Proc. International Symposium on Wearable
Computers (ISWC 2011), June 2011.

Unsupervised Activity Recognition with User's Physical Characteristics Data

Takuya Maekawa, Shinji Watanabe
NTT Communication Science Laboratories
{maekawa.takuya,watanabe.shinji}@lab.ntt.co.jp

Abstract

This paper proposes an activity recognition method that models an end user's activities without using any labeled/unlabeled acceleration sensor data obtained from the user. Our method employs information about the end user's physical characteristics such as height and gender to find other users whose sensor data prepared in advance may be similar to those of the end user. Then, we model the end user's activities by using the labeled sensor data from the similar users. Therefore, our method does not require the end user to collect and label her training sensor data. We confirmed the effectiveness of our method by using 100 hours of sensor data obtained from 40 participants, and achieved a good recognition accuracy almost identical to that of a recognition method employing an end user's labeled training data.

1. Introduction

Activity recognition technology has various kinds of real-world applications such as care of the elderly, fitness support, and lifelogging. In particular, many studies recognize activities by using body-worn accelerometers to capture characteristic movements of parts of the body such as the hands or waist [1, 12]. This paper also focuses on acceleration-based activity recognition. However, because most studies employ a supervised machine learning approach and thus require an end user's labeled training data, this approach places a large burden on the user. Some approaches generate the user's specific activity models with small amounts of her and other users' labeled data prepared in advance. However, this approach still requires the end user to undertake data collection and labeling.

The new approach proposed in this paper can generate specific activity models for an end user without needing any labeled/unlabeled sensor data. Assume that we wish to generate activity models for an end user (*a target user*). We prepare labeled training data from many other users (*source users*) in advance. Our method finds and selects source users whose sensor data may be similar to those of the target user by using information about their physical characteristics (PC information) such as height, weight, dominant hand, and sport experience. Then, our method generates the target user's activity models from the labeled data of selected source users. For example, when we generate a 'walk' activity model of a target user, we find source users whose 'walk' activity sensor data may be similar to those of the target user by employing only their PC information. We then generate the 'walk' activity model of the target user obtained from the 'walk' sensor data obtained from the source users. Generally, a larger amount of training data means a better learned model. This is because, when we have large amounts of training data, we can capture and model many different human activities. However, we believe that when we construct an activity model of a target user, modeling her activity with source users' training data that are very different from those of the target user may reduce the accuracy with which her activities are recognized due to the mismatch between the two sets of data. For example, 'brush teeth' sensor data from a right-handed user may be very different from those of a left-handed user. Therefore, we construct the model solely from selected training data. In addition, when we recognize the target user's test data, we adapt the learned model to the target user by using the test data to achieve more accurate recognition. Our method simply asks a target user to input her PC information and so imposes only a small burden on the user.

The contributions of this paper are that we propose a new activity recognition method that models an end user's activities using information about her physical characteristics and adapts the models to the user by employing statistical methods, and we investigate the effectiveness of our method using large amounts of labeled sensor data, namely 100 hours of sensor data from 40 experimental participants.

2. Related Work

We introduce activity recognition studies that reduce the labeling effort required of an end user. Ohmura et al. [9] recognize activities based on small amounts of end user sensor data by using adaptation techniques such as maximum-likelihood linear regression (MLLR) and maximum a posteriori (MAP) adaptation techniques [7, 4], which are usually used in speech recognition. Forster et al. [3] implement an adaptive gesture recognition system that employs brain decoded signals to detect recognition errors automatically, and re-train the recognition model according to the detected errors. Stikic et al. [14] reduce labeling costs and achieves highly accurate recognition by using an active learning technique. That is, the recognition system locates a sensor data segment that it finds hard to recognize, and then asks the end user to input the correct answer for the segment. Huynh et al. [5] also reduce labeling effort by combining generative models and discriminative classifiers. More specifically, they first obtain clusters of activities by employing a generative approach without any labels and then boost the recognition results of the generative models by employing a discriminative approach with small numbers of labels. Krassnig et al. [6] model the activities of an end user by using labeled sensor data from other users of the same gender as the end user. There are several activity recognition studies employing *non*-wearable sensors that also attempt to reduce labeling work. Kasteren et al. [16] employ ubiquitous sensors installed in a house, and recognize activities in the house by using labeled training data from other houses with a transfer learning technique. Perkowitz et al. [10] employ RFID tags attached to daily objects and automatically generate activity models that involve object usage information from web pages such as cooking recipes. The method we propose here employs information about the physical characteristics of the users to reduce their effort in addition to employing adaptation techniques.

3. Our Approach

3.1. Overview

Our recognition method consists of three procedures; preparation, activity modeling, and activity recognition. As preparation, we first compute the similarities between the activities of source users by using labeled acceleration data collected from the source users in advance. Second, for each activity class, we learn the relationship between the activity similarities and the attributes of the users' PC information. That is, we learn a model that estimates the activity similarity of a certain activity class for two users by using their PC information. We call the model a *user similarity model*. For example, a user similarity model of the 'walk' activity class estimates 'walk' activity similarity between two users by using their gender, height, weight, etc. The PC information includes physical information about the source users such as their height and age. The PC information also includes information related to the source users' activities

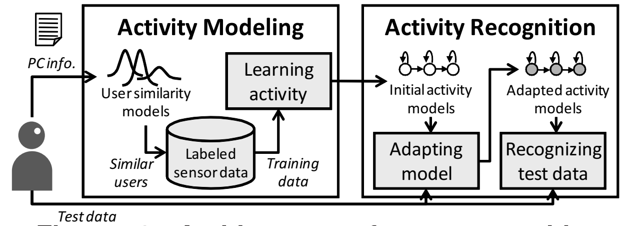

Figure 1. Architecture of our recognition method.

that we want to recognize such as their dominant hand in a 'tennis' activity and the frequency of a 'wash dishes' activity in their daily lives.

Fig. 1 shows the architecture of our activity modeling and recognition procedures. In the activity modeling procedure, for each activity class, a user similarity model of the class estimates source users whose sensor data may be similar to those of the target user by using PC information about the target and source users. Then, we generate a model of the activity class by using labeled data from the source users. We call a model generated from similar source users' labeled data an *initial activity model*. In the activity recognition procedure, we recognize the activities of a target user's test data with the initial activity models. Here, we generate adapted models for her with the test data to achieve more accurate recognition.

3.2. Preparation

We want to construct a model that estimates activity similarity between two users by using PC information about the two users for each activity class. Therefore, we employ PC information to compute the attributes needed to estimate the activity similarities. As shown in Fig. 2, we prepare a pair consisting of an attribute set and an answer (activity similarity of the activity class) for every combination of two source users. With the first pair in Fig. 2, we regard source users A and B as base and object users, respectively, and compute attributes from their PC information and the activity similarity from their labeled data. We prepare the pairs for each activity class, and learn a user similarity model of the class by using the pairs. The procedures are described in detail below.

Computing attributes. We compute attributes from PC information about the two users. We assume two types of PC information; numerical and nominal information. Numerical information includes a user's height, weight, and age. We generate four attributes from numerical information about two users. Using the age information obtained from base user A and object user B, for example, we generate a numerical value for the base user's age, a numerical value for the object user's age, the difference between the base user's age and the object user's age, and the ratio of

users (base - obj)	age					gender			activity similarity
	base user	object user	difference	ratio		base user	object user	difference	
A - B	29	32	3	1.1	…	male	male	0	0.3
A - C	29	50	21	1.7	…	male	male	0	0.1
		⋮					⋮		
Z - Y	36	36	0	1.0		male	female	1	0.01
	attributes								answer

Figure 2. Attributes and activity similarities of a certain activity class.

the object user's age to the base user's age as shown in Fig. 2. We compute the attributes for each piece of numerical information for users A and B.

Nominal information includes a user's gender, dominant hand, and sport experience. For example, gender information has 'male' and 'female' values, and sport experience information has 'yes,' 'somewhat,' and 'no' values. We generate three attributes from the nominal information about two users. From the gender information about base user A and object user B, for example, we generate a nominal value for the base user's gender, a nominal value for the object user's gender, and the difference between the base user's gender and the object user's gender. We define the difference between the two nominal values as 0 when they are identical, and 1 when they are different. Note that there is nominal information that has ordered values, and we allocate a numerical value to each nominal value. We normalize the numerical values so that the minimum and maximum values become 0 and 1, respectively. With sport experience, for example, we define 'yes' as 1, 'somewhat' as 0.5, and 'no' as 0. We regard the difference between the numerical values allocated to two nominal values as the difference between the two nominal values. We compute the attributes for each piece of nominal information for users A and B.

Computing activity similarity. We extract features from the source users' sensor data and employ them to compute activity similarities between the source users. We extract features based on existing activity recognition studies. Because we assume time-series acceleration data, we compute a feature vector for each sliding time window. We extract features based on the FFT components of 64 sample time windows. As features, we use the mean, energy, and dominant frequency, as described below. The mean is the DC component of the FFT coefficients, and can characterize the posture of parts of the body. For example, a mean corresponding to a hand posture during tooth brushing may have particular characteristics. The energy feature is calculated by summing the magnitudes of the squared discrete FFT components. Note that the DC component of the FFT coefficients is excluded from this summation. The energy can be used to distinguish low intensity activities such as standing from high intensity activities such as walking [17, 1]. The dominant frequency is the frequency that has the largest FFT component, and it allows us to distinguish between repetitive motions with similar energy values [8]. We construct a feature vector concatenating the above features extracted from all the body-worn accelerometers. When each user wears four three-axis accelerometers, we can construct a 36-dimension feature vector ($3 \times 3 \times 4 = 36$).

We compute the similarities between source users' activities simply by using a Gaussian mixture model (GMM). Assume that we wish to compute activity similarity between the 'walk' activities of source users A and B. We regard user A as a base user and user B as an object user, and compute the similarity between the object user's 'walk' activity sensor data and the base user's 'walk' activity model. We first model the 'walk' activity of user A with the GMM by using feature vectors extracted from her 'walk' sensor data. We employ the EM algorithm to estimate the GMM parameters [2]. There are 32 mixtures in our implementation. Then, we compute the GMM likelihood of each feature vector extracted from the 'walk' sensor data of object user B. We simply assume that the average likelihood over feature vectors corresponds to the similarity between the 'walk' activity of base user A and that of object user B. When we regard user B as a base user and user A as an object user, we can also compute the similarity between the 'walk' activity of base user B and that of object user A. We apply the procedure to all combinations of two source users and all activity classes.

Learning relationship between activity similarities and attributes of PC information. As described above, we can compute a pair of attributes and the activity similarity for two users according to an activity class. We prepare pairs for every combination of two source users and employ them to learn a user similarity model of the activity class by using the sequential minimal optimization (SMO) algorithm for regression [13] implemented in the Weka [18] toolkit. We generate the model for each activity class. If we have PC information about two users, we can use the model to estimate activity similarity for the two users.

3.3. Activity modeling

We generate a model for recognizing an activity class of a target user.

Finding similar source users. For each activity class, we estimate the activity similarity between the target user and each source user by using a user similarity model of the activity class. That is, we compute attributes for estimating activity similarity from PC information about the target and each source user, and then estimate the similarity of the activity class. After that, we generate a ranking of source users by their estimated similarities.

Learning activity. By using labeled data of the activity class from the top n similar source users, we generate an activity model of the target user. We investigate an appropriate n value in the evaluation section. We learn an activity class with a left-to-right HMM (implemented in [15]) where its values of observed variables correspond to extracted feature vectors, and we represent its output distributions by using Gaussian mixture densities. See section 3.2 for a description of feature vector extraction. We employ the Baum-Welch algorithm [11] to estimate the HMM parameters. We call an activity model generated from similar source users' labeled data an *initial activity model*. We learn an activity model for each activity class that we want to recognize. In our implementation, we use six states HMMs with 64 Gaussians.

3.4. Activity recognition

We recognize the unlabeled test data from a target user by using her activity models. That is, we extract a feature vector from the test data at each time slice (see section 3.2.) and then classify the feature vector into the corresponding activity class. Here, we adapt her initial activity models to the target user by using the test data. We then recognize the test data by using the adapted models. Here, we employ MLLR adaptation to compute a linear transformation of the mean parameters of Gaussian mixtures in the HMMs. That is, we shift the output distributions of the initial activity models (HMMs) by using the test data so that each state in the HMMs is more likely to generate test data. A new estimation of the adapted mean $\hat{\boldsymbol{\mu}}$ is given by

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \\ &= \boldsymbol{W}\boldsymbol{\xi}, \end{aligned}$$

where $\boldsymbol{\mu}$ is the initial mean, $\boldsymbol{A}$ is a $k \times k$ transformation matrix, where k is the number of dimensions of the feature vector, $\boldsymbol{b}$ is a bias vector, $\boldsymbol{W}$ is a $k \times (k+1)$ transformation matrix that is decomposed into $\boldsymbol{W} = [\boldsymbol{b}\ \boldsymbol{A}]$, and $\boldsymbol{\xi}$ is the extended mean vector $\boldsymbol{\xi} = [1\ \mu_1\ \mu_2 \cdots \mu_k]^T$. Therefore, we estimate the $\boldsymbol{W}$ that reduces the mismatch between the initial models and the test data by using the EM technique.

Here, we can adapt the initial models more exactly by transforming Gaussians for each HMM or each group of Gaussians. This is because the Gaussian distributions of activity classes are considered to be different for each user. Since the test data of the target user are unlabeled, we first recognize the test data with the initial models and then adapt the models according to the recognition results to achieve a more exact adaptation. After the initial recognition procedure, we can determine which feature vector is assigned to

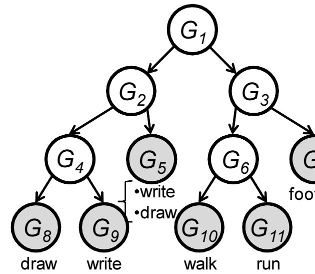

Figure 3. Example binary regression tree.

an output of which HMM state in an initial model. Therefore, for each HMM state in an initial model, we can estimate transformations of Gaussians in a state that reduces the mismatches between the Gaussian distribution and the distribution of its corresponding feature vectors. However, the initial recognition results of the test data may include several errors. In addition, when we have insufficient test data for an HMM state, we overestimate the transformations of the Gaussians in the activity model, which degrades the recognition accuracy.

To achieve more flexible adaptation, we use a regression tree technique [7] developed in relation to speech recognition, which changes the number of transformation matrices according to the amount of test data automatically. A binary regression class tree is created to cluster Gaussians into two groups in each internal node. The leaf nodes of the tree specify the final groups of the Gaussians. Fig. 3 shows an example of a binary regression tree. The root node G_1 corresponds to all the Gaussians in the initial activity models and the node splits the Gaussians into two groups; G_2 and G_3 by using a centroid splitting algorithm with a Euclidean distance measurement. We iterate the splitting until the requested number of leaf nodes are created. That is, similar Gaussians are grouped together hierarchically in the tree. In the example in Fig. 3, each leaf node is dominated by Gaussians in an initial activity model(s) of an activity class(es) that is specified below the node. For instance, G_{10} has many Gaussians in the 'walk' activity model. That is, G_{10} is a group of Gaussians consisting of many Gaussians in the 'walk' HMM. Also, Gaussians of similar activity classes are involved in the same node. For instance, G_3 has Gaussians of the 'walk,' 'run,' and 'football' activity classes. As above, by tying together similar Gaussians that belong to different HMM states, we can robustly estimate a transformation of the Gaussians even if there are recognition errors since the tying class is insensitive to the recognized HMM states. For example, even if the initial models mistakenly recognize several feature vectors of a 'walk' activity as a 'run' activity, we can mitigate the effect of the error when we focus on an ancestor node, e.g., G_6, and transform the node that ties the Gaussians in G_{10} and G_{11} together.

By using the regression class tree, we can also achieve a flexible transformation according to the amount of test data.

Table 1. Activities performed in our experiment and their average duration (minutes).

A	stand 0.19	F	descend stairs 0.09	K	draw on whiteboard 0.84
B	walk 0.44	G	bicycle 1.04	L	write in notebook 0.58
C	run 0.41	H	brush teeth 1.18	M	play pingpong 1.00
D	sit 0.65	I	wash dishes 1.66	N	vacuum 0.83
E	ascend stairs 0.10	J	use PC 0.55		

Table 2. Physical characteristics information used in our experiment.

name	value	name	value
gender	{male, female}	age	numerical
height	numerical	weight	numerical
dominant hand (writing)	{right, left}	dominant hand (pingpong)	{right, left}
dominant hand (brushing)	{right, both, left}	dominant hand (vacuuming)	{right, both, left}
pingpong experience	{yes, somewhat, no}	calligraphy experience	{yes, no}
touch typing capability	{yes, somewhat, no}	frequency of dish washing	{usually, sometimes, rarely, never}
frequency of bicycling	{usually, sometimes, rarely}	vacuum cleaner type	{canister, hand-held}
bicycle type	{upright, folding}		

When we have very few test data, we apply only a global transformation to Gaussians in the root node. Otherwise, we can apply more specific transformations to deeper nodes. That is, we can change the nodes to which a transformation is applied according to the amount of test data. Assume that G_4 involves similar Gaussians commonly found in 'write' and 'draw' activities such as those corresponding to 'erase lines and characters,' i.e., G_8 and G_9 involve Gaussians corresponding to 'erase lines' in a 'draw' activity and 'erase characters' in a 'write' activity, respectively. Also, assume that we have insufficient 'erase characters' test data. In this case, we focus on G_4 and transform the Gaussians in G_8 and G_9 together. This transformation permits us to adapt Gaussians corresponding to 'erase characters' to the target user with few 'erase characters' test data. As above, because several activity models have similar Gaussians, we consider that incorporating regression class trees for activity recognition may work well. To our knowledge, no work has reported incorporating regression class trees for activity recognition. For more details on the regression tree, see [7].

4. Evaluation

We evaluate our method with 100 hours of sensor data obtained from 40 paid experimental participants.

4.1. Data set

We collected sensor data with our developed wireless sensor nodes equipped with three-axis acceleration sensors and sampling rates of 30Hz. Each participant wore the sensor nodes on the wrists of both hands, waist, and right thigh. Here, the most natural data would be acquired from the normal daily lives of the participants. However, obtaining sufficient samples of such data from large numbers participants is very costly. Therefore, we collected sensor data by using a semi-naturalistic collection protocol [1] that permits greater variability in participant behavior than laboratory data. In the protocol, the participants perform a random sequence of activities following instructions on a worksheet. The participants are relatively free as regards how they perform each activity because the instructions on the worksheet are not very strict, e.g., "brush your teeth at the sink" and "vacuum the room with a hand-held vacuum cleaner." During the experimental period, the participants completed data collection sessions that included the random sequence of activities listed in Table 1. These activities were basically selected from those reported in acceleration-based activity recognition studies. Each participant completed ten sessions in total in our experimental environment. To annotate the collected data, a companion recorded the participants with a web camera during the experiment. The web camera was connected to a mobile computer carried by the companion. The sensor data from the four sensor nodes attached to the participant were also sent to the mobile computer. We describe how several of the activities in Table 1 were performed in detail. In activity J, we instructed the participants to enter several sentences on the computer keyboard. In activities K and L, we instructed the participants to write some sentences in a notebook and on a whiteboard, respectively. In activity M, each participant played pingpong with a worker in our laboratory. We collected a total of 9871 labeled activities from the 40 participants.

Each participant also filled out a questionnaire that asked for the PC information listed in Table 2. We selected various kinds of PC information about the participants ranging from basic physical characteristics information such as weight and gender to information related to the activities listed in Table 1. The activity-related information includes the dominant hand used in several activities, and the frequencies of several activities in their daily lives. The PC information also includes the types of objects used in the experiment, i.e., types of bicycle and types of vacuum cleaner. That is, we specified which type of object each participant should use in the experiment. As regards bicycles, each participant used a specified type of bicycle (upright or folding bicycle) in all ten of her sessions. This simulated a situation where a participant might use her own bicycle at anytime during her daily life. Fig. 4 shows the distribution of the PC

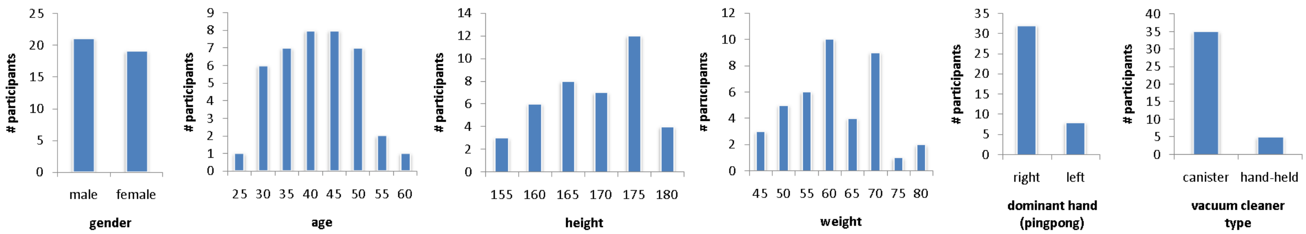

Figure 4. Distribution of selected physical characteristics of our participants.

information about our paid subjects.

4.2. Evaluation methodology

We evaluated our recognition method with 'leave-one-participant-out' cross validation. That is, we regarded one participant as a target user and remaining participants as source users, and we computed the activity recognition performance of the target user's sensor data (test data). We iterated the procedure so that each participant became a target user once. We evaluated the recognition (classification) performance of our method by using its error rate. The error rate is described as $error\ rate = 1.0 - F\text{-}measure = 1.0 - \frac{2 \cdot precision \cdot recall}{precision + recall}$. Precision and recall were calculated based on the results for the estimated class at each time slice. The smaller error rate gave a better classification performance. To validate the effectiveness of our proposed method, we tested the following seven methods.

- User-dependent modeling (DPN): We model a target user's activities with her labeled data. This method requires labeling work by the target user.

- Random selection (RND): We model a target user's activities with randomly selected source users' labeled data. When n is 39, we use labeled data of all the source users because the number of the participants is 40. When we select a large n, we can incorporate various activity patterns into the target user's activity models and we can train the models with sufficient amounts of data. This method is a naive method.

- Random selection and MLLR adaptation (RND+MLLR): After modeling a target user's activities with randomly selected source users' labeled data, we adapt the models using a simple MLLR adaptation that performs a global transformation of Gaussians.

- Random selection and MLLR adaptation with regression tree (RND+Tree): After modeling a target user's activities with randomly selected source users' labeled data, we adapt the models using the regression tree.

- PC information-based selection (PC): This method models a target user's activities with source users' labeled data selected by using the user similarity models described in section 3.2. Note that this method does not perform any adaptation.

- PC information-based selection and MLLR adaptation (PC+MLLR): After modeling a target user's activities with source users' labeled data selected by using the user similarity models, we adapt the models using a simple MLLR adaptation that performs a global transformation of Gaussians.

- PC information-based selection and MLLR adaptation with regression tree (PC+Tree): This is our proposed method.

4.3. Performance when estimating similar source users

Before examining the activity recognition performance, we briefly evaluate the estimation of similar source users. Our method can estimate activity similarity between each source user and a target user by using their PC information, and then provide a ranking of source users in terms of their similarities for each activity class. We compare the estimated ranking with a correct answer (correct ranking) that is created by actually computing the activity similarity between the target user and each source user by using their labeled sensor data. We evaluated the accuracy of the estimated ranking by using the accuracy rate of the estimated top-n similar source users. The accuracy rate is the ratio of the number of source users correctly estimated as top n similar users to n. Fig. 5 shows the transitions of the average accuracy rate and a random guess ratio when we change n. When n is 2, for example, the random guess ratio is $2/39 = 0.0513$. For any n, our method outperformed the random guess method. For example, when we want to obtain the top-10 similar source users, i.e., n is 10, our method can find similar users with an average accuracy of about 50%. The advantage of our method is that it does not require any sensor data from the target user but only PC information about the target user.

4.4. Performance for activity recognition

We first show the performance of the DPN method that generates user dependent models with labeled data from a

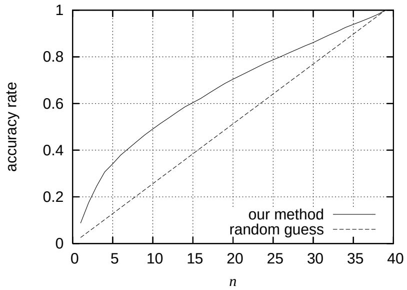

Figure 5. Average accuracy rate of similar user estimation.

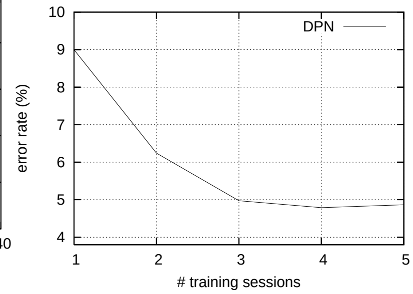

Figure 6. Transition of error rates of DPN method.

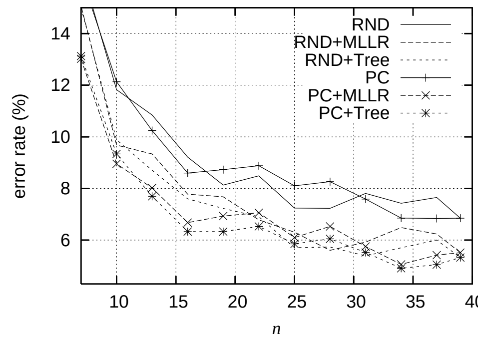

Figure 7. Transitions of error rates of recognition methods.

target user. Fig. 6 shows the transition of its error rates (percentages) when we change the number of training sessions. When there are two training sessions, for example, the DPN method learns the activity models of the target user by using the first two of her total of ten sessions and tests the models with the remaining eight sessions. The lowest error rate was 4.79% when the number of training sessions was four. We aim to achieve this error rate without a target user's labeled data.

Fig. 7 shows the error rate transitions of the remaining six methods when we change n. Note that n corresponds to the number of selected source users. We first focus on RND and PC methods that do not use any adaptation technique. Basically, their error rates decreased as n increased. This is because, when we have sufficient amounts of training data, they permit us to capture varieties of activities. When n was 39, the RND and PC error rates were identical because these methods employed training data from all source users. The RND and PC error rates when n was 39 were 6.85%, and this was the lowest error rate for RND. With the PC method, we want to better this 6.85% error rate. However, the lowest error rate of PC when n was smaller than 39 was 6.84%, namely it was almost identical to 6.85%. Note that PC method achieves a good error rate even when PC method does not use training data from all source users.

Then, we focus on RND+MLLR and PC+MLLR. For most n values, PC+MLLR outperformed RND+MLLR. The lowest error rate with RND+MLLR was 5.52% when the method used training data from all source users, i.e., n was 39. On the other hand, the lowest error rate with PC+MLLR was 5.06% when n was 34, which was superior to that of RND+MLLR by 0.46%. Even though the lowest error rate of PC method was almost identical to that of RND, the lowest error rate of PC+MLLR was smaller than that of RND+MLLR. In addition, the MLLR adaptation reduced the error rates of the random and PC information-based methods by an average of 1.54% and 2.35%, respectively. That is, the MLLR adaptation effectively reduced the error rates when we selected training data from the source users using the user similarity models. This may be because the Gaussians constructed by the random methods consist of feature vectors that are very different from those of the target user, and then they had harmful effects on the estimation of the Gaussian transformations. On the other hand, the Gaussians constructed by the PC information-based methods included small amounts of 'harmful' feature vectors. That is, we consider the appropriate selection of training data can boost the effects of the adaptation techniques.

Finally, we focus on RND+Tree and PC+Tree. The lowest error rate of RND+Tree was 5.32% when the method used training data from all source users, i.e., n was 39. On the other hand, PC+Tree achieved a 4.91% error rate when n was 34, which was the lowest error rate of all six methods and was 0.41% better than that of RND+Tree. That is, by using PC information, the error rate decreased by 0.41%. Also, the lowest error rate of PC+Tree was better than that of PC+MLLR by 0.15%. While we could not outperform the error rate of the DPN method (4.79%), the lowest error rate of PC+Tree was almost identical to that of the DPN method. Also, we could reduce the error rate by 1.94% compared with the lowest error rate obtained with the naive method (RND) whose lowest error rate was 6.85%. That is, we achieved a 28.3% error reduction rate compared with the naive method $(1.94/6.85 = 0.283)$.

As shown in Fig. 7, the transitions of the error rates were not very stable. In particular, the transitions of the random methods were not stable. That is, the error rates are greatly affected by the way the source users are selected. However, the lowest error rates of any random methods when n was smaller than 39 were poorer than the error rates when n was 39. On the other hand, the lowest error rates of the PC information-based methods when n was smaller than 39 were better than those when n was 39. We consider that this was achieved by appropriately selecting training data with the user similarity models when we learned the target users' activities.

Here we discuss how to find a good n value in an actual activity recognition system. The technique is very simple, namely, we find a good n by using labeled sensor data of source users prepared in advance with a cross validation.

4.5. Effect of physical characteristics

When we have insufficient source users' sensor data that are similar to those of a target user, the activity recognition performance for the target user is poor. For example, only eight participants had a dominant left hand in ping-pong, and their error rate with RND+Tree was 6.78% when n was 39, which was poorer than that for right-handed target users (4.98%). Also, only five participants used a hand-held cleaner, and their error rate of RND+Tree was 7.31% when n was 39, which was much poorer than that for target users who used an upright cleaner (5.06%). As mentioned above, by using the user similarity models, we can reduce the error rates. Here we found that the effects of the user similarity models differed according to the target users' physical characteristics. When we employed PC+Tree, the error rate for target users who used a hand-held cleaner decreased by 1.60% when n was 25. On the other hand, the error rate for target users who used an upright cleaner decreased by only 0.45% when n was 34. In addition, when we employed PC+Tree, the error rate for left-handed target users decreased by 1.51% when n was 37. However, the error rate for right-handed target users decreased by only 0.34% when n was 34. That is, even if a target user is a minority user, e.g., she is left-handed, our method can recognize her activities with relatively good accuracy.

5. Conclusion

This paper proposed a new activity recognition method that models the activities of an end user simply by employing information about the physical characteristics of the end user. Our method does not require the end user to collect and label her sensor data. Furthermore, our method can improve recognition performance without any effort by the end user because the method adapts the activity models to the end user by using her test data. In the evaluation, we confirmed that our training data selection method and adaptation method effectively improved the recognition performance. Moreover, we found that our training data selection method boosted the effects of the adaptation techniques.

References

[1] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive 2004*, pages 1–17, 2004.

[2] A. Dempster, N. Laird, D. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[3] K. Forster, A. Biasiucci, R. Chavarriaga, J. del R. Millan, D. Roggen, and G. Troster. On the use of brain decoded signals for online user adaptive gesture recognition systems. In *Pervasive 2010*, pages 427–444, 2010.

[4] J. Gauvain and C. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2):291–298, 2002.

[5] T. Huynh and B. Schiele. Towards less supervision in activity recognition from wearable sensors. In *Int'l Symp. on Wearable Computers*, pages 3–10, 2006.

[6] G. Krassnig, D. Tantinger, C. Hofmann, T. Wittenberg, and M. Struck. User-friendly system for recognition of activities with an accelerometer. In *PervasiveHealth 2010*, pages 1–8, 2010.

[7] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language*, 9(2):171–185, 1995.

[8] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome. Object-based activity recognition with heterogeneous sensors on wrist. In *Pervasive 2010*, pages 246–264, 2010.

[9] R. Ohmura, N. Hashida, and M. Imai. Preliminary evaluation of personal adaptation techniques in accelerometer-based activity recognition. In *Int'l Symp. on Wearable Computers: Late Breaking Results*, 2009.

[10] M. Perkowitz, M. Philipose, K. Fishkin, and D. Patterson. Mining models of human activities from the web. In *WWW 2004*, pages 573–582, 2004.

[11] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[12] N. Ravi, N. Dandekar, P. Mysore, and M. Littman. Activity recognition from accelerometer data. In *IAAI 2005*, volume 20, pages 1541–1546, 2005.

[13] S. Shevade, S. Keerthi, C. Bhattacharyya, and K. Murthy. Improvements to the SMO algorithm for SVM regression. *IEEE Trans. on Neural Networks*, 11(5):1188–1193, 2002.

[14] M. Stikic, K. Van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *Int'l Symp. on Wearable Computers*, pages 81–88, 2008.

[15] T. Hori et al. Real-time meeting recognition and understanding using distant microphones and omni-directional camera. In *IEEE Workshop on Spoken Language Technology*, pages 412–417, 2010.

[16] T. van Kasteren, G. Englebienne, and B. Krose. Transferring knowledge of activity recognition across sensor networks. In *Pervasive 2010*, pages 283–300, 2010.

[17] G. Welk and J. Differding. The utility of the digi-walker step counter to assess daily physical activity patterns. *Medicine & Science in Sports & Exercise*, 32(9):S481–S488, 2000.

[18] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2004.