IETE TECHNICAL REVIEW

Published bimonthly by the Institution of Electronics and Telecommunication Engineers

March - April 2010 | Volume 27 | Issue 2

Contents

A Smoothed Naïve Bayes-Based Classifier for Activity Recognition

A. M. Jehad Sarkar, Young-Koo Lee and Sungyoung Lee

Department of Computer Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea

Abstract

A number of classifiers have been proposed by the researchers for activity recognition using binary and ubiquitous sensors. Many researchers have shown that the hidden Markov model (HMM) and the conditional random field (CRF)-based activity classifiers work well to classify activities in comparison with the widely used naïve Bayes-based activity classifier. However, it would not be an exact verdict if a naïve Bayes-based activity classifier is properly smoothed. Parameter estimation plays the central role in the performance of a naïve Bayes activity classifier. Data sparsity puts substantial challenges in parameter estimation because the sizes of the real-life activity datasets are relatively small. The distribution of the sensors may not be even among the activity classes. Additionally, some of the sensors would appear during testing but would not appear while training. This is called zero-frequency problems which assign zero probability of a sensor for a given activity. To prevent such estimation problems, we propose two smoothing techniques for adjusting the maximum likelihood to produce more precise probability of a sensor given an activity. We performed three experiments using three real-life activity datasets. It is observed that our proposed mechanism yields significant improvement in the accuracy of activity classification in comparison with its existing counterparts. We achieved the class accuracy ranging between 63% and 83%.

Keywords

Activity recognition, Naïve Bayes-based classifier, Simple and ubiquitous sensors, Smoothing, Zero-probability.

1. Introduction

The recognition of everyday activities of individuals like walking, sleeping, cooking, etc., is one of the current focuses of the researchers due to its strength in providing personalized support for many different applications [1-4]. A typical application of an activity recognition system (ARS) would be to assist sick or elderly people.

The sensor-based ARS integrates wireless sensor networks with machine learning and data mining methods to model a broad range of human activities [5]. Three types of sensors have been tried by the researchers to classify human activities: Video based, wearable sensors based, and based on sensors deployed in the environment embedded with the home appliances (e.g., door, light, and closet). Video-based methods have the disadvantage of breaking user's privacy, whereas wearable sensors require the user to wear sensors and their accuracy depends on the position of the attachments. Therefore, activity detection methods based on sensors deployed in the environment are getting more focus. In this paper, we propose a successful and accurate AR system using simple, low-cost "tape on and forget" sensors.

Address for correspondance: *Prof. Young-Koo Lee,*
E-mail: yklee@khu.ac.kr

A number of classifiers have been proposed by the researchers for activity classification using binary and ubiquitous sensors. Many researchers have shown that the hidden Markov model (HMM) [6] and the conditional random field (CRF) [3,6] based activity classifier worked well in comparison with the widely used naïve Bayesian (NB)-based classifiers. However, it would not be an exact verdict if the parameters of NB-based activity classifiers are properly smoothed.

The parameter estimation for probabilistic models uses the method of maximum likelihood (ML). Data sparsity is a major problem in estimating ML in AR because the size of the training data is relatively small in comparison with the other machine learning datasets. The distribution of the observed sensors in a dataset may not be always even between the activities. Additionally, some sensors would appear during testing but would not appear while training. This is called zero-frequency [7] problems which assign zero probability of an unseen sensor for an activity.

To prevent such estimation problem, smoothing is required to adjust the maximum likelihood of a model to make it more accurate. At the very least, it is required to not assign zero probability to the unseen sensor. When estimating a ML based on a limited amount of sensors, such as a single activity instance, smoothing of the ML is extremely important.

Most of the probabilistic methods like, the NB, the HMM, and the CRF suffer from the zero-probability problem. However, in this paper, we focused on smoothing the NB-based activity classifier.

A set of smoothing techniques have been proposed in the field of speech recognition (SR) and information retrieval (IR) [8]. The Jelinek-Mercer (JM) [9] (also referred to as the linear interpolation language model) and the Bayesian smoothing (BS) using Dirichlet priors [10] are two commonly used smoothing techniques used in IR to retrieve documents based on user's query. To our best knowledge, no smoothing techniques have been proposed in the field of AR. In this paper, we proposed two smoothing techniques which are based on JM and BS.

Our contributions in this paper are twofold. First, we propose two smoothing techniques for adjusting the maximum likelihood of the probabilities to produce a more precise activity model. Second, we perform a series of experiments with three real-life activity datasets. And we proved that our proposed mechanism yields significant improvement in the accuracy of activity classification in comparison to its existing counterparts.

The rest of the paper is organized as follows. In section 2, we present the reviews of previous works related to AR. In section 3, we discuss the background associated with the smoothing techniques proposed in the field of IR. In section 4, we discuss the overview of our proposed system. In section 5, we discuss our proposed algorithms. In section 6, we present our experimental results to support our claims. In section 7, we conclude our paper with a direction of future work.

2. Related Works

Many research groups have been investigating how to construct smart living environments that target medical care for the individual. The Intel Research Group in Seattle and the University of Washington have built a prototype system that can infer a person's activities of daily living (ADLs) [11]. In their system, the sensors are embedded on everyday objects such as a toothbrush or coffee cup. University of Rochester is building the Smart Medical Home, which is a five-room house outfitted with infrared sensors, computers, biosensors, and video cameras for use by research teams to work with research subjects as they test concepts and prototype products [12]. Georgia Tech built an Aware Home as a prototype for an intelligent space [13]. Massachusetts Institute of Technology (MIT) and TIAX are working on the PlaceLab initiative, which is a part of the House_n [14] projects. The mission of House_n is to conduct research by designing and building real living environments—"living labs" that are used to study technology and design strategies in context. Many projects are building body networks for the collection of vital signs, such as AMON. All these systems demonstrate the excitement and need for such systems [15].

AR based on sensors can be categorized into three different types: An AR system that uses simple and ubiquitous sensors which are deployed in the environment embedded with appliances, an AR system that uses video cameras which are deployed in environment usually screwed in a wall or roof and an AR system that uses wearable sensors (e.g., accelerometer) which are attached with the body of an individual. In simple sensor-based AR, an activity is recognized through a stream of sensory data acquired from different sensors. In video camera-based AR, sequences of video frames obtained from one or more cameras are used to determine the activity. In accelerometer-based AR, acceleration signals in three axes (x, y, and z) are used to infer a user's activity.

To our best knowledge, Intille *et al*. [16] were the first to employ simple and ubiquitous sensors for AR. The authors provided the context-aware experience sampling tool (ESM) [17,18] in a PDA to the users to annotate their daily activities. A NB classifier was used recognize activities. The authors have shown an excellent promise, even though their mechanism suffers from low recognition accuracy.

In [6], the authors used similar settings, except that their annotation technique was quite innovative. They employed a predefined set of voice commands to start and end an activity through a Bluetooth-enabled headset combined with speech recognition software. The problem of this annotation technique is that no one can guarantee that the start and the end point of an activity will always be marked properly by the participants. It does not even alert the participants to label the start and the end point. In addition to these, their proposed classifiers are not general purpose. They utilized hidden Markov model (HMM) and conditional random field (CRF) as the classifiers. The HMM or the CRF can be computationally very expensive because the number of observed variable grows as the number of sensors does. For example, if 50 state-change (binary) sensors are used, the number of emission for each state in HMM or CRF would be 2^{50}.

In [19], the authors introduced the simultaneous tracking and activity recognition (STAR) to perform accurate tracking and activity recognition for multiple people in a home environment using anonymous and binary sensors (motion detectors, break-beam sensors, pressure mats, and contact switches). They employed a Rao-

Blackwellized particle filter approach to determine which rooms were occupied, and to count the occupants in a room, identify the occupants, track occupant movements, and recognize whether the occupants were moving or not.

Activities can also be detected through audio, video sensors or body-attached sensors. For example, Zajdel *et al*. [20] used audio video sensors for aggression detection. They first performed an independent analysis of the audio and video streams to get the descriptors of a scene like "scream," "passing train," or "articulation energy." Next, they used a dynamic Bayesian network (DBN) [21] as a fusion mechanism that produces an aggregate aggression indication for the current scene. In [22], the authors showed how body-attached sensors can be used to recognize activities of assembly tasks. The glitches of these approaches are (i) difficulties in signal analysis, (ii) people not always comfortable wearing sensors, and (iii) expensive solution.

In [23], the authors considered a sensor network in office environment. The concept of hierarchical feature extraction is used to detect a user's activity from aggregated sensor data. The naïve Bayesian inference engine is used to take input from the feature extractor and gives a user's activity as an output.

Also, many mobility-based and object-usage-based activity classification mechanisms have been proposed. For example, in [1], the authors used a Bayesian filter to infer and predict a user's transportation mode, such as "walking," "driving," or "taking a bus" from GPS data. In [24], the authors used a DBN to classify user activities such as "using the bathroom," "making coffee," etc., based on object usage (with embedded RFID tags).

Our proposed ARS is closely related to the AR systems proposed in [16] and [6]. The differences are the way of estimating model parameters and activity classification. We used the NB-based classifier with two smoothing techniques to improve the parameter estimation accuracy.

3. Background

Our smoothing techniques are based on two popular smoothing techniques used by the language models for IR. In this section, we describe the theories related to the language models and the smoothing techniques.

IR is the way to retrieve relevant documents based on the user's query. In order to come up with good queries to retrieve the relevant documents, we need to think of the words (or terms) that would likely appear in these documents. In IR, the language modeling approach directly models that idea: If the document model is likely to generate a query, it will be a good match for the query, and it will happen if the document contains the query words often [7].

In other words, in the language modeling approach to IR, we can consider the probability of a query as being generated by a probabilistic model based on a document. For a query $q = q_1, q_2 \cdots, q_n$ and a document d, this probability is denoted by $p(q \mid d)$ [8]. In order to rank documents, the posterior probability $p(d \mid q)$ is estimated by the Bayes formula,

$$\mathrm{P(d|q)} \propto \mathrm{P(q|d)P(d)}$$

where $p(d)$ is the prior probability of a document for any query and $p(q \mid d)$ is the likelihood of the query for a given document d. In IR, the $p(d)$ is considered to be uniform and therefore ignored. The likelihood $p(q \mid d)$ is calculated as

$$P(q|d) = \prod_{i=1}^{n} P(q_i|d) = \prod_{i=1}^{n} \frac{tf_{q_i,d}}{L_d}$$

where $tf_{q_i,d}$ is the *term frequency* of the term q_i in a document d, $L_d = \sum_{t \in d} tf_{t,d}$ is the length of the document, and t is a term. This is called the *query likelihood model* which is the original and basic method of language modeling in IR.

The classic problem of language modeling is one of estimation: The terms appear sparsely in the documents. In particular, if a query term q_i does not appear in the document then $P(q \mid d)$ will be 0. This is called *zero-probability* estimation problem [7]. Such a problem leads researchers to smooth probabilities in document language models to discount nonzero probabilities and to give some probability mass to unseen terms.

A wide variety of smoothing techniques have been proposed. The JM [9] (also referred as the linear interpolation language model) and the BS using Dirichlet priors [10] are two popular smoothing methods used in language models. The main idea behind these methods is to discount the probability of the words seen in the document and assign the extra probability mass to the unseen terms according to some "fallback" model.

Jelinek-Mercer smoothing: It is a simple idea but works extremely well in practice. It usages a mixture between a document-specific and entire collection-specific multinomial distribution:

$$P(t \mid d) = \lambda P_{mle}(t \mid M_d) + (1-\lambda) P_{mle}(t \mid M_c)$$

where $0 < \lambda < 1$ is the smoothing parameter and M_d and M_c are the language models derived from a document and from the entire document collection, respectively.

Bayesian smoothing using Dirichlet priors: An alternative of JM smoothing is to use a language model built from the whole collection as a prior Bayesian distribution in a Bayesian updating process. This is written as

$$P(t \mid d) = \frac{tf_{t,d} + \mu P_{mle}(t \mid M_c)}{L_d + \mu}$$

where μ is the smoothing parameter. A large value of μ means more smoothing.

3.1 Other Smoothing Techniques

Laplace or additive smoothing [25] is the simplest smoothing method which works by adding an extra count to every term. The probability mass of a term given in a document is calculated as

$$P(t \mid d) = \frac{tf_{t,d} + 1}{L_d}$$

The problem of the Laplace smoothing is that it gives too much probability mass to unseen terms.

An improved smoothing method is the Good-Turing smoothing [26] which reestimates the frequency of the term that occurs *tf* times [27] as

$$tf_t^* = (tf_t + 1)\frac{n_{tf_t} + 1}{n_{tf_t}}$$

where n_{tf_t} is the number of terms that occur exactly tf_t times in the training data. Good-Turing is often used in combination with the backoff and interpolation algorithms rather than using it itself.

A more sophisticated smoothing technique known as Katz smoothing [28] extends Good-Turing estimation. The Katz smoothing method is a well-known backoff method which works by discounting and redistributing probability mass only for the less common terms. Such a technique is popular in speech recognition.

Absolute discounting [29] is another smoothing method used in IR. The idea is similar to the interpolation method. It works by discounting the probability of seen terms by subtracting a constant instead of multiplying it.

4. Activity Recognition System

4.1 Overview

Figure 1 shows the overview of our ARS. The proposed ARS consists of three major phases:

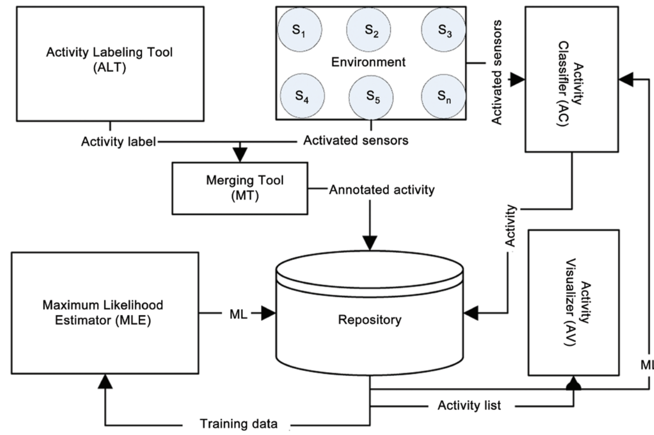


Figure 1: Activity recognition system overview.

(1) The data gathering phase: The goal of this phase is to deploy a number of sensors in the environment (e.g., home) and annotate their triggering pattern under human action for a prespecified period of time. An ARS uses the activity labeling tool (ALT) (e.g., ESM) and the merging tool (MT) to annotate the participant(s) activity.
(2) The training phase: The goal of this phase is to estimate the likelihoods of the sensors for an activity and a set of activities using the maximum likelihood estimator (MLE).
(3) The classification and visualization phase: The goal of this phase is to return the likelihood of current activities using an activity classifier (AC) and to provide a graphical user interface (GUI) to monitor the day-to-day activities using an activity visualizer (AV).

4.2 Naïve Bayesian Classifier for AR

Studies comparing classification algorithms show that a simple Bayesian classifier known as the NB classifier exhibits extremely good performance in various machine learning applications [30].

The NB-based activity classifier assumes that the effect of an object on a given activity is independent of the other object. This assumption is called activity conditional independence. For classification, the classifier computes the posterior probability P ($A \mid \Theta$) using the Bayes rule:

$$P(a_i \mid \Theta) \propto P(a_i)\prod_{k=1}^{|\Theta|} P(\theta_k | a_i)$$

where $a_i \in A$ represents an activity (e.g., bathing), A represents the set of activities, and $P(a_i)$ is the *prior probability* (*PP*) of an activity, $P(\theta_k \mid a_i)$ is the ML of θ_k given an activity a_i, $\Theta \subset O$ is the vectors of activated sensors (as the subject interact with objects with embedded sensors) at a given time frame, O is the set of objects with embedded sensors.

During training, we estimate the following probabi lities:

$$P(o_1 \mid a_i), P(o_2 \mid a_i) \dots P(o_t \mid a_i), P(a_i) \forall a_i \in A.$$

where $o_1, o_2 \dots, o_t \in O$ and t is the total number of objects (with embedded sensors) in the environment. In order to classify the activity label of Θ, $P(a_i)P(\Theta \mid a_i)$ is evaluated for each activity a_i. The classifier predicts that the activity label of vector is the activity a_i if and only if

$$P(a_i)P(\Theta \mid a_i) > P(\Theta_j)P(\Theta \mid a_j) \; for \; 1 \le j \le m, j \ne i$$

where m is the total number of activities. In other words, the classified activity label is the activity a_i for which $P(a_i)P(\Theta \mid a_i)$ is the maximum.

4.3 Smoothing Techniques

In a NB-based classifier for AR, the ML is estimated as follows:

$$P(\theta_k \mid a_i) = \frac{SF(\theta_k \mid a_i)}{\sum_{j=1}^{n} SF(\theta_j \mid a_i)}$$

where $SF(\theta_k \mid a_i)$ implies the observed sensor frequency (SF) of the k^{th} sensor for an activity a_i in the training data.

The MLE will generally underestimate the probability of any sensor that is not triggered for an activity during training. For example, Figure 2 shows the observed frequencies of three sensors: "kitchen cabinet," "medicine cabinet," and "containers" for 10 activities in an activity dataset [31]. We can see that all the three sensors have zero frequency for many activities. Hence simple ML estimation will result in many zero probabilities. So the main purpose of smoothing is to assign a nonzero probability to the unseen sensors and improve the accuracy of sensor probability estimation in general.

We propose two smoothing techniques to estimate precise MLs which are based on JM and BS (described in section 3). Before explaining these methods, we define two terms: An activity model (AM) M_{a_i} (each row of Table 1) which is associated with an activity a_i within a collection of activities and the collective model (CM) M_c which is derived from the collection of activities c (entire Table 1).

The main idea behind the proposed smoothing methods is to discount the probability of the sensors seen in the activity and assign the extra probability mass to the unseen sensors.

JM-based method: This is a simple mixture method which involves a linear interpolation of the AM with the CM, using a coefficient λ to control the influence of each:

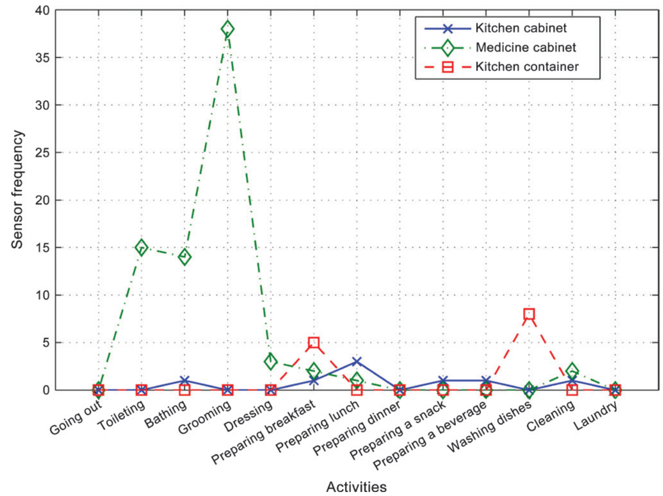


Figure 2: An example: Data sparsity.

Table 1: An example of M_a and M_c (each column represents how many times the k^{th} sensor triggers for all the activities)

Sensors activity	0_1	0_2	0_3	0_4	0_5	...	0_n
Leaving	2	17	21	1	5	...	2
Toileting	1	40	195	0	0	...	16
Showering	0	68	1	0	0	...	0
Sleeping	0	15	13	0	0	...	44
Breakfast	7	1	0	2	38	...	0
Dinner	5	0	4	4	23	...	0
Drink	0	0	0	17	30	...	1

$$P(a_i \mid \Theta) \propto P(a_i)(\lambda P(\theta_k \mid M_{a_i}) + (1-\lambda)P(\theta_k \mid M_c)) \quad (1)$$

BS-based method: This is a multinomial distribution, for which the conjugate prior for Bayesian analysis is the Dirichlet distribution. The parameters of the Dirichlet are

$$\mu P(\theta_1 \mid M_c), \mu P(\theta_2 \mid M_c), \cdots, \mu P(\theta_n \mid M_c)$$

Thus, the method is given by

$$P(a_i \mid \Theta) \propto P(a_i) \frac{SF(\theta_k \mid M_{a_i}) + \mu P(\theta_k \mid M_c)}{length_{a_i} + \mu} \quad (2)$$

where

$$P(a_i) = \frac{Total\ number\ of\ times\ activity\ "a_i"\ are\ pursued}{Total\ time\ T\ in\ seconds}$$

$$P(\theta_k \mid M_{a_i}) = \frac{SF(\theta_k \mid M_{a_i})}{\sum_{k=1}^{n} SF(\theta_k \mid M_{a_i})}$$

$$P(\theta_k \mid M_c) = \frac{\sum_{i=1}^{m} SF(\theta_k \mid M_{a_i})}{\sum_{k=1}^{n} \sum_{i=1}^{m} SF(\theta_k \mid M_{a_i})}$$

and

$$length(a_i) = \sum_{k=1}^{n} SF(q_k \,|\, M_{a_i})$$

where n and m are the number of sensors and activities, respectively, and $length(a) = \sum_{k=1}^{n} SF(\theta_k | M_a)$ is the activity length (AL).

In equations (1) and (2), $0 < \lambda < 1$ and μ are the smoothing parameters.

Comparing equations (1) and (2), distinct features of these smoothing techniques can be observed. Two types of probabilities are considered by both methods, one associated with an activity and other with the collection of activities. And both of these methods produce a movement of probability mass from seen sensors to unseen sensors. But the movement is done in different ways. In Bayesian smoothing using Dirichlet prior-based smoothing, the movement is activity-length-dependent but in the Jelinek-Mercer based method, the movement is independent of the activity length. In other words, in the BS method, μ is involved in the activity length which is opposite to in the JM method. If the length of an activity is sufficiently large, the μ can be omitted in the BS-based method.

5. Algorithms

In this section, we describe our algorithms for training and inference. During the training phase, the goal is to determine $P(\theta_k | M_{a_i})$ for an AM. This is followed by estimating the maximum likelihood, $P(\theta_k | M_c)$ for the CM. And the goal of inference is to rank the activities based on the sensors triggered for a given time period and finally produce the top-ranked activity as the output.

5.1 Training

The training phase begins after the deployment of the sensors and their triggering pattern under human actions is recorded for a prespecified period of time (e.g., 30 days).

In this phase, various model parameters like $P(\theta_k | M_{a_i}), P(\theta_k | M_c)$ and $length_{a_i}$ are estimated.

Algorithm 1 shows our proposed algorithm for training.

The algorithm first calculates the relative frequencies of the sensors for all the activities, i.e., $SF(s_k | M_{a_i})$. This is followed by calculating maximum likelihoods for the activity models, i.e., ML_{AM}. And finally it calculates the maximum likelihoods for the collective model, i.e., ML_{CM}.

5.2 Activity Classifier

The system would be ready to recognize an activity in real time, as soon as all the model parameters have been estimated. This is called inference phase. In this phase, activities are inferred by an inference engine (or classifier) which uses sensory data coming from triggered sensors (as humans interact with the object with an embedded sensor).

Algorithm 1: Training

```
Data: Activity Instances I for training, list of
      sensors S in the environment, list of activities
      A to monitor
Result: ML_AM, ML_CM → Two m x n matrices of
        the MLs for AM and CM respectively, L_A →
        An m x 1 matrix with activity lengths
/* m = number of activities */;
m = length(A);
/* n = number of sensors */;
n = length(S);
/* The sensor frequency SF(s_k | M_ai)
estimation */;
for l ← 1 to length(I) do
    /* Determine the i^th activity */;
    i = getActivityIndex(I_l, A);
    for k ← 1 to n do
        if IsON(I_l, s_k) then   /* IsON = true if
        k^th sensor is triggered for the
        given instance of an activity */
            SF_i,k = SF_i,k + 1;
        end
    end
end
/* The ML P(s_k | M_ai) estimation */;
for i ← 1 to m do
    L_ai = Σ_{k=1}^{n} SF_ik;
    /* The L_ai is the length of an
    activity a_i */;
    for k ← 1 to n do
        ML_AM_i,k = SF_i,k / L_ai;
    end
end
/* The ML P(s_k | M_c) estimation */;
T = Σ_{k=1}^{n} Σ_{i=1}^{m} SF(i,K);
for k ← 1 to n do
    total = Σ_{i=1}^{m} SF(i,k);
    for i ← 1 to m do
        ML_CM_i,k = total/T;
    end
end
```

Algorithm 2 shows our proposed classifier (based on the Jelinek-Mercer method) which used equation (1). It takes following inputs: The maximum likelihoods of the activity models, i.e., ML_{AM}, the maximum likelihoods of collective models, i.e., ML_{CM}, the list of sensors, S, deployed in the environment, the list of activities we are dealing with, and the list of triggered sensors, Θ, at a given time. It classified most probable activity as output using Naïve Bayesian based classifier.

We only show the algorithm that adopted Jelinek-Mercer smoothing. We omit the similar algorithm that used Bayesian smoothing using Dirichlet priors.

Algorithm 2: Activity classifier (Based on Jelinek-Mercer)

```
Data: ML_AM, ML_CM → Two m x n matrices of
      the MLs for AM and CM respectively, P_A →
      Prior probabilities, S → List of sensors and
      Θ → Activated sensors
Result: C → The classified activity
[m, n] = size(ML_AM);/* m = number of
activities, n = number of sensors                */;
λ = 0.7; /* Smoothing parameter                  */;
for k ← 1 to n do
    if IsActivated(Θ, s_k) then /* IsActivated =
    true if s_k ∈ S is triggered */
        for i ← 1 to m do
            if rs_i > 0 then
                rs_i =
                rs_i *(λ* ML_AM_{i,k} +(1−λ) * ML_CM_{i,k} );
            else
                rs_i* = λ* ML_AM_{i,k} +(1−λ) * ML_CM_{i,k} ;
            end
        end
    end
end
for i ← 1 to m do
    rs_i = rs_i * P_{a_i};
end
C = Max(rs);/* max would return the index
of an activity with the maximum
probability.                                     */
```

5.2.1 Using external input

There are similar activities like "preparing breakfast," "preparing lunch," and "preparing dinner" which have certain time boundaries. A group of same objects are used to pursue these activities. Our inference engine uses time boundary to further distinguish alike activities. It imposes extra weight to an activity if the activity is performed within a prespecified time boundary. For example, if the classifier classifies that the activity is "preparing a meal" and the activity is performed between 8 am and 11 am, then it gives 70% more weight to "preparing breakfast."

6. Evaluation

In order to validate our methods, we performed a series of experiments using three real world data sets. In this section, we present the results of these experiments.

The objectives are to discover

- Whether it is possible to use these smoothing methods to classify activities of daily living
- How accurate the methods would be to classify activities
- How sensitive the classifier's performance is with the different settings of smoothing parameters
- Whether our proposed mechanism yields an improvement in the accuracy of activity classification in comparison with other methods

6.1 Experimental Setup

To evaluate the performance of our methods, we used data gathered by Tapia *et al*. at MIT Place Lab [31] and by Kasteren *et al*. [6]. In MIT's experiment [31], 77 and 84 sensory data collection boards equipped with reed switch sensors were used. The authors deployed these sensors in two single-person apartments and collected data for 2 weeks. The sensors were installed in everyday objects such as drawers, refrigerators, containers to record activation/deactivation events (opening/closing events) as the subject carried out everyday activities. Their data was collected by a base station (BS) and labeled using ESM [32]. Kasteren *et al*. deployed 14 digital sensors in a house of a 26-year-old man. They attached these sensors to doors, cupboards, a refrigerator, and a toilet flush. The data collection lasted for 28 days. A total of 245 activity instances were annotated by the participant with 2120 sensor events. Their data was collected by a BS and labeled using a Bluetooth-enabled headset with speech recognition software installed in the BS. And the activities were chosen from the Katz ADL index [33].

We separated the training and testing data using the "leave 1 day out" strategy. In this strategy, 1 day was used for testing and remaining days were used for training.

As the activity instances were imbalanced between classes, two types of measurements were used to evaluate the performance of our system, similar to [6]. The time-slice accuracy was measured by

$$\frac{\sum_{i=1}^{N} detected_i == true}{N}$$

and the class accuracy was measured by

$$\frac{1}{C}\frac{\sum_{i=1}^{N_c} detected_i == true}{N_c}$$

where N is the total number of activity instances, C is the number of classes, and N_c the total number of instances for class c.

Even though the time-slice accuracy is the typical way of evaluating a classifier's accuracy [6], it is not always the perfect measurement for AR classifiers because the dataset would contain dominant classes that appear a lot frequently than others. For example, the total instances of "toileting" were 114 and the total instances of "dinner" were 10 in the dataset acquired by Kasteren *et al*. And if a classifier correctly classifies 110 instances of "toileting" (accuracy = 96.491%) and 4 instances of "dinner" (accuracy 40%), then the time-slice accuracy would be ≈92%, whereas the class accuracy would be

68%. Therefore, the class accuracy should be the primary way to evaluate an activity classifier's performance. However, in this paper we report both the time-slice and the class accuracy.

6.2 Experiment 1: Activity Recognition Accuracy

The purpose of this experiment was to see how well our methods work to classify activities. We measured the probability that an activity is correctly classified for the duration of a labeled activity, similar to [31].

To see the impact of external inputs, two types of settings were considered: With external inputs and without external inputs. The results we obtained are shown in Figures 3, 4, and 5. The summary of the time-slice and the class accuracy is shown in Table 3. The external inputs (i.e., time boundaries) were not used for the Kasteren *et al*. dataset. However, using an external input gave us around 3% of improvement. The external inputs as shown in Table 2 are used for MIT datasets to make similar activities more distinguishable.

We performed the experiment using both smoothing methods. The method that uses the JM method works well in comparison with the method that uses BS with Dirichlet priors. We set $\lambda = 0.7$ for the JM-based method to give enough probability mass to unseen sensors. And we used $\mu = 10$ for the BS-based method. In subsection 6.3, we discuss more about the effect of smoothing parameters.

6.3 Experiment 2: Varying Smoothing Parameters

The purpose of this experiment was to determine the impact of the smoothing parameters (λ and μ) on the accuracy of activity classification.

For the JM-based smoothing, we ran the test with λ values 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1, and for the BS-based smoothing, we ran the test with μ values 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. The performance results are shown in Figures 6 and 7, respectively.

Table 2: External input

Activity	MIT subject one	MIT subject two
Preparing breakfast	06 am-11 am	05 am-08 am
Preparing lunch	11 am-05 pm	11 am-02 pm
Preparing dinner	05 pm-10 pm	04 pm-08 pm

Table 3: The timeslice and class accuracies for the JM and the BS based smoothing methods(for the JM based smoothing we used $\lambda = 0.7$ and for the BS based smoothing we used $\mu = 7$)

Datasets	Timeslice accuracy (%)		Class accuracy (%)	
	JM	BS	JM	BS
MIT Subject one	72.464	70.29	64.157	63.432
MIT Subject two	66.082	65.497	62.718	62.278
Kasteren *et al.*	89.076	89.496	83.052	82.591

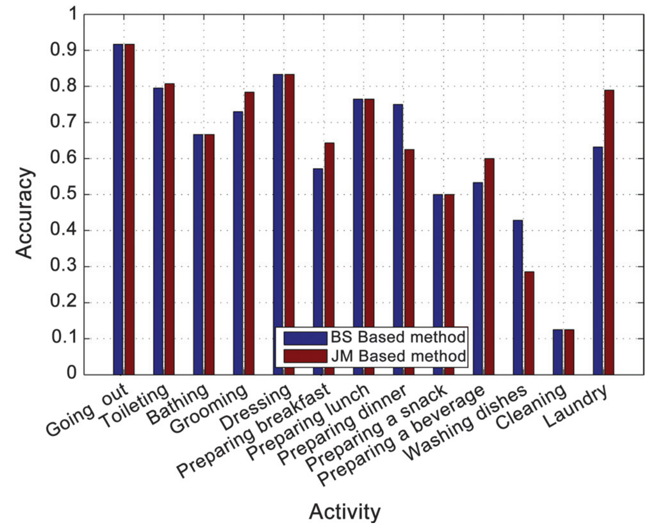


Figure 3: Accuracy for the Place Lab dataset for subject 1.

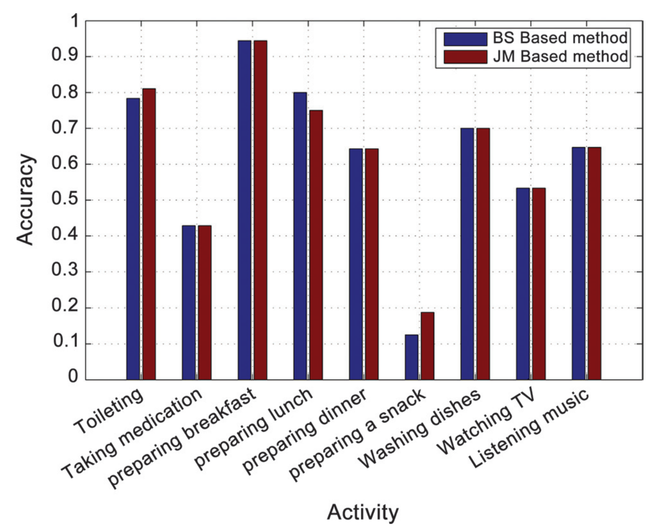


Figure 4: Accuracy for the Place lab dataset for subject 2.

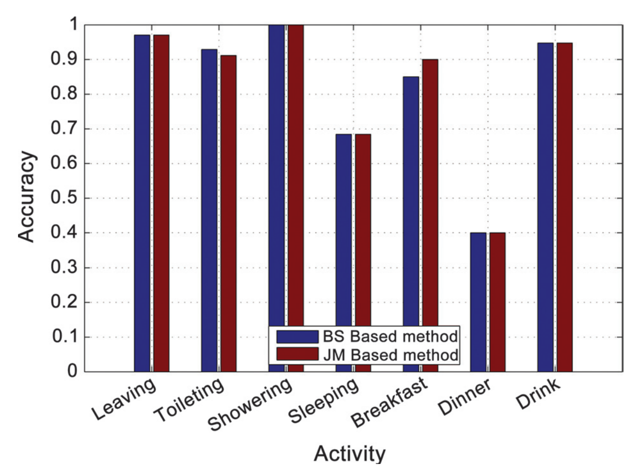


Figure 5: Accuracy for the Kasteren dataset.

Both of the smoothing methods were sensitive to smoothing parameters. However, the BS-based method was not exceptionally sensitive to the different μ values because in this method the movement of probability mass from seen sensors to unseen sensors is activity length dependent. And the activity lengths were sufficiently large with respect to the sensor frequencies to ensure enough smoothing.

The best performance for the JM- and BS-based method was observed for $\lambda = 0.7$ and $\mu = 10$, respectively.

The JM-based smoothing method works well in comparison with the BS-based method.

6.4 Experiment 3: Comparison with Other Methods

The goal of this experiment was to determine how effective the proposed method is in comparison with other methods. We have compared our system with two commonly used classifiers in activity recognition systems, naïve Bayesian (without smoothing) and hidden Markov model.

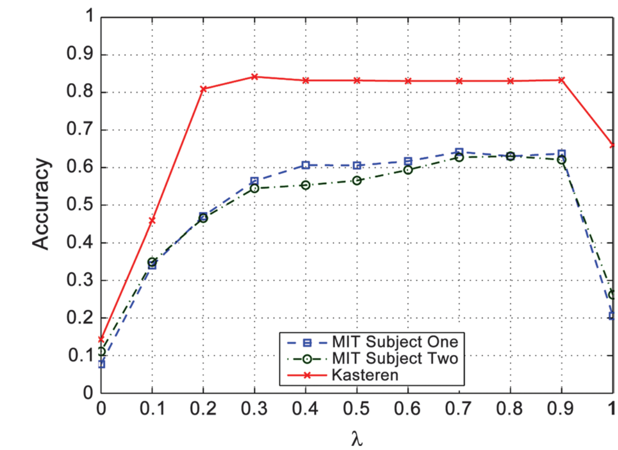


Figure 6: Varying smoothing parameter λ for Jelinek-Mercer.

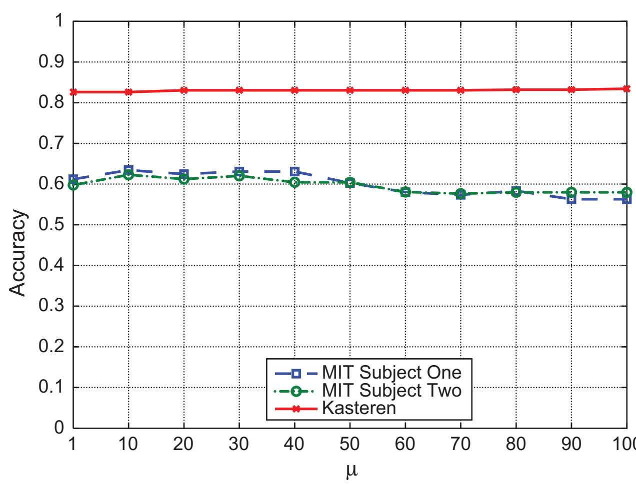


Figure 7: Varying smoothing parameter for Bayesian smoothing.

HMM is a sequential model which is a probabilistic function of the Markov chain as shown in Figure 8.

It consists of a hidden state a_t and the observations θ_t on each state. The hidden state at time t depends on the previous state at time $t-1$. And the observed variable at time t depends on the state at time t. The goal is to find the joint probability distribution:

$$P(a,\theta) = \prod_{t=1}^{T} P(a_t \mid a_{t-1}) P(\theta_t \mid a_t) \quad (3)$$

In HMM, we use a first-order Markov chain to generate a hidden state sequence. That is, given some probability of the first state a_1 and then given a_1, we generate a_2 and so on. For each time we create an output θ_t which is a function of state a_t.

For inference, the Viterbi algorithm is used to determine the label for a new observed sequence [6].

We compare the performance of our proposed methods with the NB based classifier (used by Tapia et al. in [31], without smoothing) using all the three datasets. We also compare with the HMM based classifier (used by Kasteren et al. in [6]). Although Kasteren et al. utilized both HMM and CRF, we only compare with HMM since the classification accuracy (class) of HMM was better than CRF. Figure 9 shows the comparison results. The class accuracy was used to compare the accuracy.

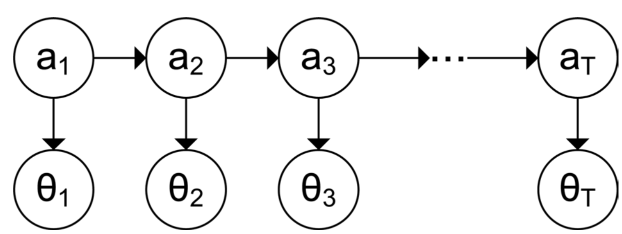


Figure 8: Hidden markov model graphical model.

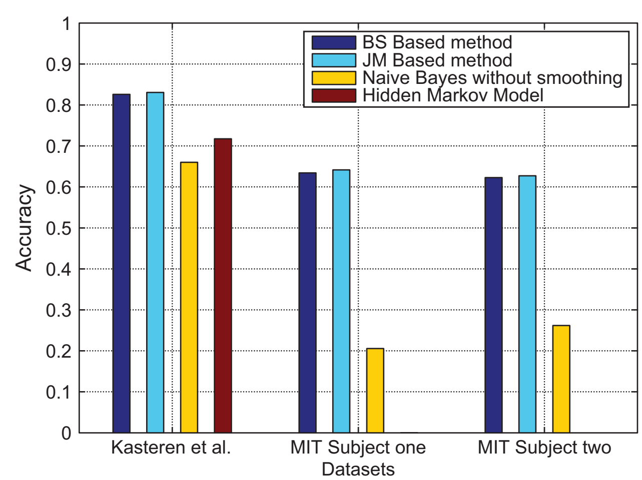


Figure 9: Comparison with other methods.

It is observed that our methods achieved superior results in all cases with respect to other methods. However, for the Kasteren *et al*. dataset, the achievement was not extremely significant in comparison with the NB-based classifier (without smoothing). This is expected because the authors used only 14 sensors, focused only on 7 activities, and the number of unseen sensors was low.

7. Discussion

In the first experiment, we showed how well our smoothing techniques work to classify activities. The corresponding confusion matrices for three datasets are shown in Tables 4, 5, and 6. We only show the confusion matrices generated by the JM-based classifier.

As expected, the classifier made more confusion between the activities which were performed in the same location using similar objects. For example, as we can see in Table 4, the classifier made more confusion between "toileting," "bathing," and "grooming" because most of the instances of these activities were performed in the same location.

In other words, it is observed that groups of similar activities are more separable if performed in different locations. For example, toileting and bathing are more separable in the Kasteren *et al*. dataset (as shown in Table 6) than in MIT's dataset (as shown in Table 4) because "toilet" and "bathroom" are two different locations in Kasteren *et al*.'s dataset.

The classifier without external inputs tends to make more confusion between similar activities which is expected. For example, as we can see in Table 6, the classifier made more confusion between "preparing breakfast" and "preparing dinner" because no external inputs (time boundaries) were used for this dataset. This is opposite for the datasets for which time boundaries were used. Therefore, adding an external input can improve the accuracy of activity determination for similar activities like these.

The proposed classifier is not instance length biased as opposed to the NB (without smoothing) or the HMM-based classifiers. In other words, the classifier has exhibited a high accuracy for an activity with fewer instances in the dataset. For example, the number of instances of "going out" in the MITs dataset for subject 1 was 12 and the classifier correctly classified 11 instances and the accuracy was ≡ 91.667%.

Choosing the right object to embed a sensor is an important factor for the accuracy of activity classification. For example, it is highly likely that a "shower faucet" will be used for "bathing." Therefore, embedding a sensor

Table 4: The Confusion matrix for MIT's dataset for subject one (using JM based smoothing method with $\lambda = 0.7$)

	Going out	Toileting	Bathing	Grooming	Dressing	Preparing breakfast	Preparing lunch	Preparing dinner	Preparing a snack	Preparing a beverage	Washing dishes	Cleaning	Doing laundry
Going out	11	0	0	1	0	0	0	0	0	0	0	0	0
Toileting	0	67	1	11	0	0	0	0	1	1	0	1	1
Bathing	0	4	12	2	0	0	0	0	0	0	0	0	0
Grooming	0	7	0	29	0	0	0	0	0	0	0	0	1
Dressing	0	1	1	1	20	0	0	0	0	0	0	0	1
Preparing breakfast	0	0	0	0	0	9	0	0	4	1	0	0	0
Preparing lunch	1	0	0	0	0	0	13	1	1	0	0	1	0
Preparing dinner	0	0	0	0	0	0	0	5	1	2	0	0	0
Preparing a snack	0	0	0	0	0	2	1	3	7	0	0	0	1
Preparing a beverage	0	1	0	0	1	1	0	0	2	9	0	1	0
Washing dishes	0	0	0	0	0	0	2	0	0	0	2	3	0
Cleaning	0	0	1	1	0	0	2	0	1	0	1	1	1
Doing laundry	1	1	1	0	1	0	0	0	0	0	0	0	15

Table 5: The Confusion matrix for MIT's dataset for subject two (using JM based smoothing method with $\lambda = 0.7$)

	Toileting	Taking medication	Preparing breakfast	Preparing lunch	Preparing dinner	Preparing a snack	Washing dishes	Watching TV	Listening music
Toileting	30	3	0	1	0	0	1	1	1
Taking medication	3	6	1	0	0	2	1	0	1
Preparing breakfast	0	0	17	1	0	0	0	0	0
Preparing lunch	0	0	0	15	0	2	3	0	0
Preparing dinner	2	1	0	0	9	1	1	0	0
Preparing a snack	1	3	1	2	2	3	1	0	3
Washing dishes	2	0	0	1	1	1	14	0	1
Watching TV	3	0	0	0	0	1	2	8	1
Listening music	1	1	0	2	1	1	0	0	11

Table 6: The Confusion matrix for Kasteren *et al.* dataset (using JM based smoothing method with $\lambda = 0.7$)

	Going out	Toileting	Bathing	Go to bed	Break fast	Dinner	Get a drink
Going out	33	1	0	0	0	0	0
Toileting	0	103	4	6	0	0	0
Bathing	0	0	23	0	0	0	0
Go to bed	0	2	4	13	0	0	0
Breakfast	0	0	0	0	18	2	0
Dinner	0	0	0	0	6	4	0
Get a drink	0	0	0	0	1	0	18

in the "shower faucet" would increase the classification accuracy of "bathing." In Kasteren *et al*.'s setup, they did not place any sensor on the "shower faucet." Replacing the "bathroom door" with the "shower faucet" would improve the classification accuracy of "bathing." Also, if we use both of these sensors, the accuracy could be improved further.

8. Summary and Future Work

In this paper, we have proposed two smoothing techniques to improve the accuracy of a NB-based activity classifier using simple and ubiquitous sensors. Our proposed methods eliminate the data sparsity issue of the NB-based activity classifier. These techniques are based on two well-known smoothing techniques used in the language model of information retrieval. We considered home settings where a number of sensors embedded with home appliances (e.g., door, light, faucet, etc.). We performed three experiments to ensure the validity of our models. We demonstrated that our methods can classify activities with a high accuracy (63-83%). We compared our methods with other methods using the same settings. It is observed that our proposed mechanism yielded significant improvement (more than 10-70%) in the accuracy of activity determination in comparison to its existing counterparts.

One of the disadvantages of our proposed AR system is that it requires a dataset for training which would acquire from a home setting in real time. In other words, our system requires to learn from the environment (LFE) to which it will be deployed. Such a technique has severe limitations:

First, the participants are required with an annotation tool to annotate activities. Typical annotation techniques include ESM, a Bluetooth-enabled headset with speech recognition software [6]. When using these tools, participants either carry a PDA (with ESM software installed on it) or wear a headset that is used as a self-reported current persuasion of an activity. In ESM, alerts are given after every time interval (prespecified), and participants must respond to an alert by annotating their current activity. In a Bluetooth-enabled headset, participants are required to annotate their activity via dialogs. Despite the simplicity of these techniques, they have several disadvantages [34]. Wearing a headset or carrying a PDA may appear cumbersome to participants. If the participants are not familiar with the system or device, it might be intimidating. Participants have to carry the device throughout the study period. Other challenging issues involve marking activity end points, data storage stability, and relying on participants to charge the device. A more sophisticated annotation technique like using video cameras could also be used [35]. However, the technique is computationally very expensive and also violates user's privacy.

Secondly, in any environment (house, hospital, or office) there could be hundreds or even thousands of activities. Given such a large number of activities and users without expert knowledge, it is impossible to annotate all the activities. Even if it is possible by experts, it will be very costly and therefore not feasible.

Therefore, an alternate and unique approach is required to train an ARS. The advancement of Internet and World Wide Web (www) encourages millions of users to promote billions of web pages of varieties of contents [36]. Fortunately a fraction of these pages describe in details how to perform daily activities. For example, the web page http://www.wikihow.com/Cook illustrates the sequence of steps required for cooking. Web pages like these portray how to just do everything. They not only state the activity but also depict what objects to use for a particular activity, how to use them, and in what sequence.

Our future goal is to develop an ARS that would learn its model parameters from www. Such a system would grab activity pages like http://www.wikihow.com/Cook and discover the relationship between activities and object (with the embedded sensors) usage. We are planning to use Google (the web search engine) to extract such associations (a quantified relative Google semantics) and translate these relationships into mathematical activity models.

The advantages of such system would be as follows:

- Elimination of the required amount of human effort in labeling activities while maintaining the high recognition accuracy
- A large amount of activities in different environments (e.g., home, office, and hospital) would be recognized
- The system would be scalable by nature
- It would be the least expensive solution.

Our initial investigation shows that such a technique would also suffer from the zero-probability problem. Therefore, we will use our proposed smoothing techniques to eliminate such estimation problem.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0002)).

References

1. D.J. Patterson, L. Liao, D. Fox, and H.A. Kautz. "Inferring high-level behavior from low-level sensors," In: A. K. Dey, A. Schmidt, and J.F. McCarthy, editors., Ubicomp, ser. Lecture Notes in Computer Science, vol. 2864. Springer; pp. 73-89, 2003.
2. M.R. Hodges, and M.E. Pollack. "An 'object-use fingerprint': The use of electronic sensors for human identification," In: Krumm, G.D. Abowd, Seneviratne, and T. Strang, editors., Ubicomp, ser. Lect. Notes. Comp. Sci. Jvol. 4717. Springer; pp. 289-303, 2007.
3. D.H. Hu, S.J. Pan, V.W. Zheng, N.N. Liu, and Q. Yang. "Real world activity recognition with multiple goals," In: UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing. New York, NY, USA: ACM, pp. 30-9, 2008.
4. D.H. Hu, and Q.Yang. "Cigar: Concurrent and interleaving goal and activity recognition," In: AAAI, D. Fox and C.P. Gomes, editors. AAAI Press; pp. 1363-8, 2008.
5. T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, B. Harrison, and B. Hemingway, *et al*. "The mobile sensing platform: An embedded activity recognition system," IEEE Pervasive Computing, vol. 7, no. 2, pp. 32-41, 2008.
6. T. van Kasteren, A. Noulas, G. Englebienne, and B. Kr¨ose. "Accurate activity recognition in a home setting," in UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing. New York, NY, USA: ACM, pp. 1-9, 2008.
7. C.D. Manning, P. Raghavan, and H. Schtze. Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press; 2008.
8. C. Zhai, and J. Lafferty. "A study of smoothing methods for language models applied to information retrieval," ACM Trans. Inf. Syst., vol. 22, no. 2, pp. 179-214, 2004.
9. F. Jelinek, and R.L. Mercer. "Interpolated estimation of Markov source parameters from sparse data," in Proceedings, Workshop on Pattern Recognition in Practice, In: E. S. Gelsema and L. N. Kanal, editors. Amsterdam: North Holland: pp. 381-97, 1980.
10. D.J.C. Mackay, and L.C.B. Petoy. "A hierarchical dirichlet language model," Natural Language Engineering, vol. 1, pp. 1-19, 1995.
11. Exploratory research projects. Available From http://techresearch.intel.com/articles/Exploratory/1435.htm. Accessed May 20, 2009.
12. Center for future health -smart medical home overview. Available from: http://www.centerforfuturehealth.com/smart_home/. Accessed May 20, 2009.
13. Aware home research institute at georgia tech. Available from: http://awarehome.imtc.gatech.edu/. Accessed May 20, 2009.
14. Mit house_n. Available from: http://architecture.mit.edu/house_n/. Accessed May 20, 2009.
15. Smart houses info. Available: http://gero-tech.net/smart_homes.html. Accessed May 20, 2009.
16. S.S. Intille, K. Larson, E.M. Tapia, J. Beaudin, P. Kaushik, and J. Nawyn, *et al*. "Using a live-in laboratory for ubiquitous computing research," in Pervasive, ser. Lecture Notes in Computer Science, KP. Fishkin, B. Schiele, P. Nixon, and A.J. Quigley, editors., vol. 3968. Springer; pp. 349-65, 2006.
17. S.S. Intille, J. Rondoni, C. Kukla, I. Ancona, and L. Bao. "A context-aware experience sampling tool," in CHI '03: CHI '03 extended abstracts on Human factors in computing systems. New York, NY, USA: ACM, 2003, pp. 972-3.
18. S.S. Intille, E.M. Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, *et al*. Larson, "Tools for studying behavior and technology in natural settings," in Ubicomp, ser. Lecture Notes in Computer Science, AK. Dey, A. Schmidt, and J. F. McCarthy, editors., vol. 2864. Springer; pp. 157-74, 2003.
19. D.H. Wilson, and CG. Atkeson. "Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors," in Pervasive, ser. Lecture Notes in Computer Science, H.-W. Gellersen, R. Want, and A. Schmidt, editors., vol. 3468. Springer; pp. 62-79, 2005.
20. W. Zajdel, J.D. Krijnders, T. Andringa, and D.M. Gavrila. "Cassandra: Audio-video sensor fusion for aggression detection," in IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS), pp. 200-5, Sept. 2007.
21. Z. Ghahramani. "Learning dynamic bayesian networks," in Adaptive Processing of Sequences and Data Structures, International Summer School on Neural Networks, "E.R. Caianiello"-Tutorial Lectures. London, UK: Springer-Verlag; pp. 168-97, 1998.
22. S.M. Ward, M.P. Lukowicz, S.M. Troster, and M.T. Starner. "Activity recognition of assembly tasks using body-worn microphones and accelerometers," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 10, pp. 1553-67, 2006.
23. C.R. Wren, and E.M. Tapia. "Toward scalable activity recognition for sensor networks," in LoCA, ser. Lecture Notes in Computer Science, M. Hazas, J. Krumm, and T. Strang, editors., vol. 3987. Springer, 2006, pp. 168-85.
24. D.J. Patterson, D. Fox, H. Kautz, and M. Philipose. "Fine-grained activity recognition by aggregating abstract object usage," in ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers. Washington, DC, USA: IEEE Computer Society, 2005, pp. 44-51.
25. G.J. Lidstone. "Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities," Transactions of the Faculty of Actuaries, vol. 8, pp. 182-92, 1920.
26. I.J. Good. "The population frequencies of species and the estimation of population parameters," Biometrika, vol. 40, no. parts 3 and 4, pp. 237-64, 1953.
27. S.F. Chen, and J. Goodman. "An empirical study of smoothing techniques for language modeling," in Proceedings of the 34th annual meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, pp. 310-8. 1996.
28. S. Katz. "Estimation of probabilities from sparse data for the language model component of a speech recognizer," Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 35, no. 3, pp. 400-1, Mar. 1987.
29. H. Ney, U. Essen, and R. Kneser. "On structuring probabilistic dependencies in stochastic language modeling," Computer Speech and Language, vol. 8, pp. 1-38, 1994.
30. J. Han. Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2005.
31. E.M. Tapia, S.S. Intille, and K. Larson. "Activity recognition in the home using simple and ubiquitous sensors," in Pervasive, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds., vol. 3001. Springer; pp. 158-75, 2004.
32. M. Csikszentmihalyi, and R. Larson. "Validity and reliability of the experience-sampling method." J Nerv Ment Dis, vol. 175, no. 9, pp. 526-36. Available from: http://view.ncbi.nlm.nih.gov/pubmed/3655778 [cited on 1987 Sep].
33. S. Katz, T. Down, H. Cash, *et al*. "Progress in the development of the index of adl." Gerontologist, vol. 10, pp. 20-30, 1970.
34. S. Consolvo, and M. Walker. "Using the experience sampling method to evaluate ubicomp applications," IEEE Pervasive Computing, vol. 2, no. 2, pp. 24-31, 2003.
35. B.Logan, J. Healey, M. Philipose, EM. Tapia, and SS. Intille. "A long-term evaluation of sensing modalities for activity recognition," in Ubicomp, ser. Lecture Notes in Computer Science, J. Krumm, G. D. Abowd, A. Seneviratne, and T. Strang, editors., vol. 4717. Springer, pp. 483-500, 2007.

36. R. Cilibrasi, and PMB. Vitányi. "The google similarity distance," IEEE Trans. Knowl. Data Eng., vol. 19, no. 3, pp. 370-83, 2007.

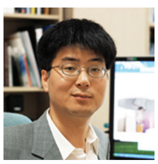
37. A.K. Dey, A. Schmidt, and J.F. McCarthy, editors., UbiComp 2003: Ubiquitous Computing, 5th International Conference, Seattle, WA, USA, October 12-15, 2003, Proceedings, ser. Lecture Notes in Computer Science, vol. 2864. Springer; 2003.

38. J. Krumm, G.D. Abowd, A. Seneviratne, and T. Strang, editors., UbiComp 2007: Ubiquitous Computing, 9th International Conference, UbiComp 2007, Innsbruck, Austria, September 16-19, 2007, Proceedings, ser. Lecture Notes in Computer Science, vol. 4717. Springer; 2007.

AUTHORS

A. M. Jehad Sarkar received B.S. and M.S. degree in Computer Science from National University, Bangladesh, in 1999 and 2000, respectively. Since September 2006, he is pursuing Ph.D degree in Computer Engineering at Kyung Hee University, Korea. From May, 2000 to August, 2006, he served as a Principal Software Engineer in ReliSource Technologies Ltd., Bangladesh. From April, 2000 to April, 2005, he served as a Senior Software Engineer in TigerIT Ltd., Bangladesh. His research interests are Activity Recognition, Web mining, Data mining and Software Engineering.

E-mail: jehad@oslab.khu.ac.kr



Young-Koo Lee received his BS, MS, and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Korea in 1988, 1994 and 2002, respectively. Since 2004, he has been an assistant professor at the Dept. of Computer Engineering, College of Electronics and Information, Kyung Hee University, Korea. From 2002 to 2004, he was a Post Doctoral Fellow Advanced Information Technology Research Center(AITrc), KAIST, Korea, and a Postdoctoral Research Associate at Dept. of Computer Science, University of Illinois at Urbana- Champaign, USA. His research interests are Ubiquitous Data Management, Data Mining, Activity Recognition, Bioinformatics, On-line Analytical Processing, DataWarehousing, Database Systems, Spatial Databases, and Access Methods.

E-mail: yklee@khu.ac.kr

Sungyoung Lee received his B.S. from Korea University, Seoul, South Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, South Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been affiliated with a director of Neo Medical ubiquitous- Life Care Information Technology Research Center, Kyung Hee University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing and applications, Context-aware Middleware, Sensor Operating Systems, Real-Time Systems, and Embedded Systems. He is a member of the ACM and IEEE.

E-mail: sylee@oslab.khu.ac.kr

DOI: 10.4103/0256-4602.60164; Paper No TR 164_09;