

IPARS: Intelligent Portable Activity Recognition System via Everyday Objects, Human Movements, and Activity Duration

Chi-yau Lin, and Yung-jen Hsu

National Taiwan University
Department of Computer Science and Information Engineering
#1 Roosevelt Rd. Sec. 4, Taipei, Taiwan 106
{r93129, yjhsu}@csie.ntu.edu.tw

Abstract

In this paper we define a general framework for activity recognition by building upon and extending Lempel–Ziv multiway tree structure called tries to model human activities of daily living (ADLs). Our activity recognition system is performed online. We show a wearable wrist, based on Radio Frequency Identification (RFID) reader to detect everyday objects, and WiFi positioning system installed in our experimental environment to capture human current position. Our activity models are formulated by translating labeled activities (such as grooming) into probabilistic collections of a sequence of action steps (such as brushing teeth → combing hair), and each action step (such as making tea) is composed of a sequence of human handling object terms (such as cup → teabag → electric air pot → teaspoon), and human movements (bedroom → kitchen). Given RFID tags, WiFi signals, and Passing of time can directly yield the state of the physical world. We experimentally validate our approach using data gathered from actual human activity.

Introduction

Activity Recognition and context-aware computing are gaining increasing interest in the AI and ubiquitous computing communities. Most existing systems have been focused on relatively low level activities with small environments or during short periods of time. In this paper, we describe a system that can recognize high level activities (*e.g.*, grooming, working, cooking, eating, and leisure), and low level activities called actions (*e.g.*, brushing teeth, working at computer, making tea, drinking fluids, and watching TV) over many weeks.

Our system: Intelligent Portable Activity Recognition System (IPARS) used wearable RFID reader to detect everyday observed objects similar to those described in (Philipose et al. 2004), but we enhanced the relation among held objects instead of just considering involved objects in

a certain activity. IPARS considers (1) sequential information, such as which object follows which object; (2) temporal information such as time of day; (3) spatial information extracted from geographic databases, including information about the kinds of activities in various locations; (4) global constraints such as the number of different homes or workspaces. Additionally, IPARS uses data collected by other users so as to improve the classification of a specific user’s activities.

IPARS builds upon previous work on extracting places from traces of users’ movements described in (Bhattacharya et al. 2002) or other activity recognition technologies (Philipose et al. 2004) and (Perkowitz et al. 2004). Our work goes beyond theirs in that IPARS also recognizes the activities associated with the places. Moreover, previous approaches to modeling personal routines and information patterns require a large amount of training data from each user, and cannot be generalized to new places or new users. By contrast, our approach requires less individual training data by leveraging data collected by others. In addition, Markov models used in their works are often constructed with states dictated by the syntactic structure of directions includes how-to from website. Transition probabilities for a typical user to touch a specific next object are arbitrarily assumed. This is far from being practical. In summary, the contributions of this paper are:

- A general framework for sensor-based activity recognition based on Lempel-Ziv family of algorithms, which are expressive and well-suited for discriminative learning.
- An extension of LZ78 algorithms to incorporate various features using aggregating and label-specific classes of activities.
- An extension of LZ78 to arrange the drawback that the convergence rate to optimal model is slow using keeping track of all possible contexts and maintaining a “more complete” activity model.
- A variable length of sliding window based on activity duration for inference in extended LeZi is efficient.

IPARS in Section 2. Details about how to model human activities will be discussed in Section 3, followed by experimental evaluations. Conclusions and future work are given in Section 5.

Activity Recognition System

In this section, we describe how IPARS is intended to be used. We then describe the high-level structure of IPARS, and place the model builder in context.

Meaning of Contexts Chosen in IPARS

Three contexts are adopted in the system architecture. First is *Sequential handling Objects*: RFID technology is used to sense everyday objects. Second is *Human Position*: WiFi positioning system supports to capture human location. The last is *Activity Duration*: Real-time response helps the system with inference and on-line learning.

Sequential Handling Objects. Fortunately, everyday objects are easy to obtain. The only difficulty is to deploy the RFID Tags to every-day objects. Objects with RFID tags can easily be sensed via our wearable RFID Reader wrist. A sequence of handling objects means a lot in our daily life.

Human Position. A person’s movement plays an important role in IPRAS. Some everyday objects are difficult for attaching RFID tags (such as bedroom, desk, chair is too big, and no obvious regions are suitable to put RFID tags). The other two factors are the environment (such as moisture, heat, etc) and high usage frequency of everyday objects (such as comb, cup, keyboard, mouse, etc). These two factors cause the attached RFID tags deformed, and hard to detect efficiently. Therefore, Human movements provide another feature for IPARS to support the inferred results to modify the loss of held objects information.

Activity Duration. Different Activities are done with different period of time used. In addition, we avoid the possibility of touching certain everyday objects uncaringly, because the time used to touch them is too short to construct an action. So, the reason why time is chosen is obvious, and therefore the duration of an activity is added to enhance the power of inference for activity recognition.

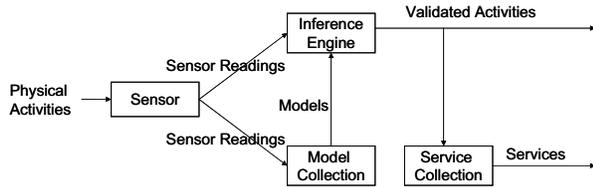


Figure 1: High-level View of IPARS

System Overview

Figure 1 presents the main components of IPARS. It is centered on an inference engine which given models for activities, and sequence of sensor readings, return the validated activities of current state.

Models. We now describe our model of activities. Figure 2 shows how activities are modeled. Each activity model composed of a sequence $A_1 \dots A_n$ of action steps in the activity. The main difference between activity and action is that activity is a set of sequential actions. For example, while we are grooming (taken as an activity) in the morning, some actions, such as “brushing teeth”, and “brushing hair” should be involved in grooming normally.

The reason why we divide an activity (grooming) into some actions (brushing teeth, and brushing hair) is because the more an activity is complex, the more difficult that systems can recognize. So, actions are used to enhance the accuracy of activity recognition.

In IPARS’s tree-structured activity models, each action maintains a handling objects trie and human movements trie.

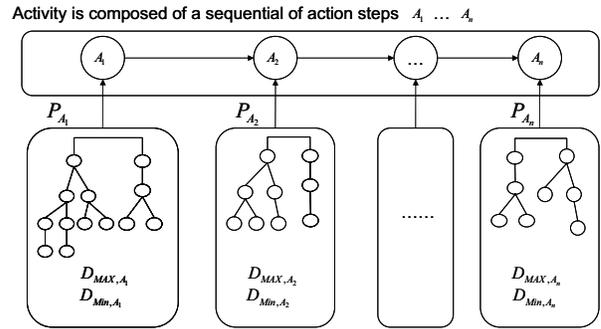


Figure 2: IPARS Generic Activity Model

In following section, we show all the details in IPARS’s activity model. We have deliberately chosen to keep our models simple, because they are faster to reason with. A key result of this paper is that even these simple models are quite effective, while supporting automated extraction and real-time tracking.

Inference Engine. The inference engine infers the current activity by collecting sensor readings and maintaining activity models learned from users. Figure 1 describes the role that inference engine plays in IPARS.

Activity Model Structure

In this section we first discuss LeZi compression algorithm and our extensions used in IPARS. Then we show how to use them for modeling activities.

The LeZi Compression Algorithm

The LZ78 data compression algorithm, as introduced by Lempel and Ziv (Ziv and Lempel 1978), is an incremental parsing algorithm that introduces exactly such a method for gradually changing the Markov order at the appropriate rate. This algorithm has been interpreted as a universal modeling scheme that sequentially calculates empirical probabilities in each context of the data, with the added advantage that generated probabilities reflect contexts seen from the parsed sequence to the current symbol.

LZ78 is a dictionary-based text compression algorithm that performs incremental parsing of an input sequence. This algorithm parses an input string x_1, x_2, \dots, x_i into $c(i)$ substrings $w_1, w_2, \dots, w_{c(i)}$ such that for all $j > 0$, the prefix of the substring w_j (i.e., all but the last character of w_j) is equal to some w_i for $1 < i < j$. Because of this prefix property, phrased substring (LZ phrase), and their relative frequency counts, can efficiently be maintained in a multiway tree structure called a trie.

Since LZ78 is a compression algorithm, it consists of two parts: the encoder and the decoder. For a prediction task, however, we do not need to reconstruct the parsed sequence and therefore do not need to consider this as an encoder/decoder system. Instead, we simply need to construct a system that breaks up a given string sequence (held object, or movement) of states into phrases (action, or activity).

As the LZ78 algorithm parses the sequence, larger and larger phrases accumulate in the dictionary. As a result, the algorithm gathers the predictability of higher order Markov models, eventually attaining the predictability of the universal model.

Compressed Tree-Structured Activity Models

We will now describe our tree-structured activity models. In IPARS, activity models are formulated by involved possible action sequences. Figure 3(a) shows the relationship between an activity and actions. In Figure 3(a), Activity-A involves a sequence of actions (action1→action2→action3), and Activity-B involves multiway action sequences. Each root of a tree represents an activity, and has all possible action steps. Figure 3(b) shows that each action node in activity model is composed of an objects' trie, a movements' trie and a reasonable activity duration between $D_{Min,Activity}$ and $D_{MAX,Activity}$.

The schema for activity recognition based on temporal and spatial patterns is shown in Figure 4. It includes five classes: *Activity*, *Action*, *Object*, *Location*, and *Duration*.

Activity. Activity is the central class in the domain. Its attribute *Label* is the only hidden variable. The set of possible labels in our experiments is {'grooming', 'doing laundry', 'leisure' and etc.}. Attribute *Action Steps Trie* serves as primary key, and is a reference attribute that points to a sequence of actions when the activity has been performed. The class also contains temporal information associated with an activity, such as *Duration*, *TimeOfDay*,

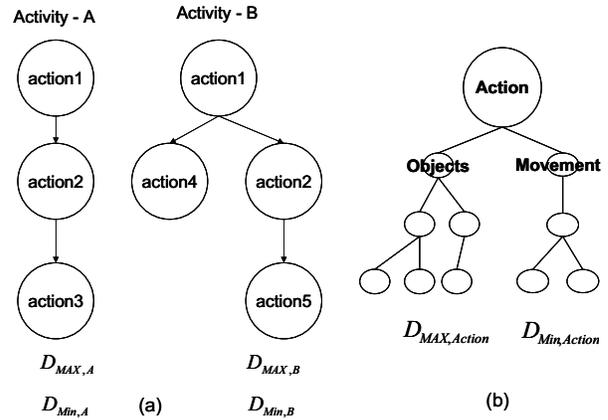


Figure 3: The Relationship between Activity and Action (a) and an Action in Activity Model (b)

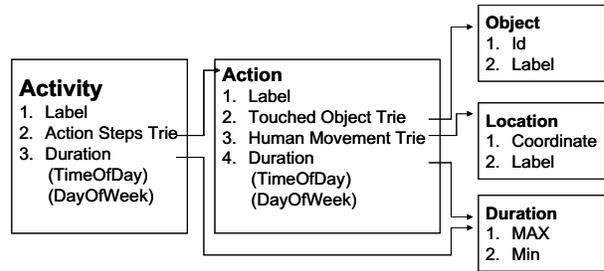


Figure 4: The Schema of the Tree-Structured Activity Model (Solid straight lines indicate reference relations among classes)

and *DayOfWeek*.

Action. Action is a primary class to construct an activity. The attribute *Label* is similar to the label in *Activity* class. The possible labels are {'brushing teeth', 'brushing hair', 'drinking fluids', 'drinking fluids', 'taking medications', and etc.}. The attribute *Touched Objects Trie* is a reference attribute that points to tagged everyday objects, and maintains the sequence of held objects involved in an action. *Human movements Trie* is similar to *Touched Objects Trie*. It points to human location entity, and maintains the sequence of human movements involved in an action. Finally, *Duration* indicates whether the action occurs in the reasonable duration.

Object: The class *Object* includes two attributes: *Id* and *Label*, which translate tagged objects' Id into labels, such as {Id: 01E007000012414756, Label: TV remote control}.

Location: The class *Location* includes two attributes: *Coordinate* and *Label*, which translate human coordinate computed from wireless positioning engine into labels of area, such as {Coordinate: leftUp = (76,194), rightDown = (153,240), Label: living room}.

Duration: *Duration* captures the period of time used in an

action. The reference attribute $MAX(D_{MAX,Action})$ and $Min(D_{Min,Action})$ refer to the maximum and minimum period of time ever used in an action.

Based on the schema, in following sections, we define how to incorporate these discriminative features in IPARS.

Learning and Inference

In IPARS, the task of inference is to estimate the labels of activities given a sequence of objects held by a person, a sequence of locations visited by a person, and a sequence of time passed. IPARS incorporates these three contexts and converts the sequences into corresponding tries built in IPARS. In addition, IPARS that recognizes activities can learn and adjust its internal model in real-time as new examples of activities are available. This will allow the algorithm to adapt to changes in the user’s routines over time.

Learning Process

IPARS provides a learning interface to users and focuses on collecting sensor data and modeling users’ action models. Following describes the learning process in IPARS.

IPARS Learning Interface. Figure 5 (Left) shows the IPARS Learning Interface and Figure 5 (Right) is the IPARS Recognizing Interface. In Figure 5 (Left), users only need to push the start button to train this specific action. The *Information List* shows the translated sensor data such as everyday object, physical area, and current time. IPARS automatically stores and constructs this training data to its model storage. The details will be described below.

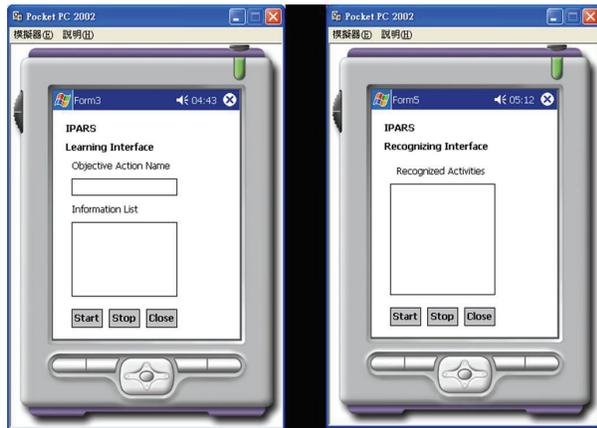


Figure 5: IPARS Learning Interface (Left) IPARS Recognizing Interface (Right)

Action Modeling Process. Figure 6 shows the activity diagram of Action Learning Process. In Figure 6, sequential contexts are divided into three sorts of data, and then *Object Filter* and *Location Filter* filter redundant sensor data such as a series of the same everyday objects, and human locations. In addition, *Touched Objects Trie*

Builder and *Movements Trie Builder* start to build and combine this new training data to original action model. *Time Estimator* not only records the duration used to train this action, but calculate a reasonable duration according to the specific action.

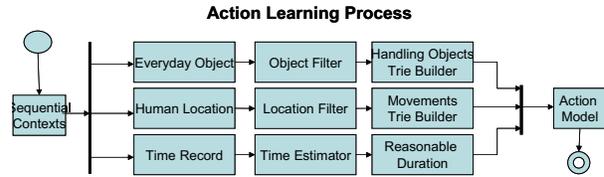


Figure 6: Activity Diagram of Action Learning Process

Supervised, Model-based and Online Learning

When designing algorithms for recognizing activities in real home environments, it is important to consider factors such as the ease of setup and training the system.

Supervised Learning. Homes and their furnishings have highly variable layouts, and individuals perform activities in many different ways. One approach to handle such variety is to use supervised learning. Sensing system must use supervised learning, because the same activity may result in a significantly different sensor activation profile based upon the empirical experiences, or routines, of the home occupant and the layout and organization of the particular home. IPARS can learn *generic* activity models from labeled action sequences of N different users. Learning a *customized* model for an individual user is a special case when $N = 1$.

Model-based Learning. Model-based algorithms use the training examples to construct a model of the target classification function. When a new classification is required, only the model previously generated is used.

Online Learning with fast update. This makes it possible to build systems that learn and update its model in real-time. Online learning offers a possibility for adapting to a person’s activity variations over time.

Learning Methods in IPARS. IPARS that recognizes activities can adjust its internal model in real-time as new examples of activities are available. This will allow the algorithm to adapt to changes in the user’s routines over time. We design a Learning Interface to simplify the learning procedure. Users do not need to know the operation of the system. We just need to input the name of certain action, and then push the start button. In this way, IPARS collects these observed data, and puts them into database. This makes it possible to build systems that learn and update its models in real-time. Online learning offers a possibility for adapting to a person’s activity variations over time.

Inferring Process

In this section, we describe how IPARS recognizes human activity by mapping current sensor data to internal activity models. IPARS makes inferences from incorporating

sequential handling objects, human movements, and the period of passed time. Figure 7 shows the activity diagram of Action Inferring Process. Full details are described from the following.

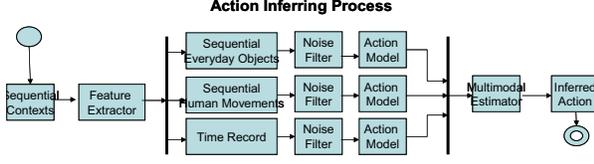


Figure 7: Activity Diagram of Action Inferring Process

Feature Selection

Given a sequence of contexts, IPARS needs to map these sensor data to all the action models. In Figure 7, *Feature Extractor* plays an important role to generate a feature window to capture these sequential sensor data. First, *Feature Extractor* checks if the data matches the initial condition in each action model. If the initial condition is satisfied, *Feature Extractor* generates a feature window with length $D_{MAX,Action}$ to capture all the following sensor data in the feature window. *Feature Extractor* converts a sequence of contexts to some blocks of feature windows. In this way, IPARS can start to analyze the information existing in the feature window.

Incorporating Sequential Handling Objects

In Figure 2, each action node contains a *Touched Objects Trie*. Each *Touched Objects Trie* is composed of all the possible process of handling objects. How to map a captured sequence of everyday objects to according action models is what we want to solve. In Figure 7, *Noise Filter* plays a role to filter all objects which are irrelevant to *Touched Objects Trie* in according action model. In addition, IPARS sequentially checks all the remained objects in the generated feature window, and then compares to the *Touched Objects Trie*. The correlation between remained objects and *Touched Objects Trie* is calculated as:

$$C(A_{t-L:t} | O_{t-L:t}) = \sum_{j=1}^m \left(\frac{d_{Max}}{d_{ij}} \right) \quad (1)$$

where the branch (b_{ij}) and depth (d_{ij}) present the attribute of possible handling objects sequence, i means an index of an action model, and j means an index of a branch in a trie. In addition, d_{Max} means the sequence of observed objects that matches the maximum depth in an object trie, and m is the total number of branches in an object trie. In this way, the equation can be obtained as above.

Incorporating Human Movements

Although we attach RFID tags to everyday object, but we still have problem to recognize that activities for which there were no interactions with objects, especially sleeping when no sensor in the bed. As a result, we collect and translate sequential location data to tree structure. The location sequence shows another feature to represent an activity. IPARS also check the location trie in action models. So, the result of location sequence is similar to the part of the object sequence.

$$C(A_{t-L:t} | L_{t-L:t}) = \sum_{j=1}^m \left(\frac{d_{Max}}{d_{ij}} \right) \quad (2)$$

Incorporating Action and Activity Duration

Different activities have different length of used time. Therefore, the duration of the activities has to be taken into account while developing algorithms for recognizing activities. In order to incorporate the activity duration, one of feature window for per activity to recognize was used, and the length of each window corresponded to the activity duration as carried out by the subject. Thus if M is the number of activities to recognize, there were M different feature windows with lengths $L_1 \cdots L_m$. Our IPARS records the $D_{MAX,Activity}$, and $D_{Min,Activity}$ for each activity. The duration or length of each feature window is $D_{MAX,Activity}$ for each activity. For example, the feature window of the subject for making tea (an action) was 124 seconds ($D_{MAX,MakingTea} = 124$ $D_{Min,MakingTea} = 56$). IPARS records each activity duration with the total amount of involved action nodes. For example, we define 'grooming' is an activity, and 'brushing teeth' ($D_{MAX,BrushTeeth} = 108$ $D_{Min,BrushTeeth} = 80$) and 'brushing hair' ($D_{MAX,BrushHair} = 52$ $D_{Min,BrushHair} = 36$) are involved in 'grooming'. Then while 'brushing teeth' costs 100 seconds, and 'brushing hair' cost 30 seconds, then the reasonable duration of 'grooming' is close to 130 seconds. Reasonable *Activity Duration* of Grooming needs to be between $D_{Min,BrushingTeeth+BrushHair}$ and $D_{MAX,BrushingTeeth+BrushHair}$. We defined *Diff* (Difference between the reasonable duration and obtained duration), and the correlation can be calculated as:

$$Diff = \begin{cases} 0, & \text{if } D_{Min} \leq D_{Real} \leq D_{MAX} \\ \frac{\min(|D_{MAX} - D_{Real}|, |D_{Min} - D_{Real}|)}{|D_{MAX} - D_{Min}|}, & \text{otherwise.} \end{cases} \quad (3)$$

$$C(A_{t-L:t} | D_{t-L:t}) = (1 - Diff) \quad (4)$$

Recognizing the Action

In the recognition step, each feature window (of length L_i)

is positioned in the current time to analyze t . The features are then calculated from time $t - L_i$ to time t . Once the features are calculated, the correlation for the current activity is calculated using IPARS's activity models. The Inferred result of activity recognition is as

$$\alpha + \beta + \gamma = 1 \quad (5)$$

$$C(A_{t-L:t} | O_{t-L:t}, L_{t-L:t}, D_{t-L:t}) = \alpha \cdot C(A_{t-L:t} | O_{t-L:t}) + \beta \cdot C(A_{t-L:t} | L_{t-L:t}) + \gamma \cdot C(A_{t-L:t} | D_{t-L:t}) \quad (6)$$

In (5), the specific weight α is added to the part of sequential handling objects, β for locations sequence, and γ for the part of the duration passed. While the probability of action is described in (6) combing the correlations from three contexts with different weights to infer the results. When $C(A_{t-L:t} | O_{t-L:t}, L_{t-L:t}, D_{t-L:t}) \geq H$ (threshold), IPARS infers this action occurred between $t - L$ and t .

Example. In Figure 8, the activity model, *Grooming*, is learned from the subject's empirical daily living. First, the action, *Brushing Teeth*, costs the subject almost 80 seconds to 108 seconds to finish. Second, another action, *Combing Hair*, costs the same subject about 36 seconds to 52 seconds to complete it. Finally, from the subject's empirical data, IPARS built the activity model of *Grooming*, involving these two sequential action steps, *Brushing Teeth* and *Combing Hair*.

How to infer the activity label by using IPARS's inference engine will be described as follow. First, we capture a sequence of sensed contexts from the subject in Figure 9, and then we starts to introduce the details in each part for activity recognition.

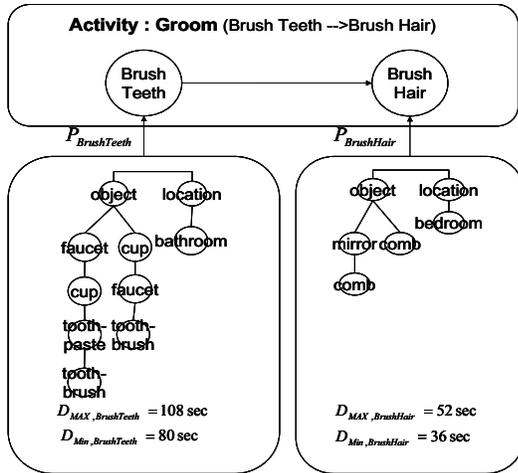


Figure 8: Activity Model of Grooming from the

Subject

Brush Teeth:

(1) *Object Sequence*: While inference engine sensed a object from the sensed object sequence, it starts to check the initially involved object in all the action models. In Figure 9, inference engine captures 'cup' and maps it to the action model (Brush Teeth) in Figure 8. The inference engine then generates a feature window of length: $L_{MAX,BrushTeeth}$ to capture all the following objects in the feature window. In Figure 9, four objects are included in the feature window between $(t - L_{MAX,BrushTeeth})$ and (t) .

Then, IPARS starts to check if any branch in object trie of 'Brush Teeth' matches the objects sequence captured by the feature window. Generated feature window contains the sequence: cup \rightarrow faucet \rightarrow toothpaste \rightarrow toothbrush and it satisfies to complete a branch in action model during a specific time between $(t - L_{MAX,BrushTeeth})$ and (t) .

	$(t - L_{MAX,BrushTeeth})$	(t)	$(t - L_{MAX,BrushHair})$	(t)
Object Sequence	...	cup, faucet, toothpaste, toothbrush	mirror, comb, TV remote control, cup...	
Location Sequence	...	bathroom	bedroom, living room, kitchen...	
Inferred Results	...	Brush Teeth	Brush Hair, Watch TV, Drink Fluids	

Figure 9: One Observed Sequential Data from the Subject

Because inference engine filters the noise: toothpaste, the sequence: cup \rightarrow faucet \rightarrow toothpaste \rightarrow toothbrush satisfies the objective sequence: cup \rightarrow faucet \rightarrow toothbrush in the subject's action model. In this way, the correlation of 'Brush Teeth' can be calculated using (1).

$$C(BrushTeeth_{t-L:t} | O_{t-L:t}) = \sum_{j=1}^m \left(\frac{d_{Max}}{d_{ij}} \right) = \frac{3}{3} = 1$$

(2) *Location Sequence*: In Figure 9, IPARS generates the same length of a feature window between $(t - L_{MAX,BrushTeeth})$ and (t) to capture all the location data at the same time. In the feature window, only 'bathroom' is involved in the duration. According to the built activity model in Figure 8, it's obvious that the correlation between location data and action model (Brush Teeth) can be calculated by (2).

$$C(BrushTeeth_{t-L:t} | L_{t-L:t}) = \sum_{j=1}^m \left(\frac{d_{Max}}{d_{ij}} \right) = \frac{1}{1} = 1$$

(3) *Passed Duration*: In the real experiment, the passed duration of 'Brush Teeth' computed by inference engine

costs 78 seconds, but the reasonable duration should be between 80 seconds and 108 seconds shown in Figure 5. So, IPARS computes the correlation of ‘Brush Teeth’ using the mentioned equation (3) and (4), we can get the results as:

$$Diff = 0.07$$

$$C(BrushTeeth_{t-L} | D_{t-L}) = 0.93.$$

From (5), we choose the specific weights to measure the results. First, $\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$ and $H = 0.5$, the weighting value is decided from the power of which context can yield the correct inference directly. Because of $C(BrushTeeth_{t-L} | O_{t-L}, L_{t-L}, D_{t-L}) = 0.979$, the action of ‘Brush Teeth’ is considered ‘happened’ during $(t - L_{MAX, BrushTeeth})$ and (t) .

Brush Hair:

In the same way, IPARS infers the action ‘Brush Hair’ ‘occurred’ in the duration between $(t - L_{MAX, BrushHair})$ and (t) .

Activity Recognition:

Then IPARS will translate the handling objects sequence (... , cup → faucet → toothpaste → toothbrush → mirror → comb ...) and (... , bathroom → bedroom ...) into two sequential actions (Brush Teeth → Brush Hair). As a result, the activity (Groom) satisfies the activity model described in Figure 5. Inference engine recognizes the activity (Groom) occurred between $(t - L_{MAX, BrushTeeth})$ and (t) .

Problems Overcome. Activities can occur sequentially, in parallel, alternating, and even overlapping, may cause lots of difficulties for systems to recognize these activities. IPARS overcomes these difficulties with the compressed tree-structured activity models.

Sequentially. First, in IPARS, we capture the period of time used in an activity. In the activity duration, IPARS knows if activities occurred consecutively in their reasonable duration, one after another.

In Parallel and Alternating. Second, IPARS overcomes the problem of parallel activities using a dynamic sliding window to scan all observed sensory data to capture when the activity starts, and ends. In Figure 3(a), Activity-A and Activity-B may occur in parallel. The sliding window scans when the first action of an activity starts, and ends in its reasonable duration. So, for a long period, we assumed Activity-A occurred from 7:30am to 8:00am, and Activity-B occurred from 7:45am to 7:50am. Then we can know Activity-A and Activity-B occurred in parallel.

Overlapping. Take Activity-A as “toileting”, and Activity-B as “watching TV”. These two activities cannot occur in parallel, alternating, or overlapping in regular situation. If the boundaries between activities are fuzzy, and difficult to

know when one activity ends and another starts, IPARS uses helpful location information to filter the uncertainty existed in “watching TV” and “toileting”. Because human cannot have more than one position (in living room and in bathroom) at a time point, that is why we add location information to support IPARS to recognize activities.

Evaluation

Study and Data Collection

In this section, we’ll describe that the study was run in home to collect data in order to develop and evaluate the activity recognition algorithms. The subject was a student 25-year-old who spent free time at home. The subject lived alone in one-bedroom apartment and RFID tags were installed in the subject’s apartment.

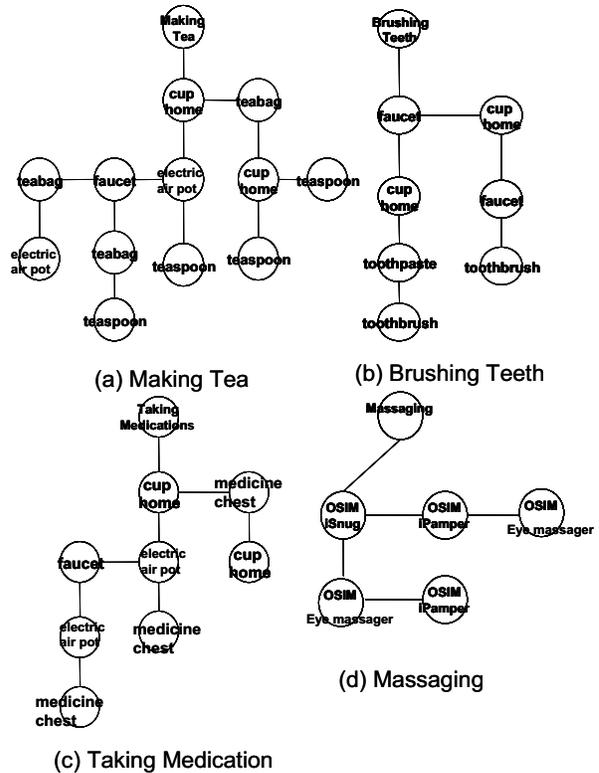


Figure 10: Four Action Models from the subject

Labeling Subject’s Activities. In the home environment, direct observation is prohibitively time-consuming and invasive. After carefully considering possible methods for labeling the subject’s activities, self report (time diaries) was selected. To minimize selective recall and selective bias, time diaries can be used. Participants write down what they do during the day either as they do it or at regular, finely-spaced intervals. Diaries provide better data than recall surveys but are burdensome for the user, and we get the activity labels from the diaries.

IPARS Performance

Table 1 shows the list of activities used in the study. In Figure 10, IPARS demonstrated four chosen action models learned from the subject’s activity models. Figure 11 (a), (b), (c), and (d) show the models for us to analysis the strengths and weaknesses of the algorithms in IPARS.

Table 1: List of Activities used in the studies

Activity	Action	Involved Everyday Objects
1. Leisure	1. Watching TV	TV remote control
	2. Massaging	OSIM_eye massager, OSIM_iPamper, OSIM_iSnug
2. Eating	3. Drinking Fluids	cup, electric air pot
	4. Making Tea	cup, electric air pot, teabag, teaspoon, faucet
3. Grooming	5. Brushing Teeth	faucet, cup, toothpaste, toothbrush
	6. Combing Hair	mirror, comb
4. Medical	7. Taking Medicine	cup, electric air pot, faucet, medicine chest
5. Working	8. Using Computer	keyboard, mouse
	9. Reading	desk lamp, book
6. Contact	10. Using Phone	mobile phone, phone

Figure 11 shows the performance of recognizing actions. The result of “Brushing Teeth” is not so good. The main reason is because the subject trained the action just two times, so IPARS is hard to recognize “Brushing Teeth” while there are lots of possible variations in person’s activity. The main solution is that he needs to train an activity more than four times.

In Figure 11 the result of “Taking Medication” is not so good, too. The reason is because the short period of time used in the activity. The subject takes medication in a short time. The action only costs him 10 seconds to 30 seconds.

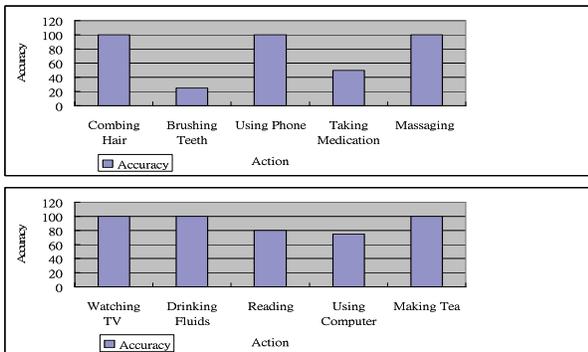


Figure 11: Accuracy of Low-level Activity Recognition

The solution to conquer the problem also needs the subject to train the system.

Conclusions and Future Work

In this paper, we present a discriminative relational approach for activity recognition based on the framework of multi-tries LeZi models, which are well-situated to model sequence of contexts for activity recognition. We showed how to perform efficient inference and learning using a mixture of discriminative contexts.

Using our approach, we developed and tested a specific model for everyday object-based, location-based, and duration-based activity recognition. The results are very promising: the system is able to learn models that can accurately label human activities solely based on handling objects and WiFi location data. We demonstrated that sequential, temporal and spatial features are very important to achieve good recognition rate. We also want to show how to obtain good priors using data from other people so as to learn an improved model for a specific person that requires less labeled data.

We plan to extend our model in a number of ways. First, by collecting data from more subjects, we can learn a set of generic models by clustering the subjects based on their similarities; then we can use a mixture of these models to better recognize activities of a new person. Second, we will integrate information from other wearable sensors (e.g. microphones or accelerometers) into our general framework, thereby enabling much finer-grained activity recognition. Third, activity recognition can focus on many different issues, such as *Reminding* reported in (Pollack et al. 2003), *Service-Providing* described in (Cook et al. 2003a, 2003b), and etc. We will focus on the issue about *healthcare* for elders. How many fluids do I drink today? Do I take medication on time? Do I do enough exercise today? These are important and interesting issues about healthcare.

Acknowledgements

This work is partially supported by the National Science Council of Taiwan, R.O.C, under grant NSC-93-2218-E-002-148.

References

- Bhattacharya, A.; and Das, S. K. 2002. LeZi-Update: An Information-theoretic approach for personal mobility tracking in PCS networks. *ACM-Kluwer Wireless Networks (WINET)* 8(2):121-137.
- Cook, D. J.; Youngblood, M. Heierman, E. Gopalratnam, K.; Rao, S.; Litvin, A.; and Khawaja, F. 2003a. MavHome: An Agent-Based Smart Home. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, 521-524.

- Cook, D. J.; Das, S. Gopalratnam, K; and Roy, A. 2003b. Health Monitoring in an Agent-Based Smart Home. In *Proceedings of the International Conference on Aging, Disability and Independence Advancing Technology and Services to Promote Quality of Life*.
- Perkowitz, M.; Philipose, M.; Patterson, D. J.; Fishkin, K. P. 2004. Mining Models of Human Activities from the Web. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, 573-582.
- Philipose, M.; Fishkin, K. P.; Perkowitz, M.; Patterson, D. J.; and Hhnel, D.; Fox, D.; and Kautz, H. 2004. Inferring Activities from Interactions with Objects. *IEEE Pervasive Computing* 3(4):50-57.
- Pollack, M. E.; Brown, L.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; and Tsamardinos, I. 2003. Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems* 44:273-282.
- Ziv, J.; and Lempel, A. 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory* 24(5):530-536.
-