Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data

Young-Seol Lee, Sung-Bae Cho*

Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, South Korea

A R T I C L E   I N F O

Article history:
Received 27 February 2012
Received in revised form 20 March 2013
Accepted 30 May 2013

Keywords:
Mixture-of-experts
Co-training
Activity recognition
Android phone

A B S T R A C T

As the number of smartphone users has grown recently, many context-aware services have been studied and launched. Activity recognition becomes one of the important issues for user adaptive services on mobile devices. Even though many researchers have attempted to recognize a user's activities on a mobile device, it is still difficult to infer human activities from uncertain, incomplete and insufficient mobile sensor data. We present a method to recognize a person's activities from sensors in a mobile phone using mixture-of-experts (ME) model. In order to train the ME model, we have applied global-local co-training (GLCT) algorithm with both labeled and unlabeled data to improve the performance. The GLCT is a variation of co-training that uses a global model and a local model together. To evaluate the usefulness of the proposed method, we have conducted experiments using real datasets collected from Google Android smartphones. This paper is a revised and extended version of a paper that was presented at HAIS 2011.

© 2013 Published by Elsevier B.V.

1. Introduction

Smart phones, such as Google Android phone and Apple iPhone, have recently incorporated diverse and powerful sensors. The sensors include GPS receivers, microphones, cameras, light sensors, temperature sensors, digital compasses, magnetometers and accelerometers. Because of the small size and the superior computing power of the smartphones, they can become powerful sensing devices that can monitor a user's context in real time. Especially, Android-based smart phones are suitable platforms to collect sensing information because the Android operating system is free, open-source, easy to program, and expected to become a dominant smartphone in the marketplace.

Many context-aware services were introduced and they tried to provide a user with convenience on a mobile phone. For instance, there are social networking applications such as Facebook [1] and MySpace [2]. Foursquare provides a location-based social networking service. It ranks the users by the frequency of visiting a specific location and encourages them to check in the place. Loopt service helps users to share friends' locations. Davis et al. attempted to use temporal, spatial, and social contexts to manage multimedia content with a camera phone [3]. Until now, most of the commercial services use only raw data like GPS coordinates.

In the mobile environment, inferring context information becomes an important issue for mobile cooperative services. By capturing more meaningful context in real time, we can develop more adaptive applications to the changing environment and user preferences. Activity recognition technology is one of the core technologies to provide user adaptive services. Many researchers have attempted to infer high-level semantic information from raw data collected in a mobile device. Bellotti et al. developed a Maggiti system which predicted a user's activities (eating, shopping, etc.) and provided suitable services on Windows Mobile platform [4]. Kwapisz et al. studied a method to recognize a user's behavior using cell phone accelerometers [5]. Chen proposed intelligent location-based mobile news service as a kind of location-based service [6]. Other researchers have studied context-aware systems considering efficient energy management. Paek et al. controlled GPS receiver to reduce energy consumption on a smartphone [7]. Ravi et al. predicted battery charge cycle and recommended the charge [8]. Other researchers tried to stop unnecessary functionalities in the context. It is difficult to recognize a user's situation because of uncertainty and incompleteness of context in mobile environment.

Most of the research used various statistical analysis and machine learning techniques such as probabilistic model, fuzzy logic, and case based reasoning. However, it is still difficult to apply them to mobile devices because of two problems. First, although the machine learning techniques are appropriate to deal
with vagueness and uncertainty in mobile environment, it is not easy to classify user's activities using only one classifier from incomplete mobile data in a complex environment. The other problem is that many supervised machine learning techniques need hand-labeled data samples for good classification performance. It requires users to annotate the sensor data. In many cases, unlabeled data are significantly easier to come by than labeled ones [9,10]. These labeled data are fairly expensive to obtain because they require human effort [11]. Therefore, we would like our learning algorithm to be able to take as much advantage of the unlabeled data as possible.

In this paper, we present an activity recognition system using mixture of experts (ME) [12] on Android phone. ME is one of the divide-and-conquer models which are effective solutions to overcome the limitations of mobile environment. The ME is based on several local experts to solve smaller problems and a gating network to combine the solutions from separate experts, each of which is a statistical model for a part of overall problem space. In order to train the ME with labeled and unlabeled data, the global–local co-training (GLCT) method is used. GLCT is one of the variations of co-training [13], and enables us to improve the performance of ME model by minimizing errors which occur due to the unlabeled data. To evaluate the usefulness of the proposed system, we conducted experiments using real datasets collected from Google Android smartphones.

Table 1

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Classifier</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank et al. [18]</td>
<td>2011</td>
<td>Geometric template matching (GTM)</td>
<td>Demonstration for activity and gait recognition system</td>
</tr>
<tr>
<td>Sun et al. [19]</td>
<td>2010</td>
<td>Support vector machine (SVM)</td>
<td>Activity recognition with an accelerometer considering people's pocket position</td>
</tr>
<tr>
<td>Brezmes et al. [20]</td>
<td>2009</td>
<td>K-nearest neighbor</td>
<td>Activity recognition with no server processing data for a reasonable low cost</td>
</tr>
<tr>
<td>Yang et al. [22]</td>
<td>2009</td>
<td>Decision tree, k-means clustering, HMM</td>
<td>Using decision tree to classify physical activities and HMM to consider activity sequences</td>
</tr>
<tr>
<td>Ravi et al. [23]</td>
<td>2005</td>
<td>Decision tables, C4.5 Decision trees, k-nearest neighbor, SVM, naive Bayes</td>
<td>Combining classifiers using Plurality Voting for activity recognition from a single accelerometer</td>
</tr>
<tr>
<td>Bao et al. [24]</td>
<td>2004</td>
<td>Decision tables, instance based learning, C4.5 decision trees, naive Bayes</td>
<td>Recognizing a variety of household activities using acceleration for context-aware computing</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Sensors</th>
<th>Classifier</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hong et al. [25]</td>
<td>2010</td>
<td>Accelerometer, RFID tags</td>
<td>Decision tree</td>
<td>Hierarchical approach for robust and accurate recognition</td>
</tr>
<tr>
<td>Wang et al. [26]</td>
<td>2009</td>
<td>Accelerometer, microphone, GPS, WiFi, Bluetooth</td>
<td>Threshold based heuristics</td>
<td>Sensor management for mobile devices to monitor user state</td>
</tr>
<tr>
<td>Choudhury et al. [27]</td>
<td>2008</td>
<td>Accelerometer, barometer, compass, humidity, temperature</td>
<td>Probabilistic model</td>
<td>Using Mobile Sensing Platform (MSP) for activity recognition</td>
</tr>
<tr>
<td>Consolvo et al. [28]</td>
<td>2008</td>
<td>Accelerometer, barometer, microphone, humidity, etc.</td>
<td>Boosted decision stump classifiers</td>
<td>System for on-body sensing, and activity inference, and three-week field trial</td>
</tr>
</tbody>
</table>

Please cite this article as: Y.-S. Lee, S.-B. Cho, Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.05.044
conquer principle, the ME model should work well for problems composed of smaller unconnected ideas. The experts can be modeled to give a solution to these smaller sub-problems, while the combination of the solutions from the expert will be given by the gating network. Each expert deals with different features from a different perspective, thereby resolving the small separable problems. Through the features of the model, it has been applied to traditional complicated recognition problems [29, 30]. One of the considerations when implementing ME models is the way to separate the input data space and generate experts. In order to divide the input data space into several subgroups, clustering techniques can be applied as shown in [31]. Some clustering techniques such as k-means clustering and hierarchical clustering have been widely used.

In supervised learning, local experts in the ME model can be trained easily by using conventional machine learning techniques, for example, MLP [32], support vector machine (SVM) [33] or probabilistic model [34]. However, in semi-supervised learning, it requires an extra learning method to deal with unlabeled data.

The simplest way to train the ME model with both labeled and unlabeled data is to apply previous semi-supervised learning methods such as individual-training or co-training directly to the model. However, it is already known that local experts are sensitive about the number of the training instances and can be biased or the variance can be increased when there are only a few amounts of instances [35]. Especially in semi-supervised learning which uses only a small amount of labeled data, the initial ME model may not be accurate enough to evaluate unlabeled data. In order to overcome this limitation, the semi-supervised learning method specialized to the ME model is required.

The EM algorithm to train the ME model in semi-supervised approach was proposed for this reason. Nevertheless, it has a limitation caused by the model assumption of EM algorithm mentioned in the previous section, and another enhanced method is still required.

In this paper, we use the GLCT method to train the ME model. The method is specialized to train the ME model since it overcomes the imperfect model problem related to the lack of training instances by using the global model together.

### 3. Activity recognition using ME

The proposed activity recognition system consists of three steps: to collect sensor data, preprocess the data and recognize a user’s activity. First, after sensor data are collected from sensors on a smartphone, the data are transferred to preprocessing units for extracting features such as mean and standard deviation. The following Eqs. (1)–(3) denote the features such as the difference between previous acceleration and current acceleration, average acceleration, and standard deviation of acceleration that are extracted from the datasets.

\[
\text{sum}_X = \sum_{i=1}^{N} \sqrt{(x_i-x_{i-1})^2}
\]  
\[
\text{mean}_X = \frac{\sum_{i=1}^{N} (x_i-x_{i-1})^2}{N}
\]  
\[
\text{std}_X = \sqrt{\left( \sum_{i=1}^{N} (x_i-x_{i-1})^2 - \text{mean}_X \right)^2}
\]

After the preprocessing step, ME model is used to recognize a user’s activities from the feature set. Fig. 1 shows the process of the entire system.

#### 3.1. Mixture-of-experts construction

Prior to applying the ME model to activity recognition, local experts must be built with separate dataset. Each expert deals with only a part of dataset. When dividing entire data into several subgroups, each subgroup should guarantee similarities between members in it. This is required to give specialties to local experts in distinguishing some similar patterns even though they are only a part of entire data as shown in Fig. 2.

The clustering technique is used to define boundaries to estimate similarities between data. K-means clustering approach is a method of cluster analysis which aims to assign n data items into k sets in which each item belongs to the set with the nearest mean value. It is a simple method which is easy to implement and shows good performance on common dataset. The distance from the centroids to data items can be calculated with various measures such as Euclidean distance and Pearson correlation coefficient. K-means clustering consists of the following process of five steps.

1. Determine \( k \), the number of clusters.
2. Calculate \( k \) centroids.
3. Calculate distances of data items to centroids.
4. Assign data items into centroids in terms of the nearest centroid.
5. Iterate (2)–(4) until there is no change of \( k \) sets.

In order to implement the global and the local models including local experts and the gating network, MLP was used for all the models. For the global model, conventional MLP was applied directly. The ME model was designed to include the five local experts in total. Here, MLP is employed as a base classifier for each expert in mixture-of-local-experts. Fig. 3 depicts the structure of the ME model used. To train the ME model, learning technique for MLP-based ME model [32] is used. \( f_i(X) \) in Eqs. (4) and (5) denotes the output of the mixture-of-experts model where \( f_i \) is the ith output of gating network and \( f_i(X) \) is the output of the ith local...
expert. X is an input vector with multivariate attributes.

\[ f_l(X) = \sum_{i=1}^{k} \pi_{i}(X) \]

Each neural network in the model used sigmoid function like Eq. (6) as activation function for a hidden layer and an output layer.

\[ a(x) = \frac{1}{1 + e^{-x}} \]

### 3.2. GLCT algorithm

GLCT algorithm is summarized in Fig. 4. Given a set L of labeled training instances, we can obtain the global model M_G and the ME model M_l which consists of N experts \( \mu_1, \mu_2, \ldots, \mu_N \) and the gating network g. The algorithm iteratively extracts the instances which are going to be labeled based on two criteria, preventing degradations and expecting improvements.

Since the models can be imperfect and show poorer performance than the opponent in some conditions, the models should carefully predict labels, especially when there are not sufficient training data for local experts to generalize the entire problem. In order to prevent degradations due to model imperfections, the algorithm compares the confidences from M_G and M_l. The models can provide labeled instances for the opponents only if each model is more confident than the other one. The instances satisfying the first criterion are added to candidate sets \( U_1 \) or \( U_G \).

After obtaining the candidate sets, the sets are sorted by considering confidence degrees in descending order. At the next step, at most N instances for each class with higher confidence degree than the certain threshold \( \epsilon \) are chosen gradually from the top of the set to be labeled. By choosing instances with sufficient confidence, it is expected that the models would be improved at the next iteration.

The chosen instances are added to the training dataset for each model, \( L_G \) and \( L_l \). The models are trained with newly extended
labeled dataset. If \( L_G \) and \( L_L \) do not change, the training stops. Afterwards, the algorithm iterates the training procedure until no additional data samples for \( L_L \) and \( L_G \) remain.

In the co-training algorithm, the local and the global models are based on feed-forward neural networks trained with back-propagation learning. The neural networks are repeatedly trained by gradient descent with the error function on given training set from the other model. Newly labeled data by a global model are used to train a local model in the next step and vice versa. In Eqs. (7) and (8), \( E_L \) and \( E_G \) denote error functions in the additional training set for a local model and a global model, respectively.

\[
E_L = \|\mathbf{O}_L - \sum \kappa_i f_L(x)\| \tag{7}
\]

\[
E_G = \|\mathbf{O}_G - f_G(x)\| \tag{8}
\]

In Eq. (7), an error function for a local model, \( \mathbf{O}_L \) is the output vector of a global model, \( \kappa_i \) is the gating value of the \( i \)th local expert, and \( f_L(x) \) is the output vector of the \( i \)th local expert. \( \mathbf{O}_L \) is the output vector of a local model and \( f_L(x) \) is the output vector of a global model. In detail, \( \mathbf{O}_L \) in Eq. (7) can be replaced with the output function of a global model at the previous step, \( f_L' \) and \( f_G(x) \) and output function of a local model at the previous step, \( \sum \kappa_i f_L(x) \) can be substituted for \( \mathbf{O}_L \) in Eq. (8) as shown in Eqs. (9) and (10).

\[
E_L = \|f_L'(x) - \sum \kappa_i f_L(x)\| \tag{9}
\]

\[
E_G = \|\sum \kappa_i f_L'(x) - f_G(x)\| \tag{10}
\]

Ideally, the co-training algorithm is stopped when there is no difference between current error and previous error as illustrated in Eq. (11).

\[
\frac{\partial E_L}{\partial x} = 0 \quad \text{and} \quad \frac{\partial E_G}{\partial x} = 0 \tag{11}
\]

However, it is practically difficult to satisfy the condition because it is satisfied only when two different views have the same error value or no change of error values from additional training set. In this paper, we determine when the co-training is stopped by the number of additional training sets \( N \) for a global model and a local model. The number of additional training sets \( N(L_L) \) and \( N(L_G) \) are decided by the confidence degree, introduced in the next section.

co-training is stopped when \( N(L_L) = 0 \) and \( N(L_G) = 0 \) \hspace{1cm} (12)

### 3.3. Measuring confidence degrees

In order to obtain confidence degrees of examples, we design the measure for confidences. Let \( \text{conf}(w,x) \) be the original confidence degree for input instance \( x \) defined by the corresponding model \( w \). \( \text{conf}(w,x) \) can be changed by the type of model used to build the global or the local model. In this work, we implemented both the global and the local models by using MLP. The output of MLP can be estimated as confidence degrees, which can be simply defined as follows:

\[
\text{conf}(w,x) = [w(x)|w(x) \text{ is output of } w \text{ for } x] \tag{13}
\]

Since the original problem space is divided into several subspaces to generate local experts, the number of training data for each expert is rather smaller than the global model. This can make the ME model have large bias or variance, because there are chances that some of the training instances for certain class are missing in some sub-regions even though the original space contains all classes. Since some experts have not learned about certain classes, they can generate unexpected values as outputs for the missing classes according to the type of models used as local experts. To prevent this problem, the existence of training instances of certain class should be checked prior to computing confidence degrees. Eqs. (14) and (15) show the modified confidence degree for class \( C \) of the global model and the ME model, respectively.

\[
\text{conf}(M_G,x,C) = \begin{cases} 
M_{G,C}(x), & \text{if } N_{M_G,C} > 0 \\
0, & \text{otherwise}
\end{cases} \tag{14}
\]

\[
\text{conf}(M_G,x,C) = \sum_{i=1}^{N} g_i \text{conf}(\mu_i, x, C) \tag{15}
\]

\( \mu_i \) is the \( i \)th local expert, and \( g_i \) is the gating network. \( M_{G,C}(x) \) and \( \mu_i(x) \) represent the outputs of \( M_G \) and \( \mu_i \) for class \( C \), respectively. \( g_i \) is the gating output for the \( i \)th local expert and \( N_{M_G,C} \) and \( N_{\mu_i,C} \) are

---

**Table 3**

A summary of the datasets used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Features</th>
<th>#Classes</th>
<th>#Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration only</td>
<td>6</td>
<td>3</td>
<td>1107</td>
</tr>
<tr>
<td>Acceleration, orientation, magnetic field, etc.</td>
<td>22</td>
<td>5</td>
<td>661</td>
</tr>
</tbody>
</table>

---

**Fig. 5.** (a) Class and datasets, and (b) Android phone based data collection interface.
the numbers of training instances for class $C$ used to train $M_C$ and $\mu_i$, respectively.

4. Experiments

This section presents the experiments conducted to evaluate the usefulness of the proposed method on the pattern classification problems. The number of hidden nodes for each neural network is fixed to ten and the learning rate is 0.3. There are two objectives of the experiments. Firstly, it is to prove the usefulness of co-training based on unlabeled data. Secondly, our proposed method shows better performance than semi-supervised self-training as [36].

As shown in Fig. 2, we divide the dataset into some groups to learn local expert models in our mixture of experts. In the initial stage, the labeled data are used to learn the model. After the first learning, the unlabeled data are labeled using the previous learned models.

4.1. Android phone datasets

For the experiments, the evaluation of the proposed method is conducted on two Android phone datasets as shown in Table 3. The two types of datasets consist of acceleration, orientation, and magnetic fields to classify a user's transportation mode which are collected from Android OS smart phones using an interface as shown in Fig. 5(b). The dataset can be downloaded and tested in the author's homepages (http://sclab.yonsei.ac.kr/movingstatus.csv and http://sclab.yonsei.ac.kr/movingstatus_2.csv).

For all datasets, data instances were partially labeled depending on the ratio of the instances labeled. Labeled data instances were chosen randomly by ignoring class distributions. The class labels are still, walk, and run in dataset 1 and still, walk, vehicle, run, and subway in dataset 2 as shown in Fig. 5(a).

4.2. Experimental results with android phone datasets

Firstly, we conduct experiments to show the improvement of performance by using unlabeled data. For each experiment, the performances of models trained in three different conditions that are trained with all labeled data, only with partially labeled data and with both labeled and unlabeled data were evaluated by 5-folds and 10-folds cross validation. All experiments were performed ten times to obtain the average accuracy.

First of all, we conducted learning process for two different datasets to show the feasibility of the proposed method. Especially, in order to show that GLCT performs well regardless of the ratio of initial labeled data, we changed the ratio of labeled instances to observe performances according to the amount of initial labeled data. Tables 4 and 5 and Fig. 6 show the result of the test on dataset 1 and Tables 5–7 and Fig. 7 show the result of dataset 2. As the result, the model trained with both labeled and unlabeled data showed better performance than the model with only labeled data. There is no big difference between 5-folds and 10-folds cross validation. The standard NN and ME models were trained well for all datasets by the proposed GLCT method. This

**Table 4**

Accuracy comparison for each algorithm on dataset 1 (5-folds).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>No. of labeled data</th>
<th>No. of unlabeled data</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture of local experts</td>
<td>1107</td>
<td>0</td>
<td>91.70 (± 7.61)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>55</td>
<td>1052</td>
<td>75.74 (± 14.56)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>55</td>
<td>1052</td>
<td>86.34 (± 11.11)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>110</td>
<td>997</td>
<td>80.87 (± 11.05)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>110</td>
<td>997</td>
<td>90.17 (± 5.42)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>166</td>
<td>941</td>
<td>83.04 (± 9.12)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>166</td>
<td>941</td>
<td>91.78 (± 4.72)</td>
</tr>
</tbody>
</table>

**Table 5**

Accuracy comparison for each algorithm on dataset 1 (10-folds).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>No. of labeled data</th>
<th>No. of unlabeled data</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture of local experts</td>
<td>1107</td>
<td>0</td>
<td>92.56 (± 1.05)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>55</td>
<td>1052</td>
<td>76.28 (± 7.39)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>55</td>
<td>1052</td>
<td>87.27 (± 9.92)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>110</td>
<td>997</td>
<td>81.07 (± 5.59)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>110</td>
<td>997</td>
<td>90.25 (± 5.82)</td>
</tr>
<tr>
<td>Mixture of local experts</td>
<td>166</td>
<td>941</td>
<td>83.48 (± 5.10)</td>
</tr>
<tr>
<td>Co-training based ME</td>
<td>166</td>
<td>941</td>
<td>91.78 (± 4.72)</td>
</tr>
</tbody>
</table>

![Fig. 6. Performances according to the ratio of labeled instances for dataset 1.](image-url)
result implied that the GLCT can be used as the training method for the ME model in semi-supervised approach.

The performance of the model was improved in every case from 7% to 10% for dataset 1 and from 7% to 8% for dataset 2. With only 5% initial labeled data, the model showed the most significant performance improvement for dataset 1. However, the results for dataset 2 did not show the big difference in performance improvement.

Table 8 shows the result of paired t-tests to examine the significant difference between ME model trained with only partially labeled data and that trained with labeled and unlabeled data. The t-value is greater than 4.78 for all tests with the degree of freedom 9 and the probability is less than 0.001, indicating that the two models are significantly different in their accuracy with 99.9% confidence.

On the other hand, Table 9 summarizes the result of F-tests to evaluate the statistical difference between ME model trained with only partially labeled data and that trained with labeled and unlabeled data. The values in Table 9 are greater than threshold for all tests, and it implies that the two models are significantly different with 99.9% confidence.

Finally, we compared GLCT with the alternative way to train the ME model to show the superiority of GLCT. In this experiment, the...
individual-training algorithm [36,37] was used as the alternative technique to train the ME. As mentioned in Section 2, it is one of the conventional semi-supervised learning techniques to train the ME. Fig. 8 shows the algorithm of the individual-training.

For the experiment, the dataset 1 and 2 with 5%, 10%, 15% labeled instances were used as shown in Table 10. GLCT produces significantly improved performance than the individual-training algorithm as shown in Figs. 9 and 10. Figs. 11 and 12 show the change of accuracies by iterating co-training and individual-training methods, respectively. Initial accuracies of the two methods were similar, but as iterations increased, the GLCT yielded greater improvement in performance than the individual-training. In many cases, the performance of individual-training tends to get worse as the iteration goes by.

Table 10

<table>
<thead>
<tr>
<th>Label ratio (%)</th>
<th>Dataset1</th>
<th>GLCT</th>
<th>Individual-training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset1</td>
<td>5</td>
<td>86.34 (± 11.11)</td>
<td>52.16 (± 5.91)</td>
</tr>
<tr>
<td>Dataset2</td>
<td>5</td>
<td>68.09 (± 3.69)</td>
<td>45.33 (± 6.27)</td>
</tr>
<tr>
<td>Dataset1</td>
<td>10</td>
<td>90.17 (± 5.42)</td>
<td>61.12 (± 3.75)</td>
</tr>
<tr>
<td>Dataset2</td>
<td>10</td>
<td>61.85 (± 2.94)</td>
<td>49.81 (± 3.90)</td>
</tr>
<tr>
<td>Dataset1</td>
<td>15</td>
<td>90.81 (± 2.96)</td>
<td>66.08 (± 5.38)</td>
</tr>
<tr>
<td>Dataset2</td>
<td>15</td>
<td>68.10 (± 2.63)</td>
<td>57.04 (± 3.85)</td>
</tr>
</tbody>
</table>

Fig. 9. Comparison between co-training and individual-training for dataset 1.

Fig. 10. Comparison between co-training and individual-training for dataset 2.

Fig. 11. Change of accuracies by iterating two methods for dataset 1.
5. Concluding remarks

In this paper, we have proposed an activity recognition system using the mixture-of-experts (ME) model with both labeled and unlabeled data. ME is a variant of divide-and-conquer paradigm, and it is suitable to solve complex problem to recognize a user’s activity from uncertain, and incomplete mobile sensor data. The GLCT method is also used to train the ME model. In order to overcome the limitation of the local experts with small amount of training data, we employed the GLCT method to improve the ME model. GLCT is a method to build a model in which unlabeled data can be used to augment labeled data, based on having two views (global model and mixture of experts). The two views are redundant but not completely correlated. It is assumed that unlabeled instances can potentially be used to increase the accuracy of the model. The contribution lies in applying ME model and the GLCT method to activity recognition on a mobile phone.

To show the usefulness of the proposed system, we conducted experiments using datasets collected from Google Android smartphones. The result of the experiments showed that GLCT could train the ME model well by predicting unlabeled instances accurately, and outperforming the individual training algorithm for the ME model. We can use co-training applied to the large volume of unlabeled acceleration data, to improve the accuracy of activity recognition. It can also be applied to activity recognition using multiple sensors such as orientation, magnetic field and so on. The important point is that a large volume of unlabeled instances can easily be collected in real world, but the labeled instances are very expensive. GLCT can be a method to use the unlabeled data for context recognition.

Even though the performance of GLCT was promising in experiments, the method should be justified with theoretical analysis. In the future, we are planning to prove that the ME model is learnable by GLCT theoretically. We also have to apply this method to other domains which have data with missing features, and analyze the effectiveness to maximize performance.

Acknowledgment

This research was supported by the Industrial Strategic Technology Development Program (10044828) funded by the Ministry of Trade, Industry and Energy (MI, Korea), and the ICT R&D Program 2013 funded by the MSIP (Ministry of Science, ICT & Future Planning, Korea).

References


Young-Seol Lee is a Ph.D. candidate in computer science at Yonsei University. His research interests include Bayesian networks and evolutionary algorithms for context-aware computing and intelligent agents. He received his M.S. degree in computer science from Yonsei University.

Sung-Bae Cho received the B.S. degree in computer science from Yonsei University, Seoul, Korea and the M.S. and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Taegyeon, Korea. He was an invited Researcher of Human Information Processing Research Laboratories at Advanced Telecommunications Research Institute, Tokyo, Japan from 1993 to 1995, and a Visiting Scholar at University of New South Wales, Canberra, Australia in 1998. He was also a Visiting Professor at University of British Columbia, Vancouver, Canada from 2005 to 2006. Since 1995, he has been a Professor in the Department of Computer Science, Yonsei University. His research interests include neural networks, pattern recognition, intelligent man–machine interfaces, evolutionary computation, and artificial life. Dr. Cho was awarded outstanding paper prizes from the IEEE Korea Section in 1989 and 1992, and another one from the Korea Information Science Society in 1990. He was also the recipient of the Richard E. Merwin prize from the IEEE Computer Society in 1993. He was listed in Who’s Who in Pattern Recognition from the International Association for Pattern Recognition in 1994, and received the best paper awards at International Conference on Soft Computing in 1996 and 1998. Also, he received the best paper award at World Automation Congress in 1998, and listed in Marquis Who’s Who in Science and Engineering in 2000 and in Marquis Who’s Who in the World in 2001. He is a Senior Member of IEEE and a Member of the Korea Information Science Society, INNS, the IEEE Computational Intelligence Society, and the IEEE Systems, Man, and Cybernetics Society.