Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

An unsupervised approach to activity recognition and segmentation based on object-use fingerprints

Tao Gu [a,*], Shaxun Chen [b], Xianping Tao [c], Jian Lu [c]

[a] University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
[b] UC Davis, One Shields Avenue, Davis, CA 95616, USA
[c] Nanjing University, 22 Hankou Road MMW Building, Nanjing, Jiangsu Province, China

ARTICLE INFO

Article history:
Received 7 October 2008
Received in revised form 21 January 2010
Accepted 21 January 2010
Available online 1 February 2010

Keywords:
Human activity recognition
Activity trace segmentation
Contrast patterns
Emerging patterns
Fingerprint
Object-use
Web mining
RFID

ABSTRACT

Human activity recognition is an important task which has many potential applications. In recent years, researchers from pervasive computing are interested in deploying on-body sensors to collect observations and applying machine learning techniques to model and recognize activities. Supervised machine learning techniques typically require an appropriate training process in which training data need to be labeled manually. In this paper, we propose an unsupervised approach based on object-use fingerprints to recognize activities without human labeling. We show how to build our activity models based on object-use fingerprints, which are sets of contrast patterns describing significant differences of object use between any two activity classes. We then propose a fingerprint-based algorithm to recognize activities. We also propose two heuristic algorithms based on object relevance to segment a trace and detect the boundary of any two adjacent activities. We develop a wearable RFID system and conduct a real-world trace collection done by seven volunteers in a smart home over a period of 2 weeks. We conduct comprehensive experimental evaluations and comparison study. The results show that our recognition algorithm achieves a precision of 91.4% and a recall 92.8%, and the segmentation algorithm achieves an accuracy of 93.1% on the dataset we collected.

1. 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, has become an important research direction in pervasive computing. In this paradigm, various sensors are typically deployed to collect a sequence of observations (i.e., time-series data), 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 healthcare and context-aware applications. A typical healthcare application is monitoring activities of daily living (ADLs) [1] for the elderly and cognitively impaired people and providing them with proactive assistance. Other application can be activity-based adaptation such as lowering TV volume when a user answers a phone call or providing instructions when a user operates unfamiliar appliances.

Recognizing activities based on sensor readings is challenging because sensor data are inherently noisy and human activities are usually performed in a non-deterministic fashion. In addition, we argue that we need to address more challenges before deploying any activity recognition system for real-life use. These challenges can be, for example, *scalability* – scale to a large number of activities, *applicability* – work efficiently in real-life situations, and*adaptability* – adapt to different users

[*] Corresponding author.
E-mail addresses: gu@imada.sdu.dk (T. Gu), sxch@ucdavis.edu (S. Chen), txp@ics.nju.edu.cn (X. Tao), lj@nju.edu.cn (J. Lu).

0169-023X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2010.01.004

since different individuals may perform the same activity in different ways. A key issue is to develop appropriate activity models that map low-level sensor features to high-level concepts. To obtain activity models, a straightforward method is to learn from a set of training data. Most existing approaches [2–11,22–29,31–33] leverage on supervised learning techniques to construct their activity models; however, learning from training data typically requires human labeling. Application developers are required to label both the underlying sensor system and the activities associated with a set of training data. Considering a large number of activities to be recognized in our daily lives, manual labeling of training data may place a significant burden to any individual involved in data collection. Hence, supervised learning approaches may have limitations in real-life deployment where scalability, applicability and adaptability are highly concerned.

In this paper, we aim to propose a simple, unsupervised approach based on RFID-tagged objects to recognize human activities without manual labeling, making our effort towards scalability, applicability and adaptability for real-life deployment. We build our activity model based on fingerprints, which are sets of object-use based contrast patterns describing significant differences of object-use between any two activity classes. This is done by first mining a set of object terms for each activity class from the web, and then mining contrast patterns among object terms based on emerging patterns [35] to contrast any two activity classes. We then propose a fingerprint-based algorithm to recognize activities. We also propose two heuristic algorithms to segment an activity trace and detect the boundary of any two adjacent activities. We evaluate the effectiveness of our proposed techniques using a real-world dataset collected by seven subjects performing 17 activities involving 132 objects in a smart home environment. The experimental and comparison results show both efficiency and robustness of our proposed system.

In summary, this paper makes the following contributions:

- We propose an activity model based on object-use fingerprints which are obtained by combining mining object-use from the web and mining contrast patterns, and a fingerprint-based algorithm to recognize activities.
- We dissociate the recognition process from the trace segmentation process, and propose two algorithms based on object-use relevance to segment an activity trace.
- We validate our algorithms using a real-world dataset collected in a smart home environment, and analyze their effectiveness and limitations through comprehensive experimental and comparison studies.

The rest of this paper is organized as follows. We discuss related work in Section 2. We describe our activity model in Section 3. We then propose our activity recognition and segmentation algorithms in Section 4, and evaluate them in Section 5. Finally, we conclude the work in Section 6.

2. Related work

Much early work in human activity recognition [25–28] has been done in the computer vision community. They leverage on video cameras, and explore various tracking methods and spatio-temporal analysis to track moving objects and recognize people's actions from video sequences.

Researchers in pervasive computing are interested in recognizing activities using on-body sensors that directly measure human, the environment and human–object interaction. Most existing work focuses on applying supervised machine learning techniques to this task where manual labeling of training data is typically required. There are two major models for recognizing human activities from artificial intelligence: logic-based approach and probabilistic approach. Early approaches such as [36] were based on logic. In this model, activities are described as a logical inference process of circumscription, and represented by a set of first-order statements called event hierarchy. However, logic-based approaches have limitations in distinguishing among consistent plans and have problems to handle uncertainty and noise in sensor data. Probabilistic models gain more popularity as sensor readings are usually noisy and human activities are typically performed in a non-deterministic fashion. Probabilistic models can be generally categorized into static classification or temporal classification. In static classification, a variety of features is first extracted from sensor readings, and then a static classifier is applied to classify different activities. Typical static classifiers include naïve Bayes used in [2,3,11], decision tree used in [2,11], k-nearest neighbor (k-NN) used in [2,8,11], and support vector machine (SVM) used in [8]. In temporal classification, state-space models are typically used to enable the inference of hidden states (e.g., activity labels) given the observations (i.e., sensor readings). We name a few examples here: hidden Markov models (HMMs) used in [9,10,13,14,30,31], dynamic Bayesian networks (DBNs) used in [4] and conditional random fields (CRFs) used in [31–33]. Specifically, Bao et al. [2] proposed to use multiple accelerometers placed in multiple locations of the human body to detect activities. The authors applied pre-trained classifiers such as decision tree and Naïve Bayes to recognize activities. Ward et al. [10] proposed to use microphones and 3-axis accelerometers to recognize continuous activities to assemble tasks in a “wood workshop”. They applied linear discriminant analysis (LDA) on the sound segments and HMM on the acceleration data. Tapia et al. [3] proposed an activity recognition system based on a set of small and simple sensors. The Naïve Bayesian classifier was chosen to predict the activity labels. Philipose et al. [4] proposed to use RFID tags attached on objects of interest and represented activities as a probabilistic *object sequence*. These activity models were then converted into DBNs to compute the probabilities of the activities. Lester et al. [29] used the multi-modal sensor board and applied both static classifier and HMM classifier to recognize activities. A variant of CRF (i.e., skip chain CRF [33]) and a variant of HMM (i.e., interleaved mixture of HMMs [37]) can be used to recognize interleaving activities, and another variant of CRF (i.e., factorial CRF [32]) can be used to recognize concurrent activities. However, supervised learning models typically require labeled training data for an appropriate training process.

Recent work shows an interesting direction towards an unsupervised approach to activity recognition. Perkowitz et al. [12] formulated activity models by translating labeled activities into probabilistic collections of terms, and mined the definitions of activities from the web. They represented the activities as HMM-based on the *object sequence* used with probabilistic distributions. However, activity traces were manually segmented and input to their HMM models, and hand segmentation still requires involvement of human effort. In addition, it may cause the data belonging to the same activity being segmented into different slices or the data two adjacent activities falls in the same slice. Wyatt et al. [13] built their activity models by mining the web in a wider scope. They presented a bootstrap method that can produce labeled segmentations automatically. Their model abstraction uses a large number of labeled web pages as the training set in which human efforts are involved. The segmentation in this work was based on the duration of an activity that may vary greatly from one user to another. Therefore, the accuracy may drop obviously when their activity models were applied to real-world scenarios. Furthermore, although they used the generic mined models to segment the trace into labeled instances of activities, their segmentation process is sequential in nature such that any error in one segment may affect the segmentations of the subsequent traces. Pentney et al. [14] demonstrated the use of a large number of hand-entered common sense database to interpret activity traces. Wang et al. [30] combined a generative common sense model of activity with a discriminative model of actions to automate feature selection. Pentney et al. [21] proposed chain graphs to represent objects used and activities performed. The approach is based on combining relational databases of large common sense created by the user with techniques for information retrieval on web. Hamid et al. modeled activity as a sequence of discrete events [34], recognition is done by discovering and matching the Motif which is defined as the subsequences with similar behavior appeared frequently in time-series data. They also proposed to represent activities as bags of *n*-grams to extract global structure information of activities and presented a computational framework for unsupervised activity discovery and classification [24].

As we discussed, existing unsupervised approaches [12,13] mine both object and *object sequence* using web mining to obtain activity models. However, different users usually perform their activities in their own ways, and even the same user may perform the same activity in several different ways. Under a real-world situation, it is difficult to obtain complete HMM-based models. Most likely, this has to be complemented by hand specification. Our work in this paper is motivated by the recent advancement of unsupervised approaches. We build a simply activity model based on object-use fingerprints. Compared to [12,13], we only mine object terms from the web. Mining sequential data has been well studied in the data mining literature [38–41]. In this paper, we apply contrast pattern mining to sensor-based human activity recognition and mine contrast patterns (i.e., fingerprints) from object terms for each activity to maximize the discriminating power of fingerprints. Furthermore, our trace segmentation algorithms operate independently such that the segmentation process for one segment does not affect that for subsequent segments. This approach is in contrast to sequential segmentation commonly employed in previous research where an error in a segmentation process affects the subsequent ones.

3. Mining object-use contrast patterns for fingerprints

This section describes our fingerprint-based activity models. The problem of activity recognition based on object-use can be formulated as follows. Given an activity trace consisting of a sequence of observations (i.e., object-use) and assume there are p activities, our objective is to build a model that can assign each observation with the correct activity label and segment the trace. To obtain a set of fingerprints for each activity class, we first mine a set of object terms for each activity class from the web. We then mine a set of contrast patterns for each class which will be used as fingerprints for the subsequent recognition process.

3.1. Mining object terms and relevance weights from the web

In the first step, given a set of activities $A = \{a_1, a_2, \ldots, a_m\}$, we mine a set of object terms $T = \{t_1, t_2, \ldots, t_p\}$ involved in each activity $a \in A$, together with their associated weights W from the web. First, we obtain a set of object terms $T = \{t_1, t_2, \ldots, t_p\}$ used for defining each activity from a set of instructions on the web. Similar to [12,13], we extract relevant instructions for an activity on two websites: www.wikihow.com and www.ehow.com. Both websites provide comprehensive instructions for many day-to-day activities such as *make tea*, *brush teeth* and *take pills*.

Fig. 1 shows a brief description of the processes involved in extracting object terms for an activity. Firstly, a collection of html documents is transformed into plain text and stemmed using Porter's stemming algorithm [15]. In stemming, morphological variants of terms (e.g., singular vs. plural) which have similar semantic interpretations are considered equivalent and reduced to their stemmed or root forms. Secondly, the number of relevant object terms is further reduced by removing object terms appearing in the stoplist [15–19] (e.g., verbs, adjectives, pronouns, adverbs, false nouns, etc.). Finally, only object terms with a mapping in the physical space are retained. The database of these objects can be easily built by extracting information from a server that stored the entire object IDs.

Next, we identify the relevance weight for each object term. Apparently, there are many object terms appearing in an activity. By observing that the occurrence frequency of an object in a particular instruction closely parallels the weight of

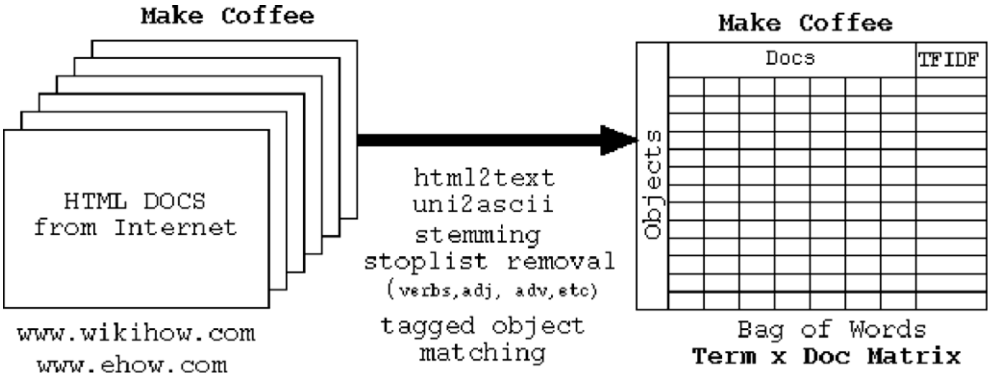


Fig. 1. Mining object terms from the web.

the object in real usage, we determine the weight W of each object term $t \in a$ by computing its term frequency – inverse document frequency (*tf–idf*) [20,21], shown as follows:

$$\texttt{tf-idf}\,{}^{a}_{i} := |t_i^a| \times \log \frac{|D^a|}{|d^a : t_i^a \in d^a|} \tag{1}$$

where $tf-idf_i^a$ is the *tf–idf* of ith object term in a; $|t_i^a|$ is the term frequency of the ith object term in a; $|D^a|$ is the total number of documents in a; $|d^a : t_i^a \in d^a|$ is the number of documents where the ith object term appears ($|t_i^a| \neq 0$).

Eq. (1) implies that if an object term is too common, it occurs in almost all documents and will have a very low *tf–idf* score. The simple factor, $\log\frac{|D|}{|d:t_i\in d|}$, denotes words if they occur in almost all documents (too general terms) and promotes words that occur in a limited number of documents (specific terms).

Since weight computation varies significantly from one activity to another because of heterogeneous sources, to establish a common basis of comparison among relative object term weights in different activities, we define our normalization technique as follows:

$$W_i^a := \frac{\log(tf - idf_i^a)}{\max\limits_{j=1}^{n^a}\left\{\log\left(tf - idf_j^a\right)\right\}} \tag{2}$$

where W_i^a is the ith object term's normalized weight in a; $tf-idf_i^a$ is the *tf–idf* of ith object term in a; n^a is the total number of object terms in a.

This normalization ensures that the topmost object term has 1.0 weight and the relative distances of succeeding object terms based on their weights do not have high variability due to the smoothing effect of the log transformation. Moreover, this transformation lessens the strong bias in weights of the topmost object terms.

We have mined 45 activity models, and Table 1 lists a partial set of object terms and relevance weights for four activities.

3.2. Mining contrast patterns from object terms

Based on object terms and their relevance weights we mined, we can build a simple model which leverages a set of top n object terms as fingerprints for each activity. However, in real-life, activities may share common objects. To maximize the discriminative power of fingerprints for each activity class, we mine a set of contrast patterns from object terms and their

Table 1
A partial view of object terms and their relevance weights.

Make tea		Make coffee		Make pasta		Fry egg	
Object	Weight	Object	Weight	Object	Weight	Object	Weight
tea	1.00	coffee	1.00	pasta	1.00	egg	1.00
water	0.85	water	0.86	flour	0.88	pan	0.99
cup	0.83	cup	0.85	pepper	0.85	oil	0.78
sugar	0.75	pot	0.82	water	0.84	burner	0.76
teapot	0.75	grinder	0.80	sauce	0.83	spatula	0.69
pot	0.74	filter	0.79	tomato	0.81	lid	0.66
bowl	0.72	sugar	0.76	cheese	0.80	water	0.65
lemon	0.70	coffeemaker	0.73	garlic	0.80	bowl	0.60
kettle	0.70	creamer	0.72	oil	0.80	butter	0.60
microwave	0.67	tablespoon	0.72	pot	0.79	dish	0.60

Note: All weights are normalized using the log smoothing function.

Table 2
A subset of CPs for the *make tea* activity.

CPs	Support (%)	Growth rate
tea, teapot	95.0	∞
tea, water, cup	70.6	341.2
water, kettle, pot, cup,	36.5	130.5

relevance weights for each class, which describe the significant differences of object terms between any two activity classes. The mining results will be used as fingerprints for activity recognition.

The definition of our contrast pattern is motivated by the concept of emerging pattern which was first introduced in [35]. An emerging pattern is a set of items whose frequency changes significantly from one dataset to another; it describes significant changes (differences or trends) between two classes of data. Unlike an emerging pattern where each item has the same weight, in our context, items have different weights and each item is associated with a weight value (i.e., a probability). While other forms of contrast pattern exist, e.g., minimal distinguishing subsequence patterns [42], we leverage the basis form. We define our contrast pattern as follows. Suppose that a dataset D associated with a class C. D consists of many instances, and an instance contains a set of items (i.e., an itemset X), where an item is an attribute-value pair and each item x is associated with a weight W_x^C. We denote the set of all items in D as T.

Definition 1. The support of an itemset X, where $X = \{x_1, x_2, \ldots, x_m\}$ and $X \subseteq T$, is defined as

$$supp_D(X) := \frac{count_D(X)}{|D|} \times W_X^C \tag{3}$$

where $count_D(X)$ is the number of instances in D containing X, and W_X^C is the aggregated weight of X in C, which is defined as $\prod_{x=1}^{m} W_x^C$.

Definition 2. Given two different classes C_1 and C_2, the growth rate of an itemset X from C_1 to C_2 is defined as $GrowthRate(X)$:=

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

Definition 3. Given a growth rate threshold $\rho > 1$, an itemset X is said to be a *contrast pattern* from a background class C_1 to a target class C_2 if $GrowthRate(X) \geqslant \rho$.

Definitions 2 and 3 are similar to the definitions in emerging patterns. Contrast patterns are those itemsets with large growth rates from C_1 to C_2. A contrast pattern 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.

We then mine contrast patterns for each of our activity classes. For each activity, denoted as a_i, we mine a set of contrast patterns to contrast its instances,D_{a_i}, against all other activity instances D'_{a_i} , where $D'_{a_i} = D - D_{a_i}$ and D is the entire dataset which is the entire object terms mined from the web in our previous step. We refer CP_{a_i} as the contrast patterns of $a_{i.}$.

We develop an algorithm to discover contrast patterns based on an emerging patterns mining algorithm described in [20], which mines closed patterns and generators simultaneously under one depth-first search scheme. After computation, we get m sets of CPs, one set per activity class, named fingerprints for this class. Table 2 presents an example of the CPs of the *make tea* activity. Column 1 shows the CPs. For example, the CP {tea, teapot} has a support of 95% and a growth rate of ∞.

4. Activity recognition and segmentation algorithms

We can now apply fingerprints we obtained to recognize activities.

4.1. Recognizing activities using fingerprints

To recognize activities using fingerprints, given a sequence of observations (i.e., $S_{0\sim T}$), we first segment its sequence using a sliding window (i.e., L_{a_i}) to obtain a test instance (i.e., $S_{t\sim t+L_{a_i}}$), and then test this instance against a score function for each possible activity. The score function basically combines the strength of each CP subset. In such combination, CP subsets are aggregated in a single function to maximize their discriminative power to achieve good performance. The sliding window L_{a_i} is the average length of all the instances of activity a_i. The activity yielding the higher score wins out and its class label will be assigned to the test instance. The above recognition 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 the recognition process aiming to label sequence segments accurately and overcome the drawback of a sliding window based our segmentation method. The score function is defined as follows.

Definition 4. Given a test instance $S_{t\sim t+L_{a_i}}$, the score function $f(a_i, S_{t\sim t+L_{a_i}})$ for activity a_i is defined as

$$f(a_i, S_{t\sim t+L_{a_i}}) := \sum_{X \subseteq S_{t\sim t+L_{a_i}}, X \in CP_{a_i}} \frac{growth\_rate(X)}{growth\_rate(X)+1} \times supp_{a_i}(X) \tag{4}$$

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.

Using contrast patterns as fingerprints, we are able to discriminate activities since contrast patterns represent the differences between activity classes. In real-life, some activities may share similar objects, however, the weight of each object appeared in each activity will unlikely be the same. Hence, by mining contrast patterns, we are able to obtain different CP subsets with different supports and growth rates for each of these activities, and aggregate each CP subsets for classification.

Algorithm 1 describes the steps involved in the activity recognition algorithm.

Algorithm 1. Fingerprint-based activity recognition algorithm

```
Input: A sequence of objects from
  $t = 0 \sim T : O = \{o_1, o_2, o_3, \ldots, o_T\}$; Activities:
  $A = \{a_1, a_2, a_3, \ldots, a_m\}$; Prediction starts at time $t$,
Output: the activity label starts from time $t$
    foreach activity $a_i, i = 1, 2, \ldots, m$ do
        get instance $S_{t\sim t+L_{a_i}} = \bigcup_{j=t}^{t+L_{a_i}} o_j$;
        compute $f(a_i, S_{t\sim t+L_{a_i}})$;
    endfor
        return $a_i$ with the highest $f$;
```

4.2. Activity trace segmentation

In Algorithm 1, we use a sliding window to get a test instance for each possible activity. Since the sliding window length of each activity is an approximation of the actual length, the segmentation may not be accurate. Moreover, any error in one segment may affect the recognition of the subsequent trace. This error may accumulate and affect the recognition accuracy seriously. This problem is referred to as the activity boundary detection problem or trace segmentation problem.

To address this problem, we propose two heuristic algorithms leveraging on object relevance weight. We have two observations. First, an object will have a higher weight value in an activity if it is important to this activity, and the same object will have a lower weight value in an activity if it has less or no relevance to this activity. Second, the weights of adjacent objects in the same activity do not vary significantly compared to the weights of two adjacent objects belonging to two different activities.

We first introduce the MaxGap algorithm. To detect the boundary between two adjacent activities, a_x and a_y, the MaxGap algorithm computes the difference between the weight of each object in activity a_x and its weight in activity a_y (*RW: Relative Weight)*. If the object is more relevant to a_x than a_y, then *RW* will be positive while the reverse will be negative. We then compute the difference of each consecutive *RW* pairs (*gap*), and the maximum*gap* is the boundary for these two activities. Algorithm 2 outlines the MaxGap algorithm. The input of the algorithm is a sequence of objects and two predicted activities. The output of the algorithm is the timestamp of an object where we should segment the two activities. The complexity of MaxGap is linear.

Algorithm 2. The MaxGap algorithm

```
Input: A sequence of objects from $t = 0 \sim T : O = \{o_1, o_2, o_3, \ldots, o_T\}$; Activities: $A = \{a_1, a_2, a_3, \ldots, a_m\}$; Predicted
  activities $a_x$ and $a_y$.
Output: the boundary between $a_x$ and $a_y$.
    foreach (x, y) $\in a_x, a_y$ do
      for ctr = x to y do
        $RW_{ctr} = W^x(o_{ctr}) - W^y(o_{ctr})$;
      for ctr = x to y-1 do
      $GAP_{ctr} = RW_{ctr} - RW_{ctr+1}$;
    endfor
        return the boundary $GAP_{ctr}$ such that is maximum;
```

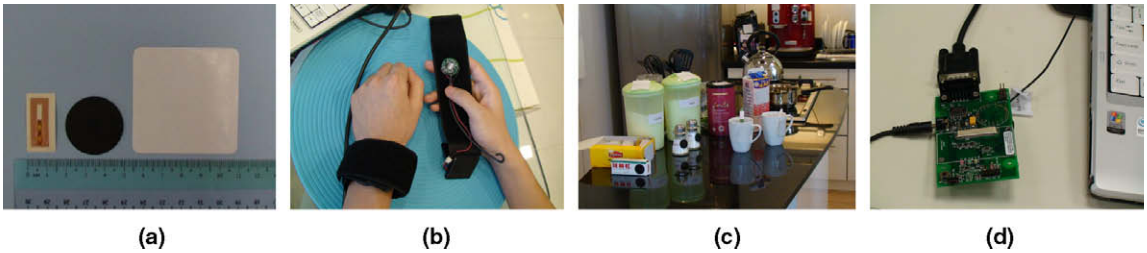


Fig. 2. (a) RFID tags, (b) RFID wristband readers, (c) RFID-tagged objects, and (d) A Linux-based server with a programming interface board and a Mica2Dot wireless module.

In cases where the two adjacent activities share common objects, boundary detection will be complicated if the common objects are located nearby the boundary. This is because their *RW*s will be close to zero. The MaxGap algorithm may fail to work in this case. To address this issue, we propose the MaxGain algorithm outlined in Algorithm 3. The input is the same as that of the MaxGap algorithm.

Algorithm 3. The MaxGain algorithm

```
foreach $(x, y) \in a_x, a_y$ do
  for $ctr = x$ to $y$ do
    $RW_{ctr} = W^x(o_{ctr}) - W^y(o_{ctr})$;
  for $ctr = x$ to $y$ do
  upperSum = 0; lowerSum = 0;
  for $upper = x$ to $ctr$ do
    upperSum $+= RW_{upper}$;
  for $lower = ctr + 1$ to $y$ do
    lowerSum $+= RW_{lower}$;
 $GAIN_{ctr}$ = upperSum − lowerSum;
 endfor
 return the boundary such that $GAIN_{ctr}$ is maximum;
```

We walk-through the MaxGain algorithm using an example shown in Fig. 2. The first column is a segment of objects extracted from an activity trace. For each possible object x_i (a candidate boundary), we compute the *upperSum* and *lowerSum* as shown in column 4 and 5, respectively. The *upperSum* is the sum of all *RW*s from *coffee* to x_i, while the *lowerSum* is the sum of all *RW*s from x_{i+1} to *tea*. We then compute the *Gain* for each candidate boundary, where *Gain* is defined as:

$$Gain_i = upperSum_i - lowerSum_i \quad \text{for } i \in \{coffee, salt, \ldots, tea\}$$

The result is shown in the last column. The timestamp of the object with the maximum *Gain* value is the boundary. In the example shown in Table 3, the *creamer* object yields the maximum Gain of 5.28 which also indicates the location of the boundary in the ground truth.

Compared to MaxGap, MaxGain considers the interplay of the group of objects between two adjacent activities, while the former algorithm only makes use of the relationship between two adjacent objects. Intuitively, MaxGain tends to be more accurate and noise-tolerant, since it is "globally optimized". When calculating the *upperSum* (or *lowerSum*) of each candidate boundary x_i, we can simply add (or subtract) the *RW* of x_{i-1}. Therefore, we conclude the complexity of *upperSum* (or *lowerSum*) is $O(n)$.

5. Experimental studies

We now move to evaluate our proposed algorithms. We develop a wearable RFID system as shown in Fig. 3. The RFID wristband reader (Fig. 3b) incorporates a SkyeTek M1-mini RFID reader, a Crossbow Mica2Dot wireless sensor module, and a rechargeable battery. It is able to detect the presence of a tag within the range of about 7 cm. The design of our RFID wristband reader is similar to Intel's iBracelet [4]. RFID tags are attached to day-to-day objects such as cup, teaspoon and book in a smart home. We have three types of tags (Fig. 3a) and all operate on 13.56 MHz. In the case of metal objects, e.g., kettle, we tagged on its plastic handle. In the case of liquid objects, e.g., water, we tagged on the faucet with a special plastic handle to sense the use. Fig. 3c shows a screen shot of tagged objects in the kitchen. When the user handles a tagged object, the RFID reader scans the tag ID and sends it wirelessly to a server (Fig. 3d) that maps the ID to an object name based on a pre-defined table. The server runs on a Linux-based laptop PC with a programming interface board and a Mica2Dot

Table 3
A walk-through example for the MaxGain algorithm.

Object	W^{coffee}	W^{tea}	RW	UpperSum	LowerSum	Gain
coffee	1.00	0.00	1.00			
salt	0.72	0.55	0.17	1.00	−2.04	3.04
water	0.86	0.85	0.01	1.17	−2.21	3.38
coffeemaker	0.73	0.00	0.73	1.18	−2.22	3.4
cup	0.85	0.83	0.02	1.91	−2.95	4.86
sweetener	0.57	0.61	−0.04	1.93	−2.97	4.9
creamer	0.72	0.49	0.23	1.89	−2.93	4.82
kettle	0.00	0.70	−0.70	2.12	−3.16	**5.28**
water	0.86	0.85	0.01	1.42	−2.46	3.88
cup	0.85	0.83	0.02	1.43	−2.47	3.9
teapot	0.00	0.75	−0.75	1.45	−2.49	3.94
water	0.86	0.85	0.01	0.70	−1.74	2.44
teapot	0.00	0.75	−0.75	0.71	−1.75	2.46
tea	0.00	1.00	−1.00	−0.04	−1.00	0.96

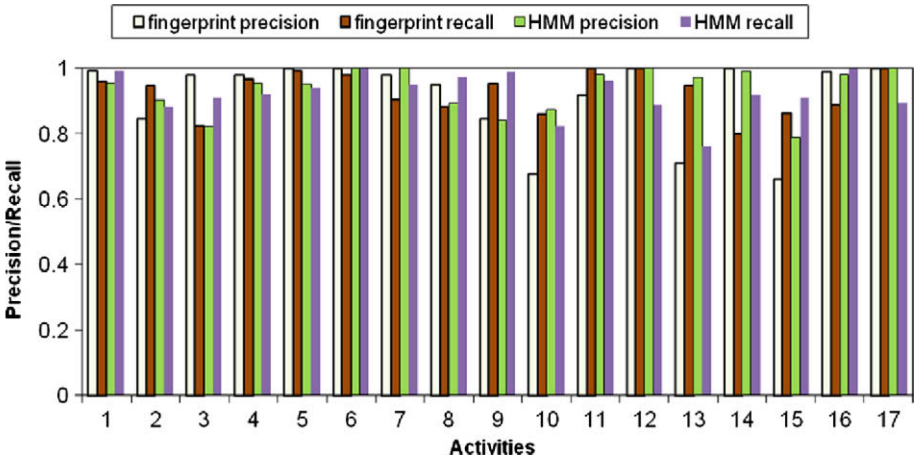


Fig. 3. Detailed comparison results of our fingerprint-based recognition algorithm and HMM (Numbers in *x*-axis correspond to Table 3).

Table 4
Activities performed in our trace collection.

1	make coffee	7	brush teeth	13	clean dining table
2	make tea	8	wash clothes	14	play PC games
3	make pasta	9	make orange juice	15	watch TV
4	make oatmeal	10	watch DVD	16	put on make-up
5	fry eggs	11	take pills	17	use toilet
6	make phone call	12	read books		

wireless module connected through its serial port. The sever samples tag IDs at a frequency of 1 Hz and records them in a text file.

We then conduct a real-world trace collect over a period of two weeks in a smart home environment. We had seven volunteers (one female and six males) and each volunteer wore an RFID wristband reader on each of her/his hands. We tagged 132 day-to-day objects. We selected a total number of 17 activities as summarized in Table 4. These activities are commonly used in other work [2–5,7–14]. Each day, each of them performed these activities in any order they like. Each subject was requested to perform their activities consecutively in their own ways based on their daily practices. A trace (i.e., a sequence of tag IDs) was logged each day in a server. There was only one subject performing activities at any given time to reduce annotation efforts. One of the volunteers annotated the traces to establish the ground truth. All the traces collected were annotated by hand.

5.1. Comparison results

To evaluate the performance of our activity recognition algorithm, we use the following metrics

Table 5
Comparison result.

	Average precision (%)	Average recall (%)
Fingerprint-based	91.4	92.8
HMM	93.5	92.5

$$precision = \frac{TP}{TP + FP} \quad and \quad recall = \frac{TP}{TP + FN}$$

where TP (*true positive*) refers to the number of instances where an activity recognition algorithm correctly identifies an activity; FP is (*false positive*) refers to the number of instances where the algorithm spuriously detects an activity that does not actually occur; and FN is (*false negative*) denotes the number of times that an activity that actually occurred is not detected.

We compare our fingerprint-based recognition algorithm with a supervised learning model – HMM, which is commonly used in activity recognition. HMM is a generative probabilistic model consisting of a hidden variable and an observable variable at each time step. To test the HMM model, we use leave-one-out cross validation. We build a HMM model where activity labels are represented as hidden states and objects are represented as observations. The model will be trained first, and then we use the Viberbi algorithm to recover the hidden state sequence for testing.

The overall comparison result of the two models is shown in Table 5 and the detailed result is given in Fig. 3. While the learned HMM model gets the highest precision, our algorithm achieves a comparable result in terms of precision and recall. After analysis, two limitations are worth noting. First, when RFID readers failed to report tag IDs or received corrupted tag IDs, missed detection occurred. This kind of errors is a common problem in sensor-based activity recognition. Efforts to improve the reliability of sensor deployment and acquisition process require further investigation. Second, some cases of false recognition occurred when the subject touched another objects unintentionally while an activity is being performed or because of the close proximity to nearby objects. The consequence in this case will cause the objects collected around the neighborhood of these objects to carry relatively high weights in another activity. Section 5.2 evaluates this case further and discusses our potential solution.

We now compare the performances of the MaxGap and MaxGain algorithms. The performance comparison is based on the two metrics, namely: (1) **M**ean **A**bsolute **E**rror from the true **B**oundary (MAEB) and (2) **M**ean **P**ercentage of the true **B**oundaries detected (MPB). MAEB is a continuous value that measures how many objects away is the algorithm's boundary from the true boundary. A good algorithm must have MAEB value near zero. MPB, on the other hand, is the ratio between the number of true boundaries detected by the algorithm and the total number of boundaries. A good algorithm has MPB value near 100%. MAEB and MPB are summarized in as follows:

$$\mathrm{MAEB} = \frac{\sum_{j=1}^{T}\sum_{i=1}^{N}|trB_{ij} - a\lg B_{ij}|}{N * T} \quad \text{and} \quad \mathrm{MPB} = \frac{\sum_{j=1}^{T}\frac{D_j}{B_j} \times 100}{T}$$

where trB is the true boundary; $algB$ is the algorithm's boundary; N is the total number of boundaries; T is the number of traces; D_j is the number of correctly detected boundaries in a trace; B_j is the total number of true boundaries in a trace.

In the evaluation, we did not include detecting boundary in cases where two consecutive activities are of the same type since it is trivial to detect such boundaries.

Table 6 shows the comparison result. As expected, MaxGain outperforms MaxGap since MaxGain takes the consideration of *Group RW* (i.e., *RW*s of all the objects) in the neighbor of a boundary rather than the *RW* of two consecutive objects. From the above result, we observe two limitations. First, by analyzing the ground truth, many cases of false segmentation are caused by missed detection and false recognition in the activity recognition process as explained previously. However, the detection of each boundary is done in an independent way that the false segmentation for one boundary does not affect the segmentation for other boundaries. That is the reason we can still achieve relatively high accuracies in the presences of missed detection and false recognition during the segmentation process.

5.2. Robustness and scalability

This section evaluates and discusses our concerns in the previous sections. The trace collection in this paper was performed by a number of subjects in a smart home environment. These traces provided a variety of test cases in a close-to-real-life, noisy environment to validate and evaluate the effectiveness of the proposed algorithms. However, it was done

Table 6
Comparison of the segmentation algorithms.

	MAEB	MPB (%)
MaxGap	0.6	83.5
MaxGain	0.2	93.1

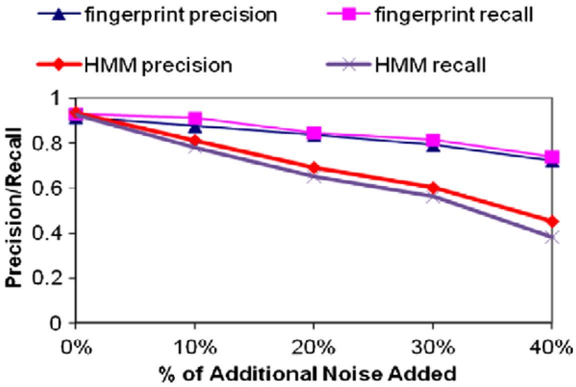


Fig. 4. Precision/recall of the recognition algorithms vs. different percentages of additional noise.

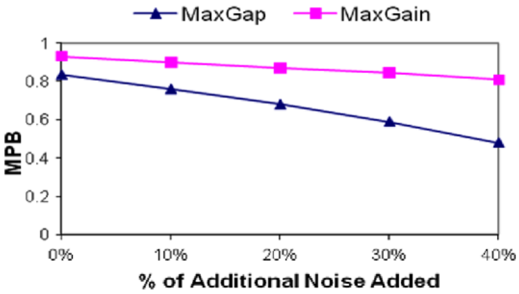


Fig. 5. MPBs of both the MaxGap and MaxGain algorithms vs. different percentages of additional noise.

in a mock-up situation. In real-life situations, we may see increasing sensor noise in the traces we collected. To evaluate the robustness of both the activity recognition and segmentation algorithms, we conducted a number of simulation studies by manually adding additional noise to our traces. Noise was randomly selected from the entire tagged IDs. This simulation allows us to test our algorithms in cases of a subject touching other tagged objects unintentionally in the activity being performed or because of the close proximity of nearby objects. We generated a set of traces by adding different percentages of noise randomly to each trace, and compared our fingerprint-based recognition algorithm with a HMM model. The comparison result is shown in Fig. 4. As expected, both two models experience performance drop; however, both decrease linearly. The precisions/recalls of our fingerprint-based algorithm and the HMM model decrease by 19.4%/18.8% and 48.5%/54.5%, respectively, when 40% of additional noise was added. The result also demonstrates the fingerprint-based recognition algorithm is more robust to sensor noise as compared to HMM since Fig. 4 shows that both the precision and recall of a HMM model decrease rapidly.

Fig. 5 shows the comparison result of the MaxGap and MaxGain algorithms in the presence of additional noise. The MPBs of MaxGap and MaxGain decrease by 35.5% and 12.1%, respectively, when 40% of additional noise was added. MaxGain outperforms MaxGap since MaxGain decreases in a much lower gradient as compared to MaxGap. This is probably because *Group RW* resists random noise much better than *RW*.

Supervised learning techniques such as HMM typically require a dataset for the training process in order to build various activity models. For better scalability, we have to obtain a very large training dataset over a long period of time, resulting in huge efforts on manual labeling. On the contrary, our fingerprint-based recognition algorithm is unsupervised, and we can mine a comprehensive, large number of activity models on the web without manual labeling. Hence, our solution can achieve better scalability than a HMM model.

6. Conclusion

In this paper, we propose to build our activity models based on object-use fingerprints by combining web mining and data mining techniques. We mine a set of object terms form the web for each activity, and mine a set of contrast patterns as fingerprints describing significant differences of object terms between any two activity classes. We then propose fingerprint-based algorithm to recognize activities. We also address the trace segmentation problem by proposing the two algorithms, MaxGap and MaxGain, based on the comparison of the relative weights of all objects sandwiched in the two adjacent activities. We conduct real-world trace collection in a smart home environment. The experimental results demonstrate both effectiveness and robustness of our algorithms.

Though RFID has been used in both research and commercially approved contexts, the long-term effects on humans require thorough investigation and further research towards human-centric RFID [43]. For our future work, we will further develop our sensor platform to include more sensor features and seek a more nature data collection which should be conducted in a real home under real-life situations. In addition, we will investigate complex issues in recognizing interleaved and concurrent activities.

References

[1] S. Katz, A.B. Ford, R.W. Moskowitz, B.A. Jackson, 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 185 (1963) 914–919.

[2] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: Proc. of Pervasive 2004, Vienna, Austria, April 2004, pp. 1–17.

[3] E.M. Tapia, S.S. Intille, K. Larson, Activity recognition in the home using simple and ubiquitous sensors, in: Proc. of Pervasive 2004, Vienna Austria, April 2004, pp. 158–175.

[4] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, D. Hähnel, Inferring activities from interactions with objects, IEEE Pervasive Computing 3 (4) (2004) 50–57.

[5] Y. Nakauchi, K. Noguchi, P. Somwong, T. Matsubara, Human intention detection and activity support system for ubiquitous sensor room, Journal of Robotics and Mechatronics 16 (5) (2004) 545–551.

[6] Y. Matsuo, N. Okazaki, K. Izumi, Y. Nakamura, T. Nishimura, K. Hasida, H. Nakashima, Inferring long-term user property based on users' location history, in: Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, January 2007, pp. 2159–2165.

[7] T. Huynh, B. Schiele, Towards less supervision in activity recognition from wearable sensors, in: Proc. of the 10th IEEE International Symposium on Wearable Computing (ISWC 2006), Montreux, Switzerland, October 2006, pp. 3–10.

[8] T. Huynh, B. Schiele, Unsupervised discovery of structure in activity data using multiple eigenspaces, in: Proc. of the Second International Workshop on Location and Context-Awareness, Dublin, Ireland, May 2006, pp. 151–167.

[9] D. Patterson, D. Fox, H. Kautz, M. Philipose, Fine-grained activity recognition by aggregating abstract object usage, in: Proc. of ISWC 2005, Osaka, Japan, October 2005, pp. 44–51.

[10] J.A. Ward, P. Lukowicz, G. Troester, T. Starner, Activity recognition of assembly tasks using body-worn microphones and accelerometers, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1553–1567.

[11] B. Logan, J. Healey, M. Philipose, E. Munguia-Tapia, S. Intille, A long-term evaluation of sensing modalities for activity recognition, in: Proc. of Ubicomp 2007, Innsbruck, Austria, September 2007, pp. 483–500.

[12] M. Perkowitz, M. Philipose, D.J. Patterson, K.P. Fishkin, Mining models of human activities from the web, in: Proc. of the 13th International World Wide Web Conference (WWW 2004), New York, NY, USA, May 2004, pp. 573–582.

[13] D. Wyatt, M. Philipose, T. Choudhury, Unsupervised activity recognition using automatically mined common sense, in: Proc. of AAAI 2005, Pittsburgh, PA, USA, July 2005, pp. 21–27.

[14] W. Pentney, A.-M. Popescu, S. Wang, H. Kautz, M. Philipose, Sensor-based understanding of daily life via large-scale use of common sense, in: Proc. of AAAI 2006, Boston, MA, USA, July 2006.

[15] M.F. Porter, An algorithm for suffix stripping, Program 14 (3) (1980) 130–137.

[16] Stop Word List, <http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words>.

[17] English Club, <http://www.englishclub.com/>.

[18] Adjective List, <http://www.keepandshare.com/doc/view.php?u=12894>.

[19] W.B. Frakes, R.A. Baeza-Yates, Information Retrieval: Data Structures and Algorithms, Prentice-Hall, 1992.

[20] M. Berry, Survey of Text Mining: Clustering, Classification, and Retrieval, Springer-Verlag, 2004.

[21] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[22] P. Rybski, M. Veloso, Robust real-time human activity recognition from tracked face displacements, in: Proc. of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), Lecture Notes in Artificial Intelligence 3808, LNAI Springer, December 2005, pp. 87–98.

[23] F. Niu, M. Abdel-Mottaleb, HMM-based segmentation and recognition of human activities from video sequences, in: Proc. of IEEE International Conference on Multimedia and Expo (ICME 2005), July 2005.

[24] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, C. Isbell, A novel sequence representation for unsupervised analysis of human activities, Artificial Intelligence Journal 173 (14) (2009) 1221–1244.

[25] Yaser Yacoob, Michael J. Black, Parameterized modeling and recognition of activities, in: Proc. of the Sixth International Conference on Computer Vision (ICCV'98), 1998.

[26] I. Haritaoglu, D. Harwood, L.S. Davis, W4: real-time surveillance of people and their activities, IEEE Transactions on Pattern Recognition and Machine Intelligence 22 (8) (2000) 809–830.

[27] T. Duong, H. Bui, D. Phung, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model, Proceedings of the CVPR 1 (2005) 838–845.

[28] Xinding Sun, Ching-Wei Chen, B.S. Manjunath, Probabilistic motion parameter models for human activity recognition, in: Proc. of the 16th International Conference on Pattern Recognition (ICPR'02), vol. 1, 2002.

[29] Jonathan Lester, Tanzeem Choudhury, Gaetano Borriello, A practical approach to recognizing physical activities, in: Proc. of International Conference on Pervasive Computing, 2006.

[30] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, M. Philipose, Common sense based joint training of human activity recognizers, in: Proc. of IJCAI 2007, Hyderabad, January 2007.

[31] Douglas L. Vail, Manuela M. Veloso, John D. Lafferty, Conditional random fields for activity recognition, in: Proc. of International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2007.

[32] Tsu-yu Wu, Chia-chun Lian, Jane Yung-jen Hsu, Joint Recognition of multiple concurrent activities using factorial conditional random fields, in: Proc. of AAAI Workshop on Plan, Activity, and Intent Recognition, California, July 2007.

[33] Derek Hao Hu, Qiang Yang, CIGAR: concurrent and interleaving goal and activity recognition, in: Proc. of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008), Chicago, IL, USA, 2008.

[34] R. Hamid, S. Maddi, A. Bobick, I. Essa, Unsupervised analysis of activity sequences using event motifs, in: Proc. of the Fourth ACM International Workshop on Video Surveillance and Sensor, 2006.

[35] Guozhu Dong, Jinyan Li, Efficient mining of emerging patterns: discovering trends and differences, in: Proc. of the Fifth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, USA, August 1999, pp. 43–52.

[36] H. Kautz, A Formal Theory of Plan Recognition, Ph.D. Dissertation, University of Rochester, 1987.

[37] Niels Landwehr, Modeling interleaved hidden processes, in: Proc. of the 25th International Conference on Machine Learning (ICML'08), Helsinki, Finland, July 2008.

[38] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proc. of the 11th International Conference on Data Engineering, Taipei, Taiwan, 1995, pp. 3–14.

[39] Yida Wang, Ee-Peng Lim, San-Yih Hwang, Efficient mining of group patterns from user movement data, Data and Knowledge Engineering 57 (3) (2006) 240–282.

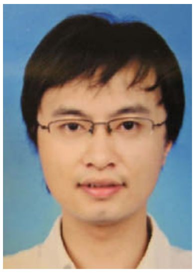
[40] Themis P. Exarchos, Markos G. Tsipouras, Costas Papaloukas, Dimitrios I. Fotiadis, A two-stage methodology for sequence classification based on sequential pattern mining and optimization, Data and Knowledge Engineering 66 (3) (2008) 467–487.

[41] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M-C. Hsu, FreeSpan: frequent pattern-projected sequential pattern mining, in: Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000, pp. 355–359.

[42] Xiaonan Ji, James Bailey, Guozhu Dong, Mining minimal distinguishing subsequence patterns with gap constraints, Journal of Knowledge and Information Systems 11 (3) (2007) 259–286.

[43] Amelia Masters, Katina Michael, Lend me your arms: the use and implications of human-centric RFID, Electronic Commerce Research and Applications 6 (1) (2007) 29–39.

Dr. Tao Gu is currently an Assistant Professor in the Department of Mathematics and Computer Science at University of Southern Denmark. He received his B. Eng. from Huazhong University of Science and Technology, and M.Sc. from Nanyang Technological University, Singapore, and Ph.D. in Computer Science from National University of Singapore. He has served as a technical program committee member for many international conferences such as Percom, Mobituitous, ICDCS, etc. His current research interest includes pervasive computing and wireless sensor networks.



Shaxun Chen received his B.S. and M.Sc. degrees in Computer Science from Nanjing University, China, in 2005 and 2008, respectively. He was a research assistant in Institute for Infocomm Research, Singapore, in 2007. Now he is a Ph.D. student at UC Davis, USA. His research interest includes wireless network security and pervasive computing.

Xianping Tao is currently a Professor in the Department of Computer Science and Technology at Nanjing University. He received his M.S. and Ph.D. in Computer Science from Nanjing University in 1994 and 2001, respectively. His research interest includes software agents, middleware systems, internetware methodology and pervasive computing.

Jian Lu is currently a Professor in the Department of Computer Science and Technology, Nanjing University. He received his B.Sc., M.Sc. and Ph.D. in Computer Science at Nanjing University in 1982, 1984 and 1988, respectively. He is currently a Professor in the Department of Computer Science and Technology, Nanjing University. He is also the Director of the State Key Laboratory for Novel Software Technology and is the Vice Director of the Institute of Software Technology at Nanjing University. His research interest includes programming methodology, pervasive computing, software agent, and middleware. He has published over 100 technical papers in the above areas. He is a member of the Board of the UNU International Institute for Software Technology and ACM. He is also the chairman of Software Engineering Professional Committee of the China Computer Federation. He has served as an organizing/program committee member for many international conferences.