Yang J, Lee J, Choi J. Activity recognition based on RFID object usage for smart mobile devices. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(2): 239–246 Mar. 2011. DOI 10.1007/s11390-011-9430-9

Activity Recognition Based on RFID Object Usage for Smart Mobile Devices

Jaeyoung Yang[1], Joonwhan Lee[2], and Joongmin Choi[3]

[1]*Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, U.S.A.*

[2]*Neowiz Internet Inc., Seoul, Korea*

[3]*Department of Computer Science and Engineering, Hanyang University, Ansan, Gyeonggi-Do, 426-791, Korea*

E-mail: {isconan, joonwhan}@gmail.com; jmchoi@hanyang.ac.kr

Received July 12, 2010; revised January 31, 2011.

Abstract Activity recognition is a core aspect of ubiquitous computing applications. In order to deploy activity recognition systems in the real world, we need simple sensing systems with lightweight computational modules to accurately analyze sensed data. In this paper, we propose a simple method to recognize human activities using simple object information involved in activities. We apply activity theory for representing complex human activities and propose a penalized naive Bayes classifier for performing activity recognition. Our results show that our method reduces computation up to an order of magnitude in both learning and inference without penalizing accuracy, when compared to hidden Markov models and conditional random fields.

Keywords activity recognition, activity theory, context-awareness, RFID

1 Introduction

Understanding human activity is very important for ubiquitous computing applications. In order to understand human activity, ubiquitous computing applications need a wide range of information such as long-term plans, regular routines for daily life[1], personal preferences, and sensor information management[2-4].

Activity recognition is a core building block for understanding human activity. Activity recognition is the ability to recognize a human's current activity based on information from various sensors such as physiological sensors, cameras, and RFID sensors. Many researchers use statistical methods to induce model encoding patterns in the data[5-6]. Inducing patterns using machine learning algorithms is promising, but the complexity of such approaches is quite high in real systems with several sensor streams and numerous activities being classified.

In this paper, we propose a *simpler* approach based on activity theory, which is able to represent activities over a range of timescales, from long-term (i.e., hours) to short-term (i.e., minutes). This approach can accurately recognize activities using information about the objects involved in the activities while limiting computational complexity. In this paper, we describe both the approach and our experimental results to demonstrate its accuracy.

The contribution of this paper is summarized as follows: the development of a computationally lightweight classifier that does not sacrifice accuracy for smart mobile devices. In order to achieve this goal: 1) we operationalize activity theory by creating appropriate computational constructs (activity theory is typically not used directly in computation despite having good computational properties); 2) we convert raw sensor data into semantic constructs, which are tightly coupled with factors such as location, time, and objects affected by human behavior, and the operational activity theory to simplify the representation of sensed data without loss of information; 3) we introduce a penalized naive Bayes classifier, in which the performance of the classifier is compensated for using penalty functions, in terms of these factors, to reduce the chance of misclassification. Using a penalized naive Bayes classifier with operationalized activity theory, we can obtain our lightweight classifier with high accuracy.

2 Related Work

We can divide previous work in activity recognition into two areas: 1) developing cheap physical sensors to

Short Paper

This work was supported by the Korea Research Foundation under Grant No. KRF-2008-357-D00221.

collect all activities done by the user and 2) building sophisticated computational models to classify or predict activity with a small amount of collected activities.

Regarding sensors, many researchers have focused on developing cheap or new sensors to capture as many human actions as possible. For improved understanding of human activity, researchers continue to try to use various sensors such as cameras[7-8], accelerometers[9], RFID sensors[10-11], state change sensors[6], and GPS[12]. As the number of sensors increases, we reduce the gap between real human life and computational models of human life based on collected user's behavior data from sensors. However, the computational load to induce these general models that encode activity patterns dramatically increases.

Regarding computational models, researchers have focused on analyzing gathered data from sensors to extract patterns for activities. Models used for recognizing activities fall into three categories: probabilistic models, logic-based models[13], and hand-crafted models[14]. In this paper, we focus our attention on probabilistic models such as naive Bayes classifier (NBC)[6], hidden Markov models (HMM)[15], conditional random fields (CRF)[16], and dynamic Bayesian networks (DBN)[5]. Probabilistic models are flexible and easy to extend. With enough appropriate training data, we can obtain high recognition performance.

NBC is an unconditional graphical model. It uses an assumption of strong independence between random variables. This assumption provides computational benefits, but can also decrease performance.

HMM and CRF are popular methods in analyzing and tagging sequential data. HMM is a generative model and CRF is a discriminative one. They are good approaches but they are not appropriate for learning human activities because, while human activities seem sequential, they are not. High-level activities, such as going to the grocery store or coming home, are sequential and there are relations between high-level activities. However, low-level actions, such as making a phone call while cooking, can unexpectedly involve unconsciously interleaved operations that have occurred in another action. This causes an increase in the number of states making HMM and CRF intractable.

DBNs are directed acyclic graphical models that deal with temporal information. DBNs are able to represent cause-effect relationships and are flexible in principle. However, there are many cyclic events in reality (e.g., getting into bed, going to the bathroom), and DBNs cannot be used to encode joint probability between actions or events.

These statistical models can build reasonable prediction models but require many qualified training examples and intensive computation. These models, then, are not applicable for running on lightweight, mobile devices that users typically carry, and for realistic smart environments with many sensor streams. In the next section, we will further describe the activity recognition problem we are interested in.

3 Problem Statement

Fig.1 illustrates the relationship between activities and state or environment. A human performs activities in order to change the state or environment according to his/her conscious goal. The environment can limit the activities that the human can choose to perform. In this case, humans take additional allowable actions leading to intermediate states to achieve a desired goal.

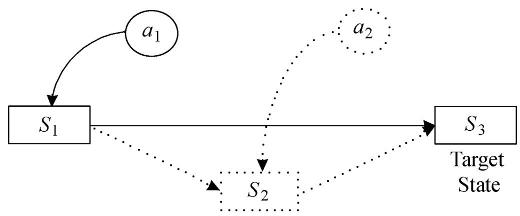

Fig.1. Sequence of actions for given states: S depicts state, and a represents a selected action for a given state.

Activity recognition is designed to learn a series of possible actions that change a given state to a target state for each conscious goal. In Fig.1, for example, we can learn action a_1 or $\{a_1, a_2\}$ depending on the environment for a given state. In this paper, our goal for the activity recognition system is also extraction of a sequence of actions called operations that are tightly coupled with environment context information.

Other research[10,15-16] has shown that there are relationships between actions. This means that action a_n can affect another action a_{n+1}. While existing work has focused on the influence of preceding actions on the current action, our approach is to leverage environmental information on things, such as objects, which humans touch while taking actions, or location where activities take place.

In this paper, we treat actions as being independent from each other, and treat object and location information as having a strong dependence on an action. Fig.2 illustrates the relationship between action and environment information. Therefore, in this paper, the problem we are focusing on is how to extract patterns, in which actions are independent from each other and actions are dependent on environment information, such as object and location information. This information is inferred from large amounts of sensor data. In the next

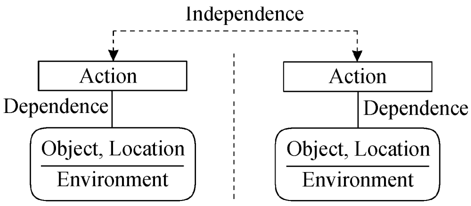

Fig.2. Relationship between actions, objects, location, and environment.

section, we describe activity theory and our approach for operationalizing it.

4 Activity Recognition Based on Activity Theory

4.1 Activity Theory

The concept of activity theory originated with Vygotsky, Leontjev, and Lurija, three psychologists from the former Soviet Union. In activity theory, the fundamental unit of analysis is human activity having three basic characteristics: objects, tools that shape the way human beings interact with the world, and communities. An activity is a form of *doing* directed to an object, and activities are distinguished from each other according to objects used. This means that the object forms the basic relationship between human activity and the environment[16].

In activity theory, human activity is described as a hierarchy with three levels: activity, action, and operation. Activities represent a motivation such as `take a rest`. Activities are realized through individual actions or chains of actions. Actions are generally conscious and are achieved through a series of operations. In contrast to actions, operations are habitual and subconscious routines made according to the conditions and constraints of the environment.

4.2 A Problem with Activity Theory

Activity theory has the appropriate support for reasoning about human activity and the environment. It also has a useful hierarchy for activities. If activity theory can be applied, it could represent all activities from long-term goals to short-term and subconscious activities. Unfortunately activity theory is a descriptive and declarative tool. It is vague about what operations are and what actions are, with the boundary between actions and operations also being unclear.

In order to use activity theory, it needs to be modified to have specific definitions. We have to have a specified set of operations and actions for computation. However, dealing with the issue of consciousness is difficult, and makes activity theory difficult to apply in a computational field.

4.3 Modification of Activity Theory

To use activity theory, we must translate it into computational constructs. We found that if we put constraints on activity theory, we could recognize activities with simple object usage based on activity theory.

Constraint 1. *Operations on objects are limited to* 5 *habitual behaviors*: `MOVE`*,* `STAY`*,* `TOUCH`*,* `HOLD`*, and* `RELEASE`.

Constraint 2. *Actions consist of at least* 2 *or more operations.*

Through these constraints, activities can be expressed not by descriptive statements, but by a sequence of operations. We can obtain a finite set of operations for an action in a fixed time slot.

Fig.3 shows our modified activity hierarchy based on the hierarchy of activity theory. According to Constraint 1, the operation level is limited to one of 5 habitual and sub-conscious behaviors. All constituents in the operation level contain object or temporal information when actions are performed.

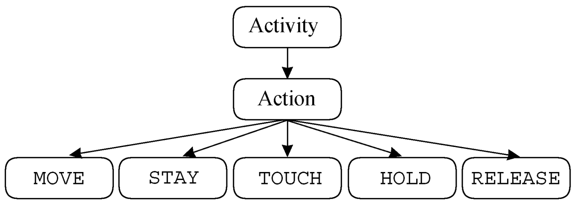

Fig.3. Activity hierarchy and added operations.

4.4 Representation

To reflect the philosophy of activity theory, we do not just use sensor data but modify the representation of the sensor data. According to activity theory, activities are distinguished from each other in terms of their objects. The motivation for an activity is connected with specific objects in a certain environment. A representation should include the relationships between tools, manipulating objects to change an environment, and the human using the tools.

We focus on these relationships to encode sensor data. In doing so, we adopt an FOL (First Order Logic)-style representation as follows:

$$\textit{Operation-Descriptor}(\textit{Object,Location,}[\textit{Time}]).$$

All defined operations in our activity theory follow this representation. In our representation, an operation has three arguments: object, location, and time. *Object* refers to tangible things that humans can interact with. *Location* represents the environmental information where an operation occurs. *Time* is the duration over which an operation was conducted. For example, `TOUCH(Dishwasher, Kitchen)` is used to encode a situation which a user manipulates a dishwasher in the kitchen. In this example, *Time* is omitted because it takes a short time to perform an operation; it is less important than the object, and would dramatically increase the required computation. Fig.4 shows an instance of translating raw data collected from RFID sensors into our representation.

Basically our representation is lightweight as well as informative. Because location and temporal information are discriminative representatives, we can express various situations regardless of the number of sensors.

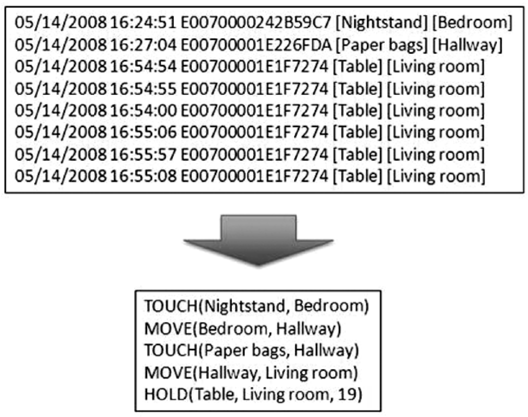

Fig.4. Conversion of raw data into proposed data representation. Raw data is represented in the upper box including date, time, RFID, object name, and location.

4.5 Learning Classifier

We use generative models to identify human activity. Generative models are based on the joint probability between evidence and classes. Assume that O is a set of features and A is a set of classes. We can represent a generative model as (1).

$$P(O, A) \propto P(A) \prod P(O|A). \tag{1}$$

We use a standard Bayesian estimate[17] method to calculate $p(x_i|y_i)$ as shown in (2), where n is the number of training examples for which $a = a_j$, n_c is the number of examples for which $a = a_j$ and $o = o_i$, p is prior estimate for $\hat{p}(o_i|a_j)$, and m is weight given to prior.

$$\hat{p}(o_i|a_j) = \frac{n_c + mp}{n + m}. \tag{2}$$

While training the generative classifier, the learning module builds extra information used in the recognition step, such as average time of duration for a particular action, average number of operations for each action, average time of day in which an action occurs, and deviation of average time.

4.6 Recognition

Assume that we have a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$ and a sequence of operations $O = \langle o_1, o_2, \ldots, o_m \rangle$. In the recognition step, we find the most probable action based on an observed operation vector. A typical way of assigning a class to the given evidence is to identify a class that gains the maximum margin of posterior probability as shown in (3)[18].

$$Action \leftarrow \mathtt{argmax}_{a_j \in A}\Big(p(a_j)\prod_{i}^{m}\hat{p}(o_i|a_j)\Big). \tag{3}$$

A strong independence assumption penalizes the performance of a classifier. By making an independence assumption, we ignore other factors in order to reduce the computational complexity, at a cost to overall accuracy or performance. In this paper, we use penalty functions in terms of time and the degree of matching between the learned model and a given instance so as to complement the independence assumption. (4) shows our penalized classifier by two factors.

$$Action \leftarrow \mathtt{argmax}_{a_j \in A}\Bigg(\frac{p(a_j)\prod_{i}^{m}\hat{p}(o_i|a_j)}{\delta(O, t|a_j)}\Bigg). \tag{4}$$

In traditional naive Bayes model, a role of denominator, $p(a_1, a_2, \ldots, a_m)$, is to normalize the formula to produce a result in the form of a probability. The equation describes the chance of events from a_1 to a_m occurring together. Typically the denominator is not used in computation because its value is the same for all classes. In contrast with the naive Bayes model, we want to reflect degrees of structural match between a trained model and a given sequence of observations by adopting the penalized function $\delta(O, t|a_j)$.

In the recognition step, two factors, the average time of duration for an action, and features (i.e., operations) consisting of a model for an action, are key elements affecting the performance of a classifier. Our data representation is tightly coupled with locations and objects. Our classifier is considered to be conditioned on locations and objects similar to CRFs. However, CRFs can lead to unexpected outcomes when two activities are interleaved, outcomes that are a regularly occurring phenomenon. In order to avoid yielding incorrect outcomes caused by unexpected situations, trained models are penalized according to the degree of mismatches of objects between a given example and given models (δ_0) and mismatchings of time (δ_t). (5) and (6) show penalty functions.

$$\delta_0 = \begin{cases} \dfrac{MM}{MA}, & \text{if } MM \leqslant MA \text{ and } MM \neq 0, \\ 1, & \text{if } MM = 0, \\ 0, & \text{if } MM > MA, \end{cases} \tag{5}$$

$$\delta_t = \frac{TD}{AD}. \tag{6}$$

Here, MM is the number of mismatched operations between an observation and each action model, MA is the number of operations in an action model, TD is the duration of observation, and AD is an average duration time of an action model.

Assume that we have a learned model consisting of three operations as shown in Fig.5 for the action `take a shower`. In recognition step, we are given a sequence of observations $O = \langle$ `MOVE(Bedroom, Bathroom)`, `TOUCH(Door, Bathroom)` $\rangle$ with a duration time, which represent the time those operations took. In this case, a mismatch between a learned model and evidences takes place, because there is a missing operation `TOUCH(Faucet, Bathtub)` in the observation sequence. Therefore the values of MM and MA are 1 and 3, respectively.

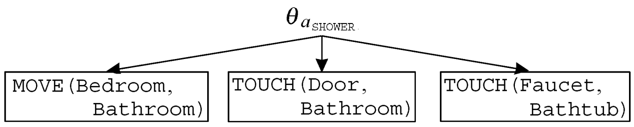

Fig.5. An example model of `take a shower` with its operations.

In order to control the degree of impact of each penalty, we control their relative weights. The final form of the penalty function $\delta(O, t|a_j)$ for an action is shown in (7).

$$\delta(O, t|a_j) = w_t\delta_t + w_0(1 - \delta_0), \quad w_t + w_0 = 1. \tag{7}$$

In our experiments, we assign equal weights for both so that $w_t = w_0 = 0.5$.

5 Experimental Results

We used Kastersen's dataset[10] in order to evaluate our method. The dataset (see Fig.4 in the previous section, for a sample) consists of 245 action instances for 7 different activities over 28 days, sensed using RFID technology. Table 1 contains a summary of the result of transforming the raw data into our activity representation based on the modified activity theory. We omitted 8 action instances because they did not contain any operations due to errors in the dataset.

In addition, we did not use the `RELEASE` operation with this dataset, because RFID technology cannot distinguish between `TOUCH` and `RELEASE` for objects. For this paper, we only used the `TOUCH` operation for identifying habitual behaviors.

Table 2 provides extra information such as duration, average number of operations for each action, the average one-hour time slot in which the action occurred (0∼23), and the deviation over the time slot. Some activities had a very high time slot deviation, meaning that they were not performed regularly and, because of this, it was hard to build a model for those activities based on temporal information. For this dataset, we found that actions did not consist of many operations. For example, `take a shower` consisted on average of only two operations, such as `TOUCH(Door, Bathroom)` and `TOUCH(Faucet, Bathtub)`.

Table 1. The Numbers of Training Examples and Test Examples for Each Action

Actions	No. Test Examples	No. Training Examples
`Leave House`	24	10
`Use Toilet`	100	13
`Take a Shower`	13	10
`Go to Bed`	10	8
`Prepare Breakfast`	11	9
`Prepare Dinner`	6	4
`Get Drink`	10	9

Table 2. Generated Statistics from Training Examples

Actions	Avg. Duration (s)	Avg. No. Oper.	Avg. Time Slot	Avg. Deviation (h)
`Leave House`	39 717	4	12	18.470
`Use Toilet`	105	3	12	53.280
`Take a Shower`	573	2	9	4.820
`Go to Bed`	28 902	4	10	3.660
`Prepare Breakfast`	202	6	9	0.400
`Prepare Dinner`	2 054	13	18	1.000
`Get Drink`	55	2	16	34.315

We used precision and recall to evaluate our method. We measured precision and recall to compare our method with other popular methods such as HMM and CRF.

In order to compare our approaches which were FOL-style representations of sensor data and penalized Naive classifiers with popular graphical models, we used HMM and CRF with a sequence of sensor data. In HMM and CRF, actions were hidden variables. An observable variable was a sequence of sensor data which was read by the system.

In the case of HMM, which is a generative graphical model as shown in (8), joint probability $p(X, Y)$ was factored into the initial state $p(y_1)$, transition distribution $p(y_n|y_{n-1})$, and observation with its parent label $p(x_n|y_n)$.

$$p(X, Y) = \prod_{n=1}^{N} p(y_n|y_{n-1})\ p(x_n|y_n). \tag{8}$$

Learning parameters of the three distributions corresponded to maximizing the joint probability $p(X, Y)$ of the paired observation and label sequence in the training data. In this paper, we used frequency counting to maximize joint distribution because we dealt with discrete data[19].

CRF, a discriminative probabilistic model, forms a series of non-negative potential functions as in (9), where Z is a normalization function.

$$p(Y|X) = \frac{1}{Z} \prod_{n=1}^{N} \Psi(n, y_{n-1}, y_n, X). \tag{9}$$

Potential function takes a specific form $\mathtt{exp}(w \cdot f(t, y_{n-1}, y_n, X))$, where w is a set of weights. These weights are the parameters we want to find when learning with the model. The feature function $f(\cdot)$ is comprised of two matching functions. Its first function describes the relation between states by a simple matching function that returns 1 when y_n and y_{n-1} are matched with particular values, and 0 otherwise. Its second function can be defined similarly, but it represents a relation between evidences and their corresponding states. It is also a simple matching function that returns 1 when y_n and X are matched with particular values, and 0 otherwise. In order to learn parameters, such as a set of weights, we used the iterative gradient method[20].

Table 3. Recognition Precision for Actions

Actions	HMM	CRF	NBC w/o Temporal Info.	NBC with Temporal Info.
`Leave House`	68.96	71.42	94.45	51.21
`Use Toilet`	95.23	95.29	94.73	92.59
`Take a Shower`	83.33	81.81	84.61	32.59
`Go to Bed`	30.90	39.13	47.05	50.00
`Prepare Breakfast`	62.50	50.00	75.00	78.57
`Prepare Dinner`	33.33	22.22	37.50	85.71
`Get Drink`	80.00	80.00	72.72	100.00
`Mean`	66.32	62.84	72.44	70.06

Tables 3 and 4 show the results of this model comparison. In order to compare with our method, we used raw data from the dataset to build models for each action. Then we trained a naive Bayes classifier based on modified activity theory with penalty functions. We performed two different experiments: one without temporal information, meaning that we did not consider time slot and deviation for actions, and one with temporal information. We expected that the temporal information would improve the accuracy of activity recognition.

Table 4. Recognition Recall for Actions

Actions	HMM	CRF	NBC w/o Temporal Info.	NBC with Temporal Info.
`Leave House`	83.00	83.33	87.50	87.50
`Use Toilet`	80.00	81.00	90.00	50.00
`Take a Shower`	76.92	69.72	84.61	84.61
`Go to Bed`	90.00	90.00	80.00	90.00
`Prepare Breakfast`	45.45	36.33	54.54	100.00
`Prepare Dinner`	50.00	33.33	50.00	100.00
`Get Drink`	80.00	80.00	80.00	60.00
`Mean`	72.19	67.67	75.23	81.73

Fig.6 compares the F-scores, weighted harmonic means of precision and recall, for each model. This shows that our method slightly improved recognition accuracy. By conditioning actions on locations and objects, our method caused training examples to be distinct from each other, allowing our classifier to clearly assign a class to evidence with marginal probability.

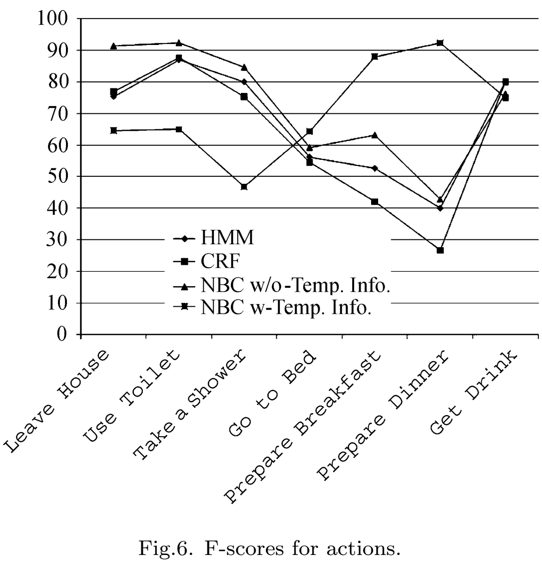

Fig.6. F-scores for actions.

The HMM and CRF models showed reasonable recognition performance except for the `go to bed`, `prepare breakfast`, and `prepare dinner` activities. Our method without temporal information slightly improved the accuracy over the HMM and CRF methods for recognizing those activities, but our method still has low accuracy. The reason why the `prepare breakfast` and `prepare dinner` activities have low accuracy is that those actions have almost the same objects and locations. In this case, the likelihood and posterior probability are almost equivalent between the two activities. In order to avoid this situation, the classifier needs more instances. However, just increasing the number of training examples does not guarantee the needed improvement. To make our method more accurate, we need more discriminative operations or actions in the classifier.

The goal of this paper is to reduce computations in inference for executable recognition models on mobile devices while maintaining reasonable recognition accuracy. Table 5 shows the number of parameters and complexity in inference for each model. While the number of parameters to be estimated is almost the same for these methods, our method reduces the computation complexity in inference, and maintains the recognition accuracy level of the HMM and CRF models. In our approach, N represents the number of actions such as `get drink` and T is the number of operations such as `TOUCH(Cup-A, Kitchen)` belonging to each action. We have 7 actions as shown in Table 1. The value of T depends on the learned models, but, on average, T is 5. This means that we can identify actions with only 5 operations regardless of the number of sensed data items. In HMMs and CRFs, N is the same as in our approach, but T is determined by the number of sensed data items. The value of T is 15 for our dataset. With these values of N and T, our approach uses 35 computational operations compared to 735 computational operations for HMMs and CRFs, an order of magnitude of improvement.

Table 5. Comparison of No. Parameters to Be Learned and Time Complexities for Each Model

Models	No. Parameters	Complexity
HMM (in chain)	$\approx N^2 + T \cdot N$	$O(N^2T)$
CRF (in chain)	$\approx N^2 + T \cdot N$	$O(N^2T)$
Our Model	$T \cdot N$	$O(NT)$

From these results, we find that temporal information should be carefully deployed. As shown in Fig.6, recognition accuracies of the `leave house`, `use toilet`, and `take a shower` activities are impaired when temporal information is considered. Those actions do not occur at regular time. The time slot deviations for these activities are higher than other activities except `get drink`, which has very distinctive objects with decreasing effect of temporal information. In contrast, the `prepare breakfast` and `prepare dinner` activities have dramatically improved recognition accuracy with temporal information. Those two activities occur at consistent time, making temporal information a strong feature for making a decision. Based on our results, we can conclude that temporal information is very useful for regular and routine activities. However, we need to be careful when using temporal information for irregular activities.

6 Conclusions

In this paper, we have proposed a lightweight classification method for activity recognition based on activity theory. Our experimental results showed that our method is promising for deployment in real applications. In terms of leveraging object usage for activity recognition, the approach described in [19] is conceptually similar to our approach, but it combines RFID and visual object recognition, resulting in a more computationally-complex system. Reducing computation while maintaining reasonable accuracy is important for developing real activity recognition system running on mobile devices. In a real environment, the number of activities to be recognized and the number of sensors will be much larger than those in the current laboratory experiments, stressing the need for low-complexity recognition approaches. High-computation recognition models are not feasible on mobile devices for such environments, due to battery and processing limitations. Our method requires far less computation, by *an order of magnitude*, than other methods, while maintaining and even providing a slight improvement in accuracy. Our modified activity theory and data presentation method, which changes raw sensor information into discriminative and meaningful information, make it possible to use less computation than other statistical methods for inference.

Our experiments revealed that some activities are more recognizable when we ignore temporal information. We will conduct further research into how to combine our models with and without temporal information and apply it in situations with a greater variety of activities. We would also like to apply our modified activity theory approach to predicting the next activity and a set of actions that will be observed. Building such a prediction model will allow us to better provide and support proactive services.

References

[1] Abe M, Morinishi Y, Maeda A, Aoki M, Inagaki J. A life log collector integrated with a remote-controller for enabling user centric services. *IEEE Trans. Consumer Electronics*, 2009, 55(2): 295-302.

[2] Baek S, Choi E, Huh J, Park K. Sensor information management mechanism for context-aware service in ubiquitous home. *IEEE Trans. Consumer Electronics*, 2007, 53(4): 1393-1400.

[3] Kim J, Jeong D, Baik D. Ontology-based semantic recommendation system in home network environment. *IEEE Trans.*

Consumer Electronics, 2009, 55(3): 1178-1184.

[4] Lee H, Lim S, Kim J. UMONS: Ubiquitous monitoring system in smart space. *IEEE Trans. Consumer Electronics*, 2009, 55(3): 1056-1064.

[5] Patterson D J, Fox D, Kautz H A. Inferring high-level behavior from low-level sensors. In *Proc. the 5th Int. Conf. Ubiquitous Computing*, Seattle, USA, Oct. 12-15, 2003, pp.73-89.

[6] Tapia E M, Intilli S S, Larson K. Activity recognition in the home using simple and ubiquitous sensors. In *Proc. Pervasive Computing*, Vienna, Austria, April 18-23, 2004, pp.158-175.

[7] Duong T V, Bui H H, Phung D Q, Venkatesh S. Activity recognition abnormality detection with the switching hidden semi-Markov model. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Diego, USA, Jun. 20-26, 2005, pp.838-845.

[8] Lao W, Han J, de With P H N. Automative video-based human motion analysis for consumer surveillance system. *IEEE Trans. Consumer Electronics*, 2009, 55(2): 591-598.

[9] Lester J, Choudhury T, Kern N, Borriello G, Hannaford B. A hybrid discriminative/generative approach for modeling human activities. In *Proc. Int. Joint Conf. Artificial Intelligence*, Edinberg, UK, Jul. 30-Aug. 5, 2005, pp.766-772.

[10] Kasteren T V, Noulas A, Englebienne G, Krose B. Accurate activity recognition in a home setting. In *Proc. the 10th Int. Conf. Ubiquitous Computing*, Tokyo, Japan, Sept. 11-14, 2008, pp.1-8.

[11] Patterson D J, Fox D, Kautz H A, Philipose M. Fine-grained activity recognition by aggregating abstract object usage. In *Proc. the 9th IEEE Int. Symp. Wearable Computer*, Osaka, Japan, Oct. 18-21, 2005, pp.44-51.

[12] Lio L, Fox D, Kautz H. Extracting places and activities from GPS traces using hierarchical conditional random fields. *Int. J. Robotics Research*, 2007, 26(1): 119-134.

[13] Landwehr N, Gutmann B, Thon I, Philipose M, Raedt L D. Relational transformation-based tagging for human activity recognition. In *Proc. the 6th Int. Workshop on Multi-Relational Data Mining*, Warsaw, Poland, Sept. 17, 2007, pp.81-92.

[14] Fogarty J, Au C, Hudson S E. Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition. In *Proc. the 19th ACM Symp. User Interface Software and Technology*, Montreux, Switzerland, Oct. 15-18, 2006, pp.91-100.

[15] Modayil J, Bai T, Kautz H. Improving the recognition of interleaved activities. In *Proc. the 10th Int. Conf. Ubiquitous Computing*, Seoul, Korea, Sept. 21-24, 2008, pp.40-43.

[16] Nardi B A. Context and Consciousness: Activity Theory and Human-Computer Interaction. MIT Press, 1996.

[17] Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 1989, 77(2): 257-286.

[18] Mitchell T. Machine Learning. McGraw Hill, 1997.

[19] Sutton C, McCallum A. Introduction to Statistical Relational Learning. MIT Press, 2006, pp.93-128.

[20] Wu J, Osuntogun A, Choudhury T, Philipose M, Rehg J M. A scalable approach to activity recognition based on object use. In *Proc. the 11th IEEE Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 14-20, 2007, pp.1-8.

Jaeyoung Yang received his B.S., M.S., and Ph.D. degrees in computer science and engineering from Hanyang University, Korea, in 1998, 2000, and 2003 respectively. He is a postdoctoral researcher in Human-Computer Interaction Institute, Carnegie Mellon University, USA. He is also a researcher of Web Intelligent Consortium Korea Center. His research interests include intelligent agents, machine learning, context-awareness, ubiquitous computing, and text mining.

Joonwhan Lee is the director of Service Develop Division of Neowiz Internet Inc., Seoul, Korea. He received his B.F.A. degree from Seoul National University, Korea, in 1995 and M.Des. degree from Carnegie Mellon University, USA in 2000. And then he received his Ph.D. degree in human-computer interaction from the School of Computer Science in the Carnegie Mellon University in 2008. His research interest focuses on human-computer interaction, interaction design, situationally appropriate user interaction, user interface of pervasive computing, adaptive user interface, and visualization.

Joongmin Choi is a professor in the Department of Computer Science and Engineering, Hanyang University, Ansan, Korea. He is the director of Intelligent Systems Research Lab. He is also the director of Web Intelligent Consortium (WIC) Korea Center. He received his B.S. and M.S. degrees in computer engineering from Seoul National University, Korea, in 1984 and 1986, respectively, and the Ph.D. degree in computer science from the State University of New York at Buffalo, USA in 1993. His research interest focuses on Web intelligence, which is a somewhat broad concept covering the areas of Web information extraction, Web data mining, semantic Web and ontologies, and other intelligent techniques for manipulating Web information. His other interest areas include intelligent agents, artificial intelligence, and context-aware personalization.