Dynamic Threshold Determination for Stable Behavior Detection

HIROYUKI YAMAHARA, FUMIKO HARADA, HIDEYUKI TAKADA, HIROMITSU SHIMAKAWA
Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, 525-8577 Shiga, JAPAN
yama@de.is.ritsumei.ac.jp, {harada, htakada, simakawa}@cs.ritsumei.ac.jp

Abstract: To provide services according to user behavior, parameters should be adapted appropriately for the precise recognition of user behavior. In particular, the threshold value which is used to create behavioral patterns matched for behavior recognition impacts accuracy of behavior recognition. Because the threshold value is common to all users in the conventional model, the threshold setting unsuitable for some users may cause low recognition rates. In this paper, we propose a behavior detection method which detects high-level user behaviors, such as "leaving home". The proposed method achieves stable behavior recognition regardless of users, by introducing a model which dynamically determines the threshold value for individual user.

Key–Words: Threshold, Context, Behavior, Ambient, Proactive

1 Introduction

We are developing a context-aware system which provides services in homes. Suppose a user is leaving his home carelessly with windows open. In order to prevent such a danger, our system informs the user that the windows are open before the user leaves his home. Such a service is valuable for the user because the service not only improves user amenity but brings relief and safety. In the above example, the timing to provide a service to the user is important. If the user is informed after the user leaves his home, the user must go back into the house for closing the windows. The user should be informed before he goes outside the house. As another example, let a home server have an attempted delivery notice when a user came home. Our system recommends the user to go to pick up a package before the user sits on a sofa to be relaxed. We refer to these services provided proactively according to user behavior as *proactive services*. In order to provide proactive services, our system must correctly detect characteristic behavior of the user in situations such as leaving home and coming home.

In this paper, we call a pattern which represents the characteristic behavior of the user as a *behavioral pattern*. Context-aware systems, including our developing system, often collect online sensor data acquired according to user behavior as behavior logs and recognize user behavior by matching the behavior logs with a behavioral pattern created in advance. To create the behavioral pattern, these systems need a specific amount of behavior logs as sample behavior logs. Sample behavior logs must be personal behavior logs, particularly in order to address characteristics of individual user behavior in homes as in this paper. First, a behavioral pattern is created with sample behavior logs on every situation to be detected. After the behavioral pattern is created, user behavior is detected by comparing behavior logs of current user behavior with the behavioral pattern of each situation.

Our system must determine threshold values, which are used for creating a behavioral pattern and for matching sensor data with the pattern. The first threshold is an *extraction threshold*. A behavioral pattern is created by extracting characteristics which frequently occur in sample behavior logs. The extraction threshold is a threshold of the occurrence frequency. If an improper value is set to the extraction threshold, behavior recognition accuracy is low because the characteristics of the user are not extracted adequately. The second threshold is a *detection threshold*. If the degree of conformity of sensor data to the pattern is more than the detection threshold, our system detects user behavior and provides services. Naturally, an improper detection threshold makes behavior recognition accuracy low. All context-aware systems require thresholds to be determined for creating a behavioral pattern and for matching the pattern. After many sample behavior logs are collected, initial values of the thresholds can be changed into more proper values by learning with the logs. However, it takes a long period to collect many sample behavior logs. That means the user might be kept away from services for a long period. Not to dissatisfy the user, proper threshold values should be determined with a small number of sample behavior logs which can be collected in a short duration. Therefore, in this paper, we discuss not the learning of values but how to determine a proper initial threshold value with a small number of sample behavior logs.

In the conventional model for determining a threshold value, a developer of a context-aware system or an expert of the system domain determines the initial threshold value before introducing the system to a user's actual environment. Having the system used by some test users on a trial basis, the expert analyzes relativity between change of recognition accuracy and changes in a threshold value. The threshold value is determined so that the recognition rate averaged for all test users becomes the highest. The determined value is used as an initial threshold value common to all users after introduction to actual user environment. However, it is difficult to achieve high recognition accuracy with the common threshold value for all users, because proper threshold values vary with individual behavioral pattern.

This paper aims to create a behavioral pattern which can stably bring out higher recognition accuracy by determining more proper threshold value than the conventional model, particularly for users whose behavior is not recognized well with the conventional model. Because it is difficult to determine the proper threshold value with only a small number of personal sample behavior logs, we also utilize data from test users as in the conventional model. However, unlike the conventional model, we cannot determine common threshold values directly and also cannot create a behavioral pattern with many data from test users in advance, because characteristics of behavior vary with individual user even in a same situation, as mentioned above. In this paper, we propose a method for determining an extraction threshold dynamically, based on a model which derives not a threshold value itself but a rule for determining the value by analyzing test user data. The knowledge acquired by analyzing test user data is not meaningful, if it is not about an attribute which has high commonality among many users. The conventional model determines the threshold value without separating attributes which have low commonality from ones of high commonality. As an attribute of high commonality, the proposed method focuses on the number of characteristics composing a behavioral pattern. We assume that there is a universally ideal number, which does not depend on individuals, of characteristics used for recognizing user behavior. The proposed method derives a determination rule of an extraction threshold by analyzing test user data with a focus on the number of characteristics composing a behavioral pattern. A value of the extraction threshold is dynamically determined based on the rule when creating an individual behavioral pattern after introducing a context-aware system to the actual user environment. The proposed method has the following advantages.

- Focusing on an attribute which has high commonality, the method acquires meaningful knowledge from test user data, from which the conventional model cannot acquire meaningful knowledge for detecting behavior of individuals.
- The method dynamically determines a threshold value for individual behavioral patterns created with a small number of sample behavior logs, using a threshold determination rule derived from test user data.
- With a proper threshold for individual behavioral pattern, the method improves the recognition accuracy for users whose recognition accuracy is low with the common threshold value.

The result of an experiment shows that the proposed method improves behavior recognition accuracy which is less than 80% with the conventional model. In some experimental subjects, it improves more than 10%.

The remaining part of this paper is composed as follows. Section 2 describes our behavior detection system. Section 3 explains a model for determining a threshold dynamically and we apply the model into our detection system. Section 4 shows evaluation result by an experiment. Section 5 presents related works. Finally, we conclude this paper in Section 6.

2 Behavior Detection in Homes

2.1 Detection of High-Level Behaviors

We consider situations of leaving home, coming home, getting up and going to bed, as situations in which proactive services can be provided effectively. For example, when getting up, our system provides a reminder service, which reminds a user of one-day-schedule and of things to be completed by the time the user leaves his home. If this reminder service is provided before the user starts preparing for leaving or for having a meal just after a series of activities when the user gets up, it enables the user to decide his next activity. When going to bed, our system provides services which bring relief and safety. For example, our system informs the user that the windows are not closed. We consider proactive services are valuable services which can prevent repentance and danger, which the user might face in the case that the services are not provided.

It is not preferable that proactive services are provided wrongly when "the user gets out of bed just for going to the toilet in the middle of sleep", or when

Figure 1: Objects embedded by RFID tags.

Figure 2: Ring-type RFID reader.

"the user goes outside his house just for picking up a newspaper". To detect user behavior correctly in situations in which proactive services should be provided, a characteristic sequence of some activities of the user in his home must be observed. Suppose the user goes to the toilet, picks up a wallet and a cell phone, wears a wristwatch and opens the front door in order. Such a sequence of activities can be a strong evidence which shows the user not just picks up a newspaper but leaves his home. In this paper, we call user behavior, such as leaving home, which is detected by observing a sequence of characteristic user activities as a *high-level behavior*. We assume that a high-level behavior is a long behavior of around ten minutes. Some existing studies propose methods for detecting user motion such as "walking" and "standing up" [1, 2]. Others try to detect simple activities such as "making tea" and "brushing teeth" [3, 4]. However, high-level behaviors, such as leaving home, cannot be correctly detected only by recognizing such simple activities because a characteristic sequence for detecting a high-level behavior is a complex sequence in which some activities are interleaved. Parts where there exist characteristic order relation between activities and parts where there do not exist characteristic order relation between activities are mixed in the sequence. In addition, sometimes rare activities, such as "picking up an umbrella" in a rainy day, can be inserted in the sequence. It is difficult to provide proactive services only by detecting simple activities. We are developing a system for detecting high-level behaviors by observing such a complex sequence of activities[5, 6].

2.2 Individual Habit in Touched Objects

To detect high-level behaviors, we must collect data which remarkably show characteristics of individual user behavior in each situation as behavior logs. We focus on the aspect that most people have habitual activities in a habitual order in situations such as leaving home and going to bed. Each user has his own characteristic behavior in such specific situations. That means the user habitually touches the same objects every time in the same situation.

We record histories of touched objects as behavior logs, using 13.56 MHz RFID tags. As shown in Fig. 1, the tags are embedded in various objects of a living space, such as a doorknob, a wallet, or a refrigerator. Every object can be identified by its unique tag-ID stored in the tag. Meanwhile, a user wears a finger-ring-type RFID reader shown in Fig. 2. With this RFID system, according to user behavior, the history of touched objects is recorded in a database as the behavior log of the user. Fig. 3 shows actual behavior logs recorded by our system. The table shows behavior logs of two users in situations of leaving home and coming home. For example, in the situation of leaving home, the habitual activities of user A are different from those of user B. From the logs, it is inferred that user A brushes his teeth, changes his clothes, picks up some portable commodities, and brings out a milk carton from the refrigerator. It is inferred that user B brushes his teeth, sets his hair, operates a VCR and then picks up some portable commodities. These behavior logs show that kind of touched objects and their order are different among individual users even in a same situation. Similarly, comparing each user's situation of leaving home to that of coming home, it is found that the user touches different kinds of objects or touches the same objects in a different order in different situations.

2.3 Behavior Detection with Ordered Pairs

We detect high-level behavior with a behavioral pattern represented by a set of *ordered pairs*, which show the order relation among touched objects.

The flow to create a behavioral pattern is shown in Fig. 4, with an example of a behavioral pattern in the situation of leaving home. The behavioral pattern is created offline after sample behavior logs are collected. Generally, existing methods based on probabilistic models, such as Hidden Markov Model (HMM), create a behavioral pattern with high recognition accuracy using both behavior logs of the situation of leaving home and logs of situations other than the situation of leaving home as sample behavior logs. However, under the constraint that a behavioral pattern must be created with a small number of sample behavior logs, even behavior logs of leaving home

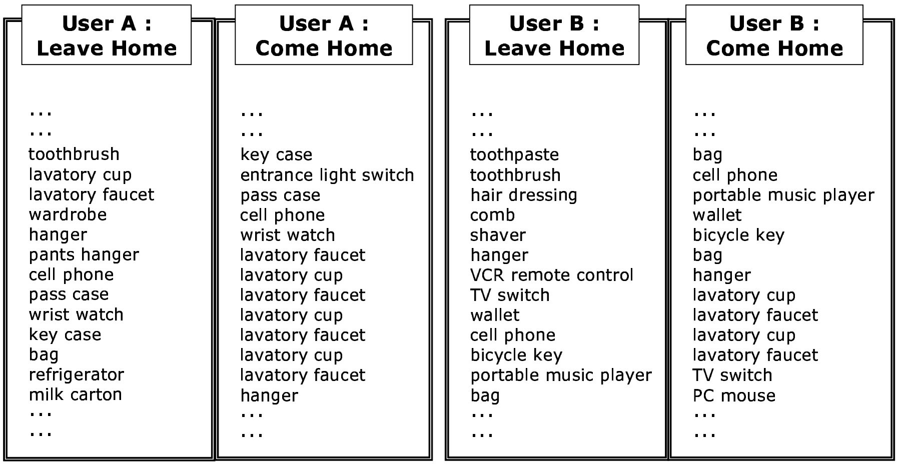

Figure 3: Examples of behavior log.

cannot be collected frequently. We cannot expect to collect behavior logs of other situations which are adequate to make recognition accuracy high. Therefore, we create a behavioral pattern only with behavior logs of leaving home.

First, w cases of behavior logs of leaving home are collected as sample behavior logs. The time length t_l of each sample behavior log is fixed. If m objects are sequentially touched in a behavior log l, then l is represented as a conjunction $\{o_1, o_2, \ldots, o_i, \ldots, o_m\}$, where, $o_{i-1} \neq o_i (1 < i \leq m)$. Second, all ordered pairs between two objects are enumerated from all collected sample behavior logs. If object o_j is touched after object o_i is touched, then the ordered pair p is represented as $\{o_i \rightarrow o_j\}$, which includes the case of $o_i = o_j$. For example, ordered pairs enumerated from a behavior log $\{o_1, o_2, o_3\}$ are $p_1 : \{o_1 \rightarrow o_2\}$, $p_2 : \{o_1 \rightarrow o_3\}$, $p_3 : \{o_2 \rightarrow o_3\}$. Next, the occurrence count of each ordered pair is counted up. It means not the number of times that each ordered pair occurred in a sample behavior log, but the number of sample behavior logs including each ordered pair in w logs. Finally, the ordered pairs where the ratio of the occurrence count to w is more than an extraction threshold e are extracted as a behavioral pattern π.

The behavioral pattern π is matched with the current behavior log of time length t_l, which is acquired online from current user behavior, every time the user touches objects. The number of ordered pairs which appear both in the behavior log and the behavioral pattern π is counted up. If the ratio of the number to the total number of ordered pairs composing the behavioral pattern π is more than the detection threshold d, user behavior of leaving home is detected.

The behavioral pattern of a set of ordered pairs can represent the user's habitual activities and their order. For example, an ordered pair, such as

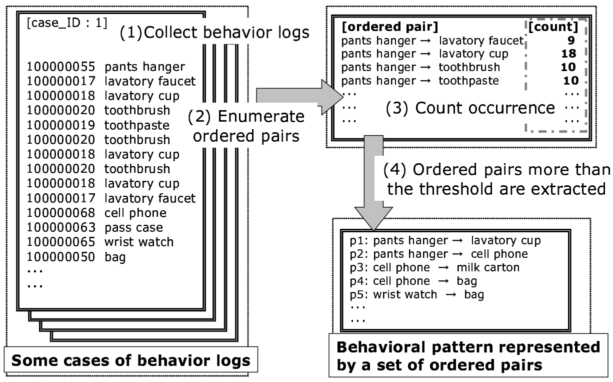

Figure 4: How to create a behavioral pattern.

$\{toothpaste \rightarrow toothbrush\}$, indicates the user's habitual activity that he "brushes his teeth" before he leaves his home. An ordered pair, such as $\{toothpaste \rightarrow pants\ hanger\}$, indicates habitual order of the user activities that "the user wears his pants after brushing his teeth" before he leaves his home. As shown also in our previous work [6], compared to the method using a Baysian Network [3, 4], a HMM [7, 8] and the method using time series association rule [9], this detection method has an advantage that the method can represent characteristics of complex user behavior by composing simple-structured behavioral pattern, which can be automatically created, with a set of the smallest unit of order.

2.4 Difficulty of Setting Threshold Values

We previously conducted an experiment in which we detected user behavior in situations of leaving home, coming home, getting up, and going to bed, using our detection method. We evaluated our method with the recognition accuracy both with $true\text{-}positive\ rate$ (TPR) and $true\text{-}negative\ rate$ (TNR). TPR shows

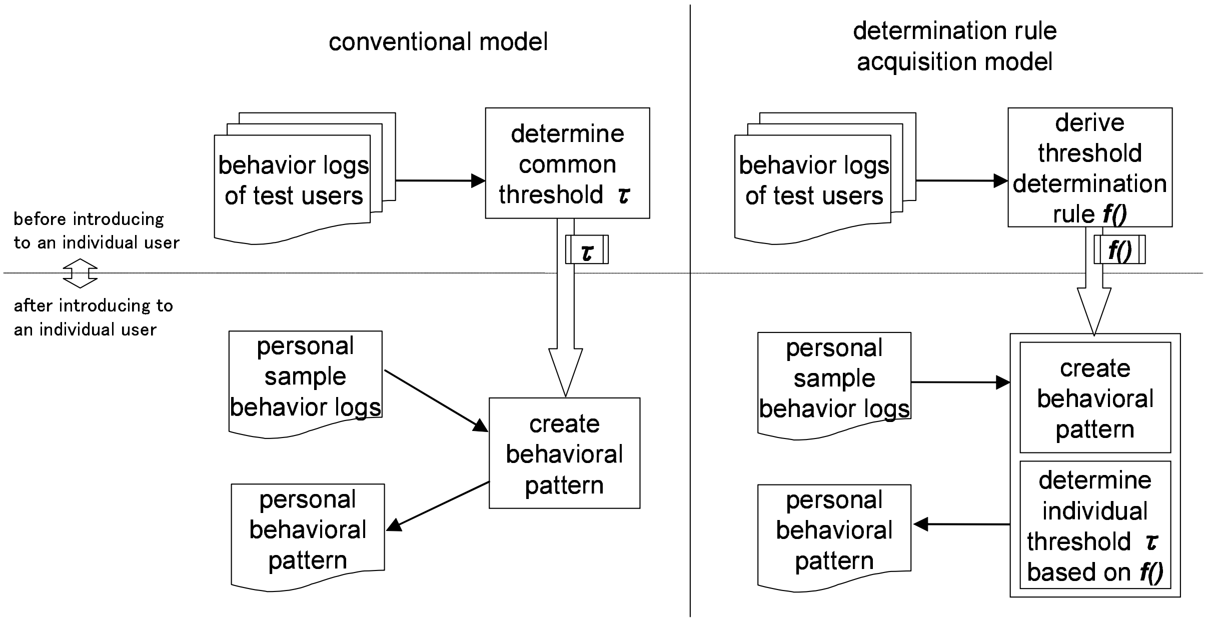

Figure 5: Determination rule acquisition model.

the rate at which behavior logs in a specific situation, which logs are referred to as $true\ cases$, are correctly detected with a behavioral pattern of the situation. TNR shows the rate at which behavior logs in situations other than the specific situation, which logs are referred to as $false\ cases$, are correctly ignored with the behavioral pattern of the situation. It is preferable that both TPR and TNR are high. As a result of the experiment, the recognition rates of some subjects have been more than 90%. Meanwhile, the recognition rates of a few users have been low rates of less than 80%. The rates vary among subjects.

The main cause of these differences is that the extraction threshold and the detection threshold are pre-determined values common to all users. Based on $half\ total\ true\ rate$ ($HTTR$), which is an average between TPR and TNR, these threshold values have been determined so that HTTR averaged for all users is maximum. It is necessary to improve the recognition accuracy of users, whose recognition rates are low with the common threshold values, by setting proper initial threshold values for individuals.

3 Dynamic Threshold Determination

3.1 Determination Rule Acquisition Model

We consider determining a threshold value dynamically for individual behavioral pattern. Unlike the conventional model which uses a fixed common threshold value, this paper proposes a model which acquires a rule to individually determine the threshold value for each behavioral pattern from the data of test users. The conventional model is illustrated on the left side of Fig. 5. The threshold determination rule acquisition model which we propose is illustrated on the right side of Fig. 5. The horizontal center line shows a partition of the two phases for introducing a context-aware system to actual user environment. The upper portion is the development phase, before introducing the system to the actual environments of individual users. The lower side is the operation phase, after introducing the system.

As shown in Fig. 5, the conventional model determines a common threshold value at the developement phase. First, the model collects behavior logs of test users. Next, for every test user, the model repeatedly creates a behavioral pattern with the logs, while matching the logs with the pattern. Analyzing the result of recognition accuracy on the matching, the model determines the threshold value with which recognition rate averaged for all test users is the highest. At the operation phase, the model creates an individual behavioral pattern with personal behavior logs. However, the threshold value is common irrespective of users.

To dynamically determine a proper threshold value for individuals, it is preferrable to acquire knowledge from personal behavior logs of individual user. However, it is difficult to determine the proper threshold value only with a small number of personal behavior logs. Therefore, the proposed model dynamically determines the threshold value by using both knowledge acquired by analysis of test user data and knowledge acquired from personal behavior logs. First, our model collects sample behavior logs of test users. Second, our model repeatedly creates a behavioral pattern with the logs and matches the logs with the pattern, for every test user. Next, our model analyzes the correlation between a threshold value and the recognition accuracy on results of the matching. If the threshold value is directly determined by the analysis, the same problem occurs as in the conventional model. Our model derives not a threshold value itself but a rule f for determining the value. The threshold value is not determined at the developement phase. At the operation phase, the threshold value is determined for individual behavioral pattern by applying the rule f to knowledge acquired from a small number of personal behavior logs.

3.2 Effect of Extraction Threshold

We apply the proposed model to our behavior detection system. The system has the extraction threshold and the detection threshold, which are described in Section 2.3. Primarily, it is important to set a proper value to the extraction threshold for extracting characteristics of user behavior in each situations. In this paper, we determine the value of extraction threshold dynamically with the proposed model.

The number of ordered pairs composing a behavioral pattern changes according to change of the extraction threshold, and affects the quality of the created behavioral patterns. It is preferable that a behavioral pattern includes many ordered pairs which are characteristics of user behavior in true cases. At the same time, the pattern should include few ordered pairs which can be characteristics of user behavior in false cases. If a behavioral pattern is composed of too few ordered pairs due to setting the extraction threshold high, then the behavioral pattern may not include some ordered pairs which should be normally included as user characteristics. The pattern will be conformed to by false cases unsuccessfully. On the other hand, if a behavioral pattern is composed of too many ordered pairs due to setting the extraction threshold low, then the behavioral pattern may include excessive ordered pairs which do not represent normal user characteristics. The pattern will not be conformed to by true cases successfully. In particular, such fluctuation is a sensitive problem under the constraint of a small number of sample behavior logs. If an improper value is set to the extraction threshold, it is impossible to extract ordered pairs adequately without excesses and shortages. Recognition accuracy is low because differences between true cases and false cases are small when matching those cases with the behavioral pattern composed by inappropriate ordered pairs. Since proper extraction threshold sharpens differences between true cases and false cases, recognition accuracy becomes high.

3.3 Rating as Determination Rule

We derive a threshold determination rule for setting the extraction threshold from data of test users. In the issue of high-level behavior detection, attributes such as kind of objects and their order have little commonality among users. It is difficult to derive a meaningful rule directly from these attributes. Therefore, we focus attention on the number of ordered pairs composing a behavioral pattern. As mentioned above, the number of ordered pairs affects the quality of behavioral patterns. The property of "the number of characteristics used for recognition", such as the number of ordered pairs, is similar to a property of human situation grasp in high impact. In psychological science, "the magical number seven, plus or minus two [10]" proposes the hypothesis which indicates that humans select about seven characteristic information items by screening a lot of information in order to instantaneously grasp the situation. This is a number common to all people. From another point of view, the person can estimate the situation properly by discarding excess information and selecting only minimum information. Consider the number of ordered pairs. In both of the case of excess ordered pairs and the case of insufficient ordered pairs, the recognition accuracy is low. This property of the number of ordered pairs is similar to the property of the number of items for human situation grasp. We assume that there is a universally ideal number of ordered pairs, which does not depend on individuals, as in the human situatin grasp. We attempt to derive a threshold determination rule for the extraction threshold by evaluating the threshold value with a focus on the number, which has high commonality, of ordered pairs.

With an example of a behavioral pattern of a user v in the situation of leaving home, we describe the proposed method which determines the threshold value dynamically. Before creating a behavioral pattern of user v, the threshold determination rule f is derived from behavior logs of x test users at the development phase. First, behavior logs in the situation of leaving home are collected as true cases, and also behavior logs in situations other than those are collected as false cases. Second, the following two steps are executed for every test user, repeatedly k times. Here, w is a given value common to all users.

1. Select w true cases as sample behavior logs and create w behavioral patterns with each setting of the extraction threshold value $e = 100 \times 1/w,\ 100 \times 2/w,\ ...,\ 100 \times w/w$, using the w true cases.

2. With all settings of the detection threshold d from 1% to 100%, match all true cases and all false cases with the w behavioral patterns.

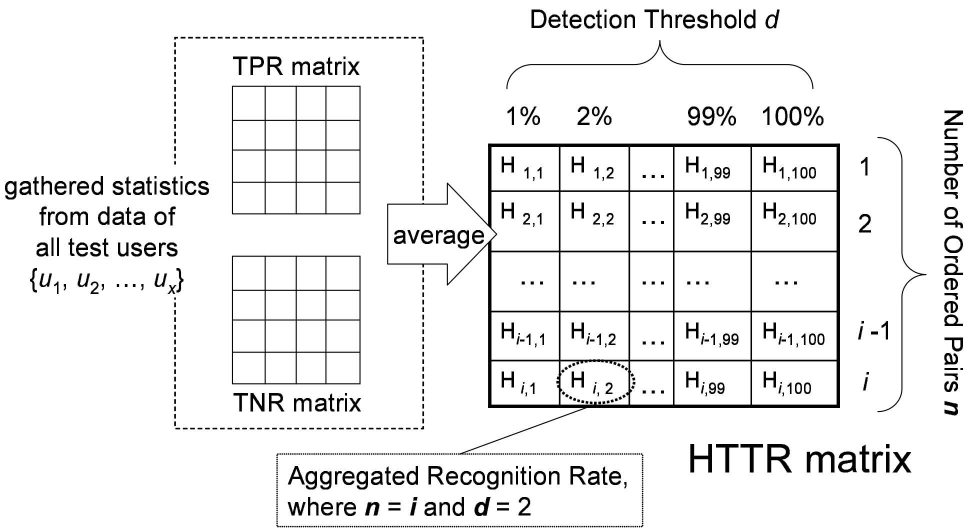

Figure 6: Matrix on statistics of test user data.

After these steps, TPR and TNR are calculated by gathering statistics on all results of the matching in above step 2. As shown in Fig. 6, TPR matrix and TNR matrix are formed for the statistics. The matrixes show the recognition rate with each number of ordered pairs and each setting of the detection threshold. When the maximum number of ordered pairs is i in all created behavioral patterns, each matrix forms $i \times 100$ matrix. We can get an HTTR matrix from these two matrixes. Each element H in the HTTR matrix is calculated by averaging each element in the TPR matrix and in the TNR matrix. In the process of statistics, the method records the number of statistical data leading to results of each row of the HTTR matrix because results of each row are respectively calculated with different numbers of statistical data.

Next, each row of the HTTR matrix is rated with a rating score. The rating score s_i of the ith row is calculated as follows.

$$s_i = \ln(p(i)) \times \max_j(H_{i,j})$$

$\max_j(H_{i,j})$ means the maximum value in 100 elements of the ith row. $p(i)$ is the proportion of the number of statistical data used for statistics of the ith row to the total number of statistical data. Because there are w settings of the extraction threshold per behavioral pattern, the total number of statistical data is $w \times k \times x$. $\ln(p(i))$ is a coefficient for adding the reliability of statistics to the rating score. This method gives a higher rating score to rows using more statistical data. Next, these rows are equally divided into c clusters, such as cluster 1:{row 1, row 2, row 3}, cluster 2:{row 4, row 5, row 6}, and so on. The rating score of a cluster is calculated by averaging rating scores of all rows in the cluster. The value of c is empirically determined. We assume that there is an ideal number of ordered pairs. However, because the number of ordered pairs composing a behavioral pattern depends on the number of ordered pairs occurring in sample behavior logs of individual user, one ideal number is not always identified using statistics of test user data. This method attempts to find, not one ideal number, but “how much number is roughly good”, by calculating rating scores of clusters. These rating scores correspond to the threshold determination rule. That is, when a behavioral pattern is created with personal sample behavior logs after introducing the behavior detection system to actual environment of user v, the extraction threshold is determined so that the behavioral pattern is composed of the number, which corresponds to as high rated cluster as possible, of ordered pairs.

Determining the threshold value based on the rule and creating a behavioral pattern with the value are dynamically executed after the context-aware system is introduced into an actual environment of individual user. However, because we assume that their processing is executed while users sleep in the night, their processing never negatively affects the speed of detection of user behavior when matching the current behavior log online with the behavioral pattern.

4 Evaluation

4.1 Experiment

This paper describes an experiment, in which we verify the efficacy of the proposed method comparing with the method using the conventional model. The experiment sets the time length t_l of a behavior log to 10 minutes. Before the experiment, we conducted a questionnaire survey for 2 weeks. In the questionnaire, subjects recorded the complete details about kind of objects the subjects touched and their order in 4 situations of leaving home, coming home, getting up, and going to bed every day. With the questionnaire results, we could confirm that many people respectively touch different objects or touch objects in different orders, in different situations. After that, we experimentally embedded the RFID system described in Section 2 into the living space. RFID tags are embedded in many household goods such as kitchen gas stove, kitchen sink, and electric appliances, in every spaces such as living, kitchen, entrance, and so on. In such spaces, we collected behavior logs of actual objects which subjects touched in the 4 situations respectively. The logs acquired online from subjects' behavior are stored in a database.

First, the threshold determination rule for the proposed method was derived by the calculations described in Section 3.3 with behavior logs of 8 subjects. In the experiment, rows in an HTTR matrix are divided into 100 clusters. Basically, each cluster includes three rows. But there are a few exceptions. Rows from the first row to the fifth row are included in a cluster which is rated as the second place from bottom, because they are empirically too small number as sample behavior logs. In addition, all of rows following the 300th row are included in the cluster same as the 300th row, whose cluster is rated as last place. Next, the following procedure was executed repeatedly 100 times per subject, in order to calculate individual behavior recognition accuracy with 8 subjects.

1. Select 5 sample behavior logs from true cases and create a behavioral pattern with the logs, based on the extraction threshold.
2. Select other 1 behavior log from true cases, and match the log with the behavioral pattern.
3. Match all behavior logs of false cases with the behavioral pattern.

Here, TPR is calculated based on cross validation. TNR is calculated by matching all false cases with all created behavioral patterns. In our previous work [5], we have conducted an experiment to find relativity between the number of sample behavior logs and recognition accuracy. As a result, recognition accuracy has not have the difference even if I heve changed the number of sample behavior logs for creating a behavioral pattern. Therefore, in the experiment, we limit the number of sample behavior logs used for creating a behavioral pattern to 5, which can be collected within a week. The extraction threshold is determined when creating a behavioral pattern in step 1 using the threshold determination rule described above. By gathering statistics of the result of all matchings, TPR, TNR and HTTR of every subject are calculated for the case in which the extraction threshold is dynamically determined.

After that, these rates in the case of using the conventional model are calculated by similar steps. In that case, the common extraction threshold is fixed to 80% in step 1 so that recognition accuracy is the highest. TPR, TNR and HTTR are calculated with all settings of the detection threshold from 1% to 100%. Two methods are compared using TPR and TNR on the detection threshold with which HTTR of each method is the highest per subject.

A user touches less objects or only limited kinds of objects in situations such as watching a TV and having a meal, which are situations other than the 4 situations to be detected in this experiment. Therefore our detection method can distinguish the 4 situations from other situations easily. Previously, we have conducted an experiment in which we recognize behavior logs including behavior logs of situations other than the 4 situations with behavioral patterns of the situation to be detected. Only up to 7% of ordered pairs, which compose individual behavioral pattern, have occurred in situations other than the 4 situations. This result shows that user behavior in situations other than the 4 situations has no chance to be mistakenly detected by our detection method. With this result in mind, we evaluate the recognition accuracy only with the 4 situations in the experiment of this paper. This means we evaluate our behavior detection method under more difficult conditions.

4.2 Discussion

Based on the result of the t-test, the experiment results are evaluated with the idea that difference of more than 5% is a statistically-significant difference between the proposed method and the method using the conventional model. As a result of the experiment, recognition rates in the proposed method are shown from Table 1 to Table 4. The tables respectively show the results of leaving home, coming home, getting up, and going to bed. Each table shows the TPR and the TNR by the proposed method. The value of $range$ shows the range of the detection threshold, which brings HTTR values whose difference from the highest HTTR value of each subject is less than 5%. The value of range is one measure of robustness to unsuitable setting of the detection threshold. Its value means a range of detection threshold value which achieves high recognition rate. The difference between the proposed method and the method using the conventional model is shown in parenthesis of each value. Positive values mean that the proposed method has in-

Table 1: Result of “Leaving Home”.

note	subj.	TPR %	TNR %*	range
	A	99	91.94	37 (+ 6)
	B	95	88.36	44 (+15)
#1	C	89 (+18)	92.84	45 (+ 7)
#2	D	94 (- 6)	98	49 (+ 7)
	E	99	99.68	46 (+18)
	F	100	95.04	32
	G	99	96.6	62 (+13)
#2	H	88 (-10)	91.14	39 (+15)

*is rounded off in the 3rd decimal place.

Table 2: Result of “Coming Home”.

note	subj.	TPR %	TNR %*	range
	A	91	95.25	33
	B	99	99.38	43 (+15)
#1,#2	C	90 (+14)	84.88 (-9.13)	36
#1	D	98 (+13)	98.8	28
	E	98	99.5	36 (+11)
	F	100	100	49 (+18)
	G	100	99.78	36
	H	100	100	33

*is rounded off in the 3rd decimal place.

Table 3: Result of “Getting Up”.

note	subj.	TPR %	TNR %*	range
	A	96	96.2	31 (+12)
#2	B	84 (-6)	82.48 (-14.3)	47 (+21)
#1	C	75 (+11)	96.23 (+12.52)	28 (-24)
#2	D	100	89.91 (-9.98)	33
#3	E	97 (+31)	59.38 (-27.13)	20 (-43)
#2	F	96	91.45 (-8.23)	40
	G	100	99.98	57 (+39)
#3	H	59 (-22)	93.6 (+30.22)	12

*is rounded off in the 3rd decimal place.

Table 4: Result of “Going to Bed”.

note	subj.	TPR %	TNR %*	range
	A	76	74.44	34 (-14)
	B	93	70.88	20
	C	95	99.98	29
#1	D	91 (+15)	95.94	40 (+11)
#1	E	47 (+12)	85.68	49 (-50)
	F	99	97.92	46 (+12)
	G	100	98.84	48
#1	H	97 (+15)	93.92	33 (+6)

*is rounded off in the 3rd decimal place.

creased the rates. The differences which are less than a statistically-significant difference are not shown in the tables.

About TPR and TNR in the tables, notable results are grouped into 3 groups from #1 to #3. In group #1, TPR or TNR have increased with the proposed method. Particularly, subject C of Table 1, subject C of Table 3, subject D and E of Table 4 have significantly increased. With the proposed method, their rates have increased more than 10% from low rates which are less than 80%. In group #2, TPR or TNR have decreased with the proposed method. However, even after decreasing, the rates can keep more than 80% for all subjects in group #2. Considering that our detection method must be introduced into a variety of user environments, the detection method must achieve high recognition accuracy stably for behaviors of as many users as possible. The detection method should not be effective on only a portion of users. In the experiment, the proposed method has decreased the rates of some subjects whose recognition rates are very high with the method using the conventional model. This decrease is not ideal result. However, the proposed method has increased significantly the rates of some subjects whose recognition rates are low with the method using the conventional model. This result shows the proposed method can achieve stabler behavior detection than the method using the conventional model. The proposed method is particularly important for initial threshold values, because it is preferable that recognition rates of all users are reasonably high rather than that recognition rates are very high only for some users and are low for others. Overall, the result of the experiment means the recognition accuracy can be improved by determining a better value of the extraction threshold with the proposed method. The result has proved the proposed method is effective. Exceptionally, the proposed method is not effective on subjects of group #3. In their TPR and TNR, one rate has increased and the other has decreased, based on just a trade-off relation.

Next, as for the value of range, there are lots of subjects whose ranges have been expanded by the proposed method in every situation. Even ranges of subjects whose TPR or TNR has not increased have expanded with our method. Ranges of subject C of Table 3 and subject E of Table 4 have been shortened. However, shortening of these ranges do not mean lowering of recognition accuracy because this shortening is an effect by increasing of TPR or TNR. These results show our method can be effective on improvement of the robustness to unsuitable setting of the detection threshold. Our method can create a behavioral pattern composed of more proper characteristics which can make differences between a behavior in a specific situation and behaviors in situations other than the situation without excesses and shortages by a better extraction threshold than the method using the conventional model. In other words, the method widens differences between the degree of conformity of true cases and the degree of conformity of false cases when matching the cases with the behavioral pattern. Therefore, the proposed method improves the robustness to unsuitable setting of the detection threshold.

5 Related Work

There are several approaches to determine proper threshold values in a variety of fields. In image processing, a determination method of a threshold used for extracting a specific area from a target image has been proposed [11]. This method can be used only if both parts to be extracted and parts not to be extracted exist together in a recognition target. Our issue does not meet such a condition, because behavior recognition in this paper considers whether a current behavior log conforms to a behavioral pattern or not. This approach in image processing cannot be applied to our issue. In other approaches, Support Vector Machines and boosting has been used for text categorization [12, 13], and HMM is used for speech and gesture recognition [14, 15]. These approaches can determine proper threshold values under the assumption that they can collect and analyze many samples of recognition target or many samples of others which have similar characteristics to samples of the recognition target instead. However, there is the constraint of a small number of sample behavior logs for creating a behavioral pattern in our issue. In addition, because characteristics of high-level behavior in homes are different among individual users, behavior logs of other people other than a user cannot be used for sample behavior logs of the user. Although these methods can be used for learning proper threshold values after many personal behavior logs have been collected, these methods cannot be used for determining proper initial threshold values.

In a field of behavior recognition, most existing works [3, 4, 9] do not discuss the setting of thresholds suitable for individual behavioral pattern.

6 Conclusion

In this paper, we proposed a model for determining threshold values dynamically according to individual behavioral pattern to achieve stable behavior detection for individual habit and individual environment. As a result of applying the proposed model into our behavior detection method, recognition rates of users whose recognition rates are less than 80% with the conventional model are improved more than 10%. The proposed model achieves stabler behavior detection than the conventional model.

In the future, we will evaluate our method by introducing it into more user environments. In addition, we will consider the following items as future works for achieving higher recognition rate. The first one is a method for learning threshold values. In this paper, we considered how to determine initial threshold values after introducing a context-aware system into an actual user environment. On the other hand, the learning method is necessary for update the initial threshold values to values more appropriate for individual user after lots of personal sample behavior logs are collected. The second one is the length of t_l of the current behavior log used for matching with a behavioral pattern. Characteristics of order relation between multiple activities are observed to detect high-level behaviors in our detection method. Therefore, in this paper, the length t_l is set to 10 minutes empirically by analyzing contents in reports and questionnaires on daily activities in situations where proactive services are provided, which has been preliminarily conducted on some experimental subjects. If t_l is set to a value more appropriate for individual user, it may be possible to achieve higher recognition rate. The third one is utilization of other contexts. For example, position of users can be a supplementary context effective for achievement of higher recognition rate. We will consider utility of other contexts and other sensors.

References:

[1] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proc. the 2004 conference on Graphics interface*, pages 185–194, 2004.

[2] Darnell J. Moore, Irfan A. Essa, and Monson H. HayesIII. Exploiting human actions and object context for recognition tasks. In *Proc. IEEE International Conference on Computer Vision 1999 (ICCV '99)*, pages 80–86, 1999.

[3] Donald J. Patterson, Dieter Fox, Henry A. Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proc. the 9th IEEE International Symposium on Wearable Computers (ISWC2005)*, pages 44–51, 2005.

[4] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *Proc. the 20th International Joint Conference on Artificial Intelligence (IJCAI2007)*, pages 2237–2242, 2007.

[5] Hiroyuki Yamahara, Hideyuki Takada, and Hiromitsu Shimakawa. An individual behavioral pattern to provide ubiquitous service in intelligent space. *WSEAS Transactions on Systems*, 6(3):562–569, 2007.

[6] Hiroyuki Yamahara, Hideyuki Takada, and Hiromitsu Shimakawa. Detection of user mode shift in home. In *Proc. the 4th International Symposium on Ubiquitous Computing Systems (UCS2007), Lecture Notes in Computer Science 4836*, pages 166–181, 2007.

[7] Shigeki Aoki, Yoshio Iwai, Masaki Onishi, Atsuhiro Kojima, and Kunio Fukunaga. Learning and recognizing behavioral patterns using position and posture of human body and its application to detection of irregular state. *The Journal of IEICE (D-II)*, J87-D-II(5):1083–1093, 2004.

[8] Cory D. Kidd, Robert J. Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth Mynatt, Thad E. Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proc. the 2nd International Workshop on Cooperative Buildings (CoBuild 1999), Lecture Notes in Computer Science, vol 1670*, pages 191–198, 1999.

[9] Taketoshi Mori, Aritoki Takada, Hiroshi Noguchi, Tatsuya Harada, and Tomomasa Sato. Behavior prediction based on daily-life record database in distributed sensing space. In *Proc. the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005)*, pages 1833–1839, 2005.

[10] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(3):81–97, 1956.

[11] Yumiko Kimura, Daishi Watabe, Hideyashu Sai, and Osamu Nakamura. New threshold setting method for the extraction of facial areas and the recognition of facial expressions. In *Proc. the IEEE Electrical and Computer Engineering, Canadian Conference (CCECE/CCGEI2006)*, pages 1984–1987, 2006.

[12] Lijuan Cai and Thomas Hofmann. Text categorization by boosting automatically extracted concepts. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR2003)*, pages 182–189, 2003.

[13] James G. Shanahan and Norbert Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. In *Proc. the 12th International Conference on Information and knowledge Management (CIKM2003)*, pages 247–254, 2003.

[14] Taichi Asami, Koji Iwano, and Sadaoki Furui. A stream-weight and threshold estimation method using adaboost for multi-stream speaker verification. In *Proc. the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2006)*, volume 5, pages 1081–1084, 2006.

[15] Feng Niu and Mohamed Abdel-Mottaleb. Hmm-based segmentation and recognition of human activities from video sequences. In *Proc. the 2005 IEEE International Conference on Multimedia and Expo (ICME2005)*, pages 804–807, 2005.