audio recorder were synchronized before the experiments. Location information was recorded partially by hand due to the limitation of our indoor location system. All the traces were annotated by hand to establish the ground truth together with video recording. Fig. 4 shows some snapshots of various activities being performed during our experiments.

A total number of 420 annotated instances was collected for both users and the details are given in Table III. We use ten-fold cross-validation for our evaluation. We evaluate the performance of our algorithm using the time-slice accuracy which is a typical technique in time-series analysis. The time-slice accuracy represents the percentage of correctly labeled time slices. The length of time slice Δt is set to 15 seconds as our experiment shows different Δt doest not affect the accuracy much. This time-slice duration is short enough to provide precise measurements for most of activity recognition applications. The metric of the time-slice accuracy is defined as follows.

$$Accuracy = \frac{\sum_{n=1}^{N}[predicted(n) == ground\_truth(n)]}{N}, \\ where \quad N = \frac{T}{\Delta t} \tag{9}$$

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

TABLE IV
OVERALL ACCURACY BREAKDOWN IN USERS AND ADLS

User	Accuracy	
	Single-user ADLs	**Multi-user ADLs**
User 1	86.69%	95.06%
User 2	85.57%	95.71%

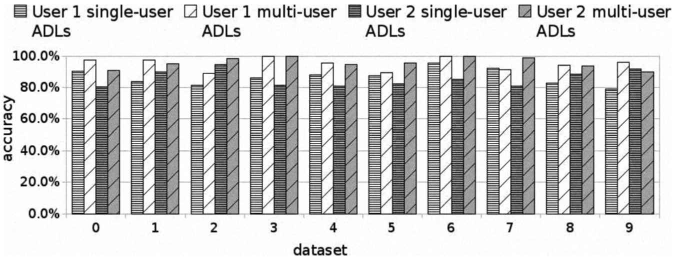

Fig. 5. Accuracy Breakdown in Users and ADLs

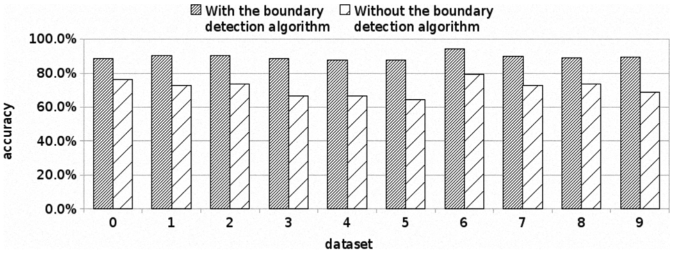

Fig. 6. Effect of the Boundary Detection Algorithm

B. Experiment 1: Performance Analysis

In the first experiment, we evaluate the overall accuracies with respect to different users and different ADLs. The results are summarized in Table IV, and the detailed breakdown in users and ADLs are shown in Fig. 5. The overall accuracy of both users for both *Single-user* and *Multi-user ADLs* achieves 89.72%, demonstrating that the *epMAR* recognizer is effective for recognizing activities in a multi-user scenario. Table IV also shows that the accuracy of *Multi-user ADLs* is higher than that of *Single-user ADLs* for both users. We analyze this phenomenon and suggest the reason as follows.

In a *Multi-user ADL*, each user usually performs only partial steps of this activity, resulting in the EPs of *Multi-user ADLs* are subsets of the EPs of *Single-user ADLs*. A test instance is likely to contain more EPs of a *Multi-user ADL* than its corresponding *Single-user ADL*, hence, the score of the *Multi-user ADL* tends to be higher. However, although the result is in favor of *Multi-user ADLs*, an error in labeling for one user will not influence others as the test data are labeled separately.

Next, we evaluate the effect of the boundary detection algorithm for all the datasets. Fig. 6 shows the results of *epMAR* with the boundary detection algorithm achieves 18.2% higher than that of without the algorithm. The result demonstrates that the boundary detection algorithm works well to re-segment and adjust the boundary of two labeled activities, and improves the accuracy significantly.

TABLE V
MINING PARAMETERS ANALYSIS

min support / *delta*	6	7	8
0	88.54%	88.12%	87.65%
1	87.83%	88.54%	89.72%
2	86.82%	88.12%	88.79%

TABLE VI
COEFFICIENTS ANALYSIS

Coefficients (c_1, c_2, c_3, c_4)	*Accuracy of Single-user ADLs*	*Accuracy of Multi-user ADLs*	*Overall Accuracy*
(1.0, 1.0, 1.0, 1.0)	84.52%	94.87%	87.75%
(1.5, 1.0, 1.0, 1.0)	83.14%	96.63%	87.43%
(1.0, 1.5, 1.0, 1.0)	86.15%	95.36%	89.72%
(1.0, 1.0, 1.5, 1.0)	84.87%	91.28%	86.07%
(1.0, 1.0, 1.0, 1.5)	85.34%	93.46%	88.76%

C. Experiment 2: Parameter Analysis

In this experiment, we first evaluate and analyze the effect of our mining parameters. There are two critical parameters – the first one is *min support*, which is the minimum support threshold and defined as an integer. The support sum of a frequent itemset in all classes must be no less than *min support*. The second parameter is *delta*, which is also defined as an integer. The support sum of an itemset in all the minority classes must be no larger than *delta*. The EPs of an activity A_i are itemsets that the support sum in all activities are no less than *min support*, and the support sum in all non-A_i activities are no larger than *delta*. Table V shows the effect of both parameters to the overall accuracy. We observe that while the parameter pair (*min support* = 8; *delta* = 1) achieves the best accuracy, other pairs achieves a similar result resulting a smooth accuracy plane. It demonstrates that our algorithm is not sensitive to the tuning of mining parameters.

We then evaluate the effect of coefficients (i.e., *c1*, *c2*, *c3*, and *c4*,). Table VI shows the accuracies using different sets of coefficient. Each coefficient represents the importance of each score (i.e., *EP*, *coverage*, *activity-correlation*, and *user interaction*) in our activity model. The table shows the coefficient set (1.0, 1.5, 1.0, 1.0) achieves the best performance, and it reveals that both subjects seem to always perform each of their activities in a constant duration. We actually use this coefficient set for all other experiments.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we study the fundamental problem of recognizing activities for multiple users from sensor readings in pervasive computing. We design our sensor platform and conduct a real-world trace collection. Our dataset contains comprehensive activity instances for a variety of *single-user ADLs* and *multi-user ADLs* cases for our study.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818

We then investigate a challenging problem of how to recognize both single-user and multi-user activities in a multi-user environment. We propose a novel activity model based on Emerging Pattern and design the *epMAR* recognizer to recognize both *Single-user ADLs* and *Multi-user ADLs*. The results demonstrate both the effectiveness and reliability of our system.

For our future work, we will further develop our sensing platform to include more sensor features and fully explore the discriminating power of EPs. Possible sensor platforms we can leverage on include [28]. We also plan to conduct real-life trace collection done by real users in a residential home environment with the availability of funding and resources. We will also look into a more complex scenario where *single-user ADLs* and *multi-user ADLs* are mixed with *interleaved* or *concurrent* activities as we mentioned in Section 1.1. We plan to leverage on our earlier work [32] on recognizing *interleaved* or *concurrent* activities of a single user for our further studies.

Acknowledgment

This work is partially supported by National 973 and 863 projects of China under the research grants 2009CB320702 and 2007AA01Z178, National Natural Science Foundation of China under the research grants NSFC 60721002 and NSFC 60736015.

References

[1] D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. *Inferring high-level behavior from low-level sensors.* In Proc. of Ubicomp 2003, pp. 73-89, Seattle, Washington, October 2003.

[2] S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, and M. W. Jaffe. Studies of Illness in the Aged. *The Index of ADL: A Standardized Measure of Biological and Psychological Function.* Journal of the American Medical Association, vol. 185, pp. 914-919, 1963.

[3] M. Pollack, L. Brown, D. Colbry, C. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. *Autominder: An intelligent cognitive orthotic system for people with memory impairment.* Robotics and Autonomous Systems, 44:273282, 2003.

[4] V. Kaptelinin and B. A. Nardi. *Acting with Technology: Activity Theory and Interaction Design.* Cambridge, Massachusetts: The MIT Press, 2006.

[5] E. R. Smith and D. M. Mackie, *Social Psychology*, 2nd ed. London: Routledge, 1999.

[6] R. Wilensky. *Planning and Understanding: A Computational Approach to Human Reasoning.* Addison-Wesley Publishing Co., 1983.

[7] P. Dourish. *Seeking a Foundation for Context-Aware Computing.* Human-Computer Interaction, vol. 16, pp. 229-241, 2001.

[8] Yaser Yacoob and Michael J Black. *Parameterized modeling and recognition of activities.* In Proc. of International Conference on Computer Vision (ICCV'98),1998.

[9] D. Moore, I. Essa, and M. Hayes. *Exploiting human actions and object context for recognition tasks.* In Proc. of International Conference on Computer Vision (ICCV'99), 1999.

[10] L. Bao and S. S. Intille. *Activity Recognition from User-Annotated Acceleration Data.* In Proc. of PERVASIVE 2004, Vienna, Austria, April 2004.

[11] E. M. Tapia, S. S. Intille, and Larson, K. *Activity Recognition in the Home using Simple and Ubiquitous Sensors.* In Proc. of PERVASIVE 2004, pp. 158-175, Vienna Austria, April 2004.

[12] B. Logan, J. Healey, M. Philipose, E. Munguia-Tapia, and S. Intille. *A Long-Term Evaluation of Sensing Modalities for Activity Recognition.* In Proc. of Ubicomp 2007.Austria, Sept. 2007.

[13] Y. Nakauchi, K. Noguchi, P. Somwong, and T. Matsubara. *Human Intention Detection and Activity Support System for Ubiquitous Sensor Room.* JSME Journal of Robotics and Mechatronics, 16(5), pp.545-551, 2004.

[14] D. Patterson, D. Fox, H. Kautz, M. Philipose. *Fine-Grained Activity Recognition by Aggregating Abstract Object Usage.* In Proc. of ISWC 2005, Osaka, October 2005.

[15] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. *A practical approach to recognizing physical activities.* In Proc. of PERVASIVE 2006, Dublin, Ireland, October 2006.

[16] J. A. Ward, P. Lukowicz, G. Troester, and T. Starner. *Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers.* IEEE Trans. Pattern Analysis and Machine Intelligence, 28(10): 1553-1567, October 2006.

[17] D. Wyatt, M. Philipose and T. Choudhury. *Unsupervised Activity Recognition Using Automatically Mined Common Sense.* In Proc. of AAAI 2005, Pittsburgh, July 2005.

[18] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hhnel. *Inferring Activities from Interactions with Objects.* IEEE Pervasive Computing, v3 no. 4, pp. 50-57, Oct 2004.

[19] R. Hamid, S. Maddi, A. Bobick, I. Essa. *Unsupervised analysis of activity sequences using event Motifs.* In Proc. of 4th ACM International Workshop on Video Surveillance and Sensor, 2006.

[20] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell. *A Novel Sequence Representation for Unsupervised Analysis of Human Activities.* Artificial Intelligence Journal, 2008, in press.

[21] D. L. Vail, M. M. Veloso, and J. D. Lafferty. *Conditional random fields for activity recognition.* In Proc. of International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2007.

[22] Derek Hao Hu, Sinno Jialin Pan, Vincent Wenchen Zheng, Nathan Nan Liu, Qiang Yang.*Real World Activity Recognition with Multiple Goals.* InProc. of the Tenth International Conference on Ubiquitous Computing(Ubicomp 2008), Seoul, Korea, September 2008.

[23] T.L.M. van Kasteren, A. K. Noulas, G. Englebienne and B.J.A. Krse. *Accurate Activity Recognition in a Home Setting.* InProc. of International Conference on Ubiquitous Computing (Ubicomp'08). Korea, September 2008.

[24] Sangho Park, and Mohan M. Trivedi. *Multi-person Interaction and Activity Analysis: A Synergistic Track- and Body- Level Analysis Framework.* Machine Vision and Applications: Special Issue on Novel Concepts and Challenges for the Generation of Video Surveillance Systems, 18(3), pp. 151-166, Springer Verlag Inc., 2007.

[25] Zhi-Hao Linand Li-Chen Fu. *Multi-user Preference Model and Service Provision in a Smart Home Environment.* In theProc. of IEEE International Conference on Automation Science and Engineering (CASE 2007). Sept. 2007.

[26] N. Nguyen, H. Bui and S. Venkatesh.*Recognising Behaviour of multpile people with hierarchical probabilistic and statistical data association.* In Proc of the 17th British Machine Vision Conference(BMVC 2006), Edinburgh, Scotland, September 2006.

[27] S. Gong and T. Xiang. *Recognition of group activities using dynamic probabilistic networks.* In Proc.of ICCV 2003, pages 742-749, Nice, France, October 2003.

[28] Tanzeem Choudhury, et al. *The Mobile Sensing Platform: An Embedded System for Capturing and Recognizing Human Activities.* IEEE Pervasive Computing, vol. 7, no. 2, pp. 32-41, April-June 2008.

[29] Guozhu Dong and Jinyan Li. *Efficient mining of emerging patterns: discovering trends and differences.* In Proc. 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD'99), pages 4352, San Diego, CA, USA, Aug 1999.

[30] Jinyan Li, Guimei Liu and Limsoon Wong. *Mining statistically important equivalence classes and delta-discriminative emerging patterns.* In the Proceedings of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007), pages 430-439, San Jose, 2007.

[31] G. Dong, X. Zhang, L. Wong, and J. Li. *CAEP: Classification by Aggregating Emerging Patterns.* In DS'99 (LNCS 1721), Japan, Dec. 1999.

[32] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. *epSICAR: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition.* In Proc. of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (Percom '09), Galveston, Texas, March 9-13, 2009.

[33] M. Brand, and N. Oliver, and A. Pentland. *Coupled Hidden Markov Models for Complex Action Recognition.* In IEEE computer society conference on computer vision and pattern recognition, 1997.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6818
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6818