Less Annotation on Personalized Activity Recognition Using Context Data

Lisha Hu✉*†‡, Yiqiang Chen✉*†, Shuangquan Wang*†, Jindong Wang*†‡

*Beijing Key Laboratory of Mobile Computing and Pervasive Device,Institute of Computing Technology,CAS
†Pervasive Computing Research Center,Institute of Computing Technology, Chinese Academy of Sciences
‡University of Chinese Academy of Sciences, Beijing, China

Email:{hulisha,yqchen,wangshuangquan,wangjindong}@ict.ac.cn

***Abstract*—Miscellaneous mini-wearable devices (e.g. wristbands, smartwatches, armbands) have emerged in our life, capable of recognizing activities of daily living, monitoring health information and so on. Conventional activity recognition (AR) models deployed inside these devices are generic classifiers learned offline from abundant data. Transferring generic model to user-oriented model requires time-consuming human effort for annotations. To solve this problem, we propose *SS-ARTMAP-AR*, a self-supervised incremental learning AR model updated from surrounding information such as Bluetooth, Wi-Fi, GPS, GSM data without user's annotation effort. Experimental results show that *SS-ARTMAP-AR* can gradually adapt individual users and become more and more incremental intelligence.**

***Index Terms*—activity recognition, activities of daily living, smartwatch, incremental learning.**

I. Introduction

Knowledge of activities of daily living (ADLs, e.g. walking, running, going upstairs/downstairs) can be learned by making use of various wearable devices (e.g. wristband, smartwatch, armband [3][4][5]) worn or even embedded into clothes or accessories [1][2], hence wearable activity recognition (AR) techniques promote the development of ubiquitous computing applications such as context awareness, energy expenditure and personal healthcare. For example, wearable AR techniques for healthcare improve user's health conditions by collecting information related to ADLs, analyzing it and returning the feedback to the user.

An AR classification model is usually learned offline based on the data collected from several users. Once done, this model is embedded into the wearable devices permanently to recognize ADLs for all future unknown users. Normally, such a generic static model may not well fit for a specific user with distinctive personalities in terms of wearing styles and ADLs. For example, a model learned based on the data from the dominant wrist (e.g. right wrist [6][7]) may not well work for the users wearing device on his/her non-dominant wrist (e.g. left wrist), or having an opposite dominant wrist. Therefore, it would be better if the model can adjust itself adaptively by considering the data of its single user to precisely grab his personalities.

To deal with the problem, a lot of incremental learning [8][9][10] and transfer learning methods [11][12] have been proposed in literatures to adapt AR models to new domains or individuals. In this paper, we focus on the incremental learning mechanism. Generally, incremental learning method consists of two stages to learn a classification model [8]. An initial model is created firstly based on a few labeled instances as the training set, and then updated automatically if new (labeled/unlabeled) instances come. According to the various types of instances (labeled, unlabeled, unlabeled with new features/attributes) which could be utilized in the second stage, incremental learning mechanism can be divided into three categories: supervised [8], semi-supervised [9] and self-supervised incremental learning [10].

It is unrealistic to utilize supervised incremental learning mechanism to update an AR model consistently. Too much user annotations are required to generate labeled instances for updating the model, which obviously interrupts user's daily living. Gladly, both semi-supervised and self-supervised incremental learning methods do not need user annotation. However, for semi-supervised incremental learning mechanism, continuously learning unsupervised knowledge (learned from unlabeled instances in the second stage) may degrade its supervised knowledge (learned from labeled instances in the first stage)[10]. Self-supervised incremental learning, on the other hand, also learns from new experiences (more features within unlabeled instances) without, in a certain level, degrading its supervised knowledge [10], which makes it more feasible to personalized AR model adaptation. In this paper, we concentrate on the challenges of constructing an AR model in the self-supervised incremental learning mechanism.

In the first stage of self-supervised incremental learning, a generic model is learned based on the data of motion sensors (accelerometer in this paper) collected from several users' ADLs. In the second stage, other data sources besides motion sensors are needed to extract new features for the unlabeled instances of the target user. Recently, dynamic contextual information has shown its superiority in recognizing user's ADLs effectively and unobtrusively. For example, the number and change rate of Bluetooth device scanned in user's surroundings can help to identify whether he is taking the subway or not [13][14]. As a consequence, features related to contextual information of the target user are extracted as well in the second stage of self-supervised incremental learning. Our contributions are:

- A self-supervised incremental learning framework is proposed to construct the personalized AR model for the target user.
- Several AR models are constructed by automatically mining its target user's personalities, learning from the knowledge of nearby surroundings, and updating themselves accordingly.
- We propose to take advantage of the contextual information in user's surroundings to help personalize a generic model. According to the consistent discovery of the target user's particular patterns in his daily life, the model gradually suits its target user better.
- We validate our method on real-world AR datasets. Experiments show that those AR models can recognize user's ADLs accurately based on only a few labeled and plenty of unlabeled data of the target users.

The remainder of this paper is structured as follows. Section II discusses some related work. We present an overview of the self-supervised learning (SSL) and Self-supervised ARTMAP model in Section III. After that, we give detailed description about the real-world datasets and preprocess steps in Section IV. We validate the performance of our approach in Section V. In the end, we conclude the paper and discuss some future extensions.

II. Related Work

In this section, we provide a brief introduction of existing AR works in literature from different perspectives.

A. Model: offline batch mode learning/online incremental learning

Generic AR models are conventionally constructed offline based on the data collected beforehand. Once completed, the model remains unchanged regardless of whoever its user is or what kind of scenarios it is utilized. For instance, the Kung Fu model in the wristband [15] is a static hidden Markov model and could not adapt itself to a new user. The data distribution is greatly different between varying users.

Recently, more and more researchers realize the phenomenon that the distribution of data is heavily affected by varying users. The performance will degrade when the model trained on some users is used to a new user. To solve this problem, a number of online incremental learning models have been proposed. [8][16][17]. An Online Sequential Reduced Kernel Extreme Learning Machine is applied to update the initial model and adapt it to new users based on the recognition results of high confidence level [8].A lifelong learning framework from sensor data streams is constructed for predicting user modeling, which continuously learns from their users and adapts the system through stream-based active learning [16]. New task can be learned semi-incrementally from a partial decision tree model which captures knowledge from a previous task [17].

B. Annotation: supervised/semi-supervised learning

Many researches in AR community focus on discovering specialized features from readings of sensor data in distinct activities. For them, labels (activities) of all the data need to be annotated during data collection process. In order to dig daily routines from ADLs, users are asked to annotate fine-grained activities (e.g. washing hands, picking up food), as well as the daily routines such as commuting or working [18]. Authors in [2] manually annotate data of elder's twelve daily activities based on the videos collected from safety goggles of "Smart Glasses". Another way to obtain the labels of AR data is: user starts and stops each activity by orders. The exact timing of each activity is recorded by the computer [19].

In real-world AR applications, unlabeled sensor data are relatively easy to obtain, whereas labels are expensive and tedious to collect. Learning an effective AR model according to a small subset of labeled data with large amounts of unlabeled data has attracts more and more researchers' attention. Two evaluated semi-supervised techniques, self-training and co-training, are employed to learn activity models from limited amount of labeled training data [20].A semi-supervised traffic classifier is obtained with a small number of labeled flows mixed with a large number of unlabeled flows using statistics to classify the traffics [9]. In [21], authors present a new and efficient semi-supervised training method for parameter estimation and feature selection in conditional random fields which can simultaneously select discriminative features via modified Logit Boost and utilize unlabeled data via minimum entropy regularization.

C. Universality: generic/personalized model

Personalization of activity recognition has become a topic of interest recently. Increasing works emerge in AR literature focusing on the personalization problem [8][16][22][23]. To solve the problem that a model cannot accurately recognize activities of a specific user, TransEMDT (Transfer learning EMbedded Decision Tree) is presented which integrates decision tree and k-means clustering algorithm for personalized activity-recognition model adaptation [22]. Authors in [23] present "uWave" application for interaction based on personalized gestures and physical manipulations of a consumer electronic or mobile device. Unlike statistical methods, "uWave" requires a single training sample for each gesture pattern and allows users to employ personalized gestures and physical manipulations [23].

Specifically, Support vector machine (SVM) is a popular supervised learning model represented as a hyperplane which separates the data from two classes with a maximal margin [24].Extreme learning machine (ELM) is fast single hidden layer feedforward neural network for classification and regression with the weights between input layer and hidden layer are randomly assigned [25]. Semi-supervised and incremental versions [26][27][28][29] of those traditional models have been proposed to deal with problems of unlabeled data for training or coming one by one or chunk by chunk. Some

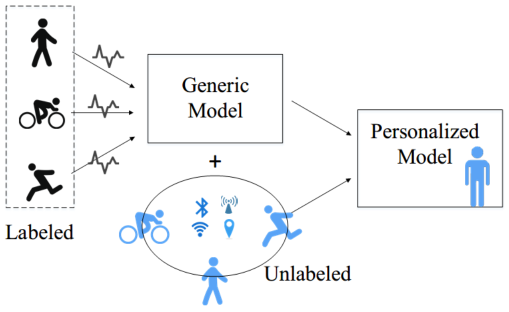


Fig. 1. Self-supervised ARTMAP

representative supervised, semi-supervised and incremental models above are compared with our model in Section V.

III. Self-supervised learning framework

A. Framework

A self-supervised incremental learning framework is proposed which consists of two stages to construct a user-oriented AR model.See Fig. 1.

- In the first stage, called "Model Initialization Stage", a **generic model** is learned based on the labeled motion data from different activities of multiple people.
- In the second stage, called "Model Update Stage", the generic model is growing to be a **personalized model** based on the consistently incremental unlabeled data of a target user. Those data for model update contain characterizations of both motion and context.

The generic model is in charge of grasping the crucial locomotion knowledge of activities from the motion data of people. Making the generic model as a starting point, we are trying to construct a user-oriented model by gathering the individual personalities of our target-user as well as the contextual information during his ADLs.

B. Activity Recognition Model

Adaptive Resonance Theory (ART) is a cognitive and neural theory solving the problem of how a machine can learn knowledge quickly from the new data without forgetting previously learned knowledge. Early ART neural networks (e.g. Fuzzy ART [30], Gaussian ART [31]) are proposed for unsupervised incremental learning, they incrementally learn a set of templates called categories. See Fig. 2

ARTMAP [32], also known as Predictive ART, is the neural network architecture for supervised incremental learning problem. Each input activates a single category node during both training and testing. When a node is first activated during training, it is mapped to the class the input belongs to. If the input activates a node in testing, the corresponding class mapped to this node is regarded as the prediction.

For those ART and ARTMAP networks, the input has a fixed number of features throughout training and testing. Self-supervised ARTMAP [10], learn partial knowledge from labeled inputs in the first stage, then enrich itself according

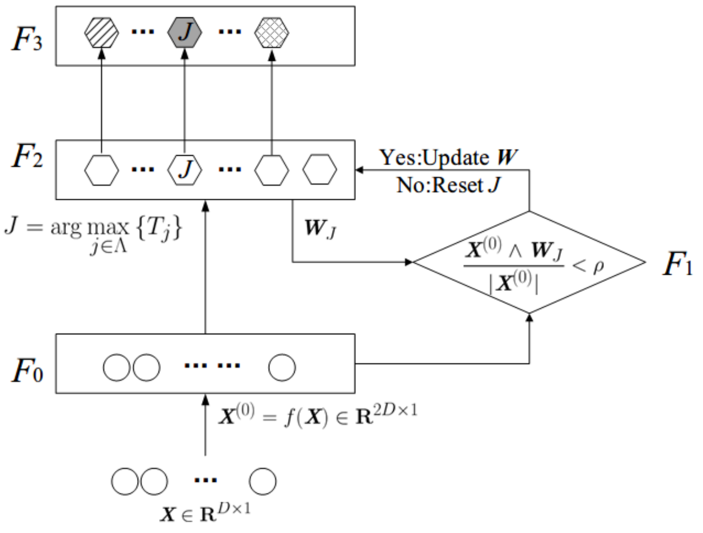

F_3
J
F_2
J
Yes:Update $\boldsymbol{W}$
No:Reset J
$J = \arg\max_{j\in\Lambda}\{T_j\}$
$\boldsymbol{W}_J$
$\frac{|\boldsymbol{X}^{(0)} \wedge \boldsymbol{W}_J|}{|\boldsymbol{X}^{(0)}|} < \rho$
F_1
F_0
$\boldsymbol{X}^{(0)} = f(\boldsymbol{X}) \in \mathbf{R}^{2D\times 1}$
$\boldsymbol{X} \in \mathbf{R}^{D\times 1}$

Fig. 2. Five steps for data collection and preprocess

to unlabeled inputs containing additional novel features in the second stage.

In the following section, we briefly elaborate the learning process of the Self-Supervised ARTMAP. Let the inputs for training be a set of feature vectors of equal length $\boldsymbol{X}_i \in \mathbf{R}^D, i = 1, \cdots, N$. Since the unlabeled inputs in the second stage contains more features than the labeled ones in the first stage. Let the total number of features contained by the unlabeled inputs be D, and the number of features contained by only the labeled inputs be $d(d < D)$. The values of features $d+1, \cdots, D$ within those labeled inputs are set to "1"s.

$$\boldsymbol{X}_i = \begin{cases} (X_{i1}, \cdots, X_{id}, 1, \cdots, 1)^T, i = 1, \cdots, n \\ (X_{i1}, \cdots, X_{id}, X_{id+1}, \cdots, X_{iD})^T, i = n+1, \cdots, N \end{cases} \tag{1}$$

and

$$Y_i \in \{1, 2, \cdots, C\}, i = 1, \cdots, n$$

In F_0 field, each input $\boldsymbol{X}_i$ is encoded into the vector $\boldsymbol{X}_i^{(0)} \in \mathbf{R}^{2N\times 1}$ using a preprocessing step called "Complement Coding".

$$\boldsymbol{X}_i^{(0)} = f(\boldsymbol{X}_i) = \begin{cases} (X_{i1}, \cdots, X_{id}, 1, \cdots, 1, \\ 1 - X_{i1}, \cdots, 1 - X_{id}, 1, \cdots, 1)^T, i = 1, \cdots, n \\ (X_{i1}, \cdots, X_{iD}, 1 - X_{i1}, \cdots, \\ 1 - X_{iD})^T, i = n+1, \cdots, N \end{cases} \tag{2}$$

This step allows the system to distinguish between the consistently present features and those sometimes absent and sometimes present. The complement coded input activates a "committed" node J in F_2 field according to choice function T_j in the winner-take-all mechanism.

$$J = \arg\max_{j\in\Lambda}\{T_j\} \tag{3}$$

$$\Lambda = \{j : T_j > \alpha D\} \subseteq \{1, \cdots, C\} \tag{4}$$

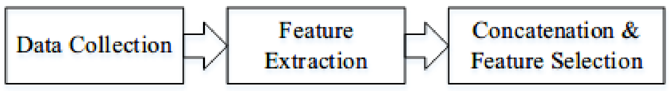


Fig. 3. Five steps for data collection and preprocess

$$T_j = \frac{(2D - |\boldsymbol{X}_i^{(0)}|) - (|\boldsymbol{W}_j| - |\boldsymbol{X}_i^{(0)} \wedge \boldsymbol{W}_j|)}{1 - \gamma\Phi_j} - \alpha(D - |\boldsymbol{W}_j|), \quad j = 1, \cdots, C \tag{5}$$

In equation (5), α is the signal rule parameter ranging from $(0, 1)$, and Φ_j represents the initial degree of commitment of the node j. A node becomes "committed" in F_3 field, the first time it is activated by a complement coded input $\boldsymbol{X}^{(0)}$. Here a committed node means the node is permanently linked to the label Y of the input $\boldsymbol{X}^{(0)}$, and a weight vector $\boldsymbol{W}$ corresponding to the node is generated to converge to the complement coded input $\boldsymbol{X}^{(0)}$.

In F_1 field, the bottom-up input $\boldsymbol{X}_i^{(0)}$ is matched against $\boldsymbol{W}_J$ that is read out by the activated node J. The matching criterion is set by a parameter ρ called vigilance.If node J agrees to the matching criterion and predicts the correct class, the weight $\boldsymbol{W}_J$ is updated in a way that this node can still success if other similar inputs come; otherwise, another node is to be activated.

In this paper, a self-supervised ARTMAP activity recognition model (SS-ARTMAP-AR) is learned for each user within the Self-supervised learning framework.

IV. Data collection and preprocessing

In order to validate above framework, data are collected and preprocessed through 3 steps (Fig. 3). In the following paper, we use the word "sample" to represent the raw sensor reading and "instance" to describe the data generated for training a classifier.

A. Data Collection

We collect data from three users, 2 postgraduate students (User "A" and User "B") and 1 office worker (User "C") during their daily life for more than 5 months in total. Each user is provided with a Shimmer 2r device [33] containing a tri-axial accelerometer and a Bluetooth module to communicate with an android smartphone paired with the Shimmer(Fig. 4). The Shimmer is worn in user's non-dominant (left) wrist, where the acceleration samples are collected and transmitted wirelessly to the phone according to an Android application "Shimmer Data" (Fig. 4), which displays the data as well as saves them in the phone's extra storage card.

Besides "Shimmer Data", another two applications "Sensor List" and "Activity Recorder" also run on the phone (Fig. 4). "Sensor List" is in charge of collecting the context data in surroundings, including the scanned Bluetooth devices, nearby Wi-Fi Access Points (APs), GPS readings if any, Global System for Mobile communication (GSM).The attributes of

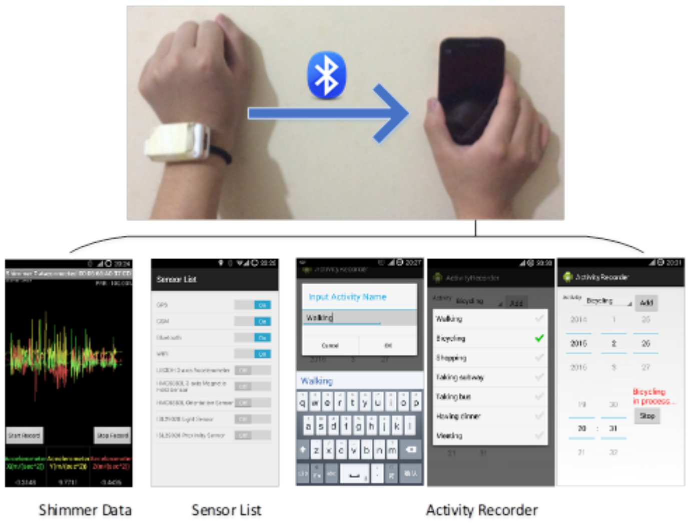


Fig. 4. Devices and applications for data collection and user annotation

TABLE I
Description of Sample attributes of each sensor

Sensor	Sample Attributes
Accelerometer	Timestamp, X-axis value, Y-axis value, Z-axis value
Bluetooth	Timestamp, DeviceName, Address, BondState, BlueToothClass, DeviceClass, MajorDeviceClass, ScanID
Wi-Fi	Timestamp, ssid, bssid, Capabilities, Level, Frequency, ScanID
GPS	Timestamp, Longitude, Latitude
GSM	Timestamp, cid, lac, mcc, mnc, bsss

those samples are listed in Table I. Thorough descriptions of the attributes can be found in [34][35][36][37].

During the data collection process, users are required to annotate the ground truth (activity) with the application "Activity Recorder" (Fig. 4). User inputs each activity name and saves it in the left drop-down menu using the "Add" button. Before user begins an activity (e.g. bicycling), he chooses the corresponding item and presses the "Start" button. The status is changed to "in process", meanwhile a message about the activity and the begin time is saved into a text file. Similarly, when the user stops bicycling, he presses the "Stop" button, at the same time a message about the activity and its terminal time is also saved into the file. Therefore, we get the ground truth $\langle activityType, beginTime, endTime \rangle$ for each activity collected.

The sampling rate of the accelerometer is 102.4Hz and the value ranges from $-6g$ to $+6g$. The scanning process of Bluetooth devices involves an inquiry scan of about 12 seconds, followed by a page scan. Each scanned Bluetooth device is returned in the form of a Broadcast service[34]. Due to the lengthy time cost in starting and scanning procedures of Bluetooth and Wi-Fi, it collects Bluetooth and Wi-Fi data every 30 seconds. One GPS sample is collected per minute owing to the long signal propagation distance between the satellite and the user.Besides, we collect GSM data every 5

TABLE II
ACTIVITY FREQUENCY ANNOTATED BY 3 USERS

ID	Activity	Time Duration(Hour)		
		A	B	C
1	Walking	8.932	19.246	3.467
2	Bicycling	24.486	7.738	23.654
3	Climbing stairs	4.285	1.107	1.339
4	Taking elevator	3.630	3.150	1.435
5	Taking the subway	1.310	3.542	0.991
6	Having a meal	15.486	16.175	7.767
7	Taking a nap	16.167	12.843	/
8	Meeting	24.770	3.424	/
9	Shopping	0.773	/	0.461
10	Watching movie	1.760	/	5.745
11	Taking the bus	/	0.619	15.801
12	Riding on the back of a bicycle	1.934	/	/
13	Watching table tennis game	2.624 /	/	
14	Taking a taxi	/	1.919	/
15	Fitness training	/	0.760	/
16	Having fun in KTV	/	1.478	/

TABLE III
FEATURES EXTRACTED FOR THE DATA OF 5 SENSORS

Sensors	Extracted Features	Feature #
Accelerometer	Mean, Standard Deviation (STD), Minimum, Maximum, Energy, Mean Crossing Rate, Mean_Shape, STD_Shape, Skew_Shape, Kurt_Shape, Mean_Amplitude, STD_Amplitude, Skew_Amplitude, Kurt_Amplitude	42
Bluetooth	Device #, ChangeRate_BTAddress, 3 bonding state, 12 "Major Device Classes", 83 "Minor Device Classes"	100
Wi-Fi	AP #, Mean_level, ChangeRate_APAddress, ChangeRate_capability, ChangeRate_frequency	5
GPS	Longitude, Latitude	2
GSM	ChangeRate_GSM, bsss	2

seconds.

For accelerometer and GSM, the number of samples is determined by the sample rate and the duration of data collection process. Whereas things are different for Bluetooth, Wi-Fi, and GPS, since the number of samples collected for them depends on whether there is signal (for GPS), as well as how many Bluetooth devices/APs can be scanned. Sometimes no Bluetooth, Wi-Fi or GPS data is collected because no such signals exist in surroundings. For instance, when a user is shopping indoor in a supermarket, there is no GPS data that can be collected.

The Shimmer and the phone are taken along in the morning before user going out and took off in the late evening after back home. All the data collected are manually transmitted to a SQLite database in a laptop. The Shimmer and the phone are charged at night for the next day's usage. Due to the memory and battery restrictions of the phone, users cannot collect the data all day long. Therefore, we ask them to stop data collection and close all the applications when they are working/studying in the office/lab which may take more than half of daytime. Table II lists all the activities annotated by users and the time duration of each activity. 12 distinct activities are collected by user "A" and "B", and 9 activities by user "C". Basically, data of each activity are collected for more than 1 hour. In total, we get 238 hours of collected data.

B. Feature Extraction

1) Accelerometer data

For a window of acceleration samples (data of 5 seconds) in each axis ($X/Y/Z$-axis), we extract fourteen features from time and frequency domains in a window of samples by sliding window mechanism (window size=500, step size=100). Together we get forty-two features (Table III) from three axes to generate an acceleration instance.

2) Bluetooth data

a) For each scanned Bluetooth sample, the documentation for Android SDK has defined its major class from 12 "Major Device Classes", as well as its minor class from several "Minor Device Classes" in the context of the major class assignment [34]. For instance, a Samsung Galaxy S5 phone belongs to the "Smartphone" minor class in the "Phone" major class. An "Acer E1-471G-53214G50Mnks" laptop belongs to the "Laptop" minor class in the "Computer" major class. Besides, we can also check the on-going bonding state (None, Bonding, Bonded) of the scanned Bluetooth sample [34]. In all, we extract 98 features with Boolean value (0/1) consisting of 3 bonding states, 12 "Major Device Classes" and 83 "Minor Device Classes" for each sample.

b) Several Bluetooth samples may be collected during one-time Bluetooth scan. Our feature extraction process for Bluetooth data is based on all the samples in same scan. We count the number of samples ("Device #") as the first extracted feature. In comparison with the last scan, the percent of samples with new Bluetooth address ("ChangeRate_BTAddress") is defined as the second feature. For each of the 98 features above, we count the percent of samples in the scan whose value is 1 as the feature value. Therefore, a Bluetooth instance contains 100 features (Table III).

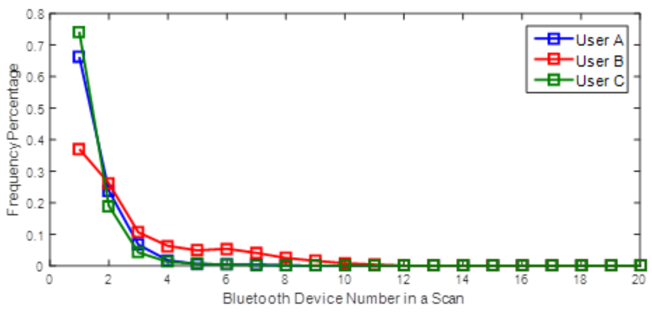


Fig. 5. Devices and applications for data collection and user annotation

Fig. 5 shows the number of Bluetooth devices found in a scan and its corresponding percentage. For the datasets of all the users, there are only less than 3 Bluetooth devices in surroundings most of the times. It truthfully represents the environments of users' ADLs since we did not intentionally add any devices with their Bluetooth opening for the data collection.

3) Wi-Fi data

For the Wi-Fi samples in one scan, we count the number of samples with distinct AP address as the first feature ("AP #"), and calculate the average signal strength as the second feature ("Mean_level"). The percent of samples with new AP address ("ChangeRate_APAddress"), capability ("ChangeRate_capability"), and frequency ("ChangeRate_frequency") are also collected as the features. In all, a Wi-Fi instance owns 5 features (Table III).

4) GPS and GSM data

Each GPS sample generates a GPS instance with longitude and latitude as its features (Table III). Each GSM sample generates a GSM instance with 2 features. One feature is the signal strength "bsss", the other feature "ChangeRate_GSM" is a Boolean value representing whether "cid" or "lac" is changed or not compared with the last GSM sample.

C. Concatenation and Feature Selection

For all the instances of accelerometer/ Bluetooth/ Wi-Fi/ GPS/ GSM sensors, we concatenate their features according to the timestamps of instances to create new instances.

In order to remove redundant features from the feature set, we utilize the supervised attribute filter "Normalize Filter" in Weka [38] to select the optimal feature subset for each user.

V. Results

A. Accuracy

The performance of SS-ARTMAP-AR is validated on each user's dataset. For simplicity, we randomly select 3 labeled instances with acceleration features, 500 unlabeled instances with acceleration, Bluetooth, Wi-Fi, GPS, GSM features selected, from each class of the dataset. The labeled/unlabeled instances are the training data in Model Initialization Stage / Model Update Stage of SS-ARTMAP-AR. Another 300 instances from each class constitute the testing set. The prediction accuracies of 3 generated SS-ARTMAP-AR models on each corresponding testing set are listed in Table IV.

Generally speaking, the accuracy varies on different users and classes. For 4 typical daily activities ("Walking", "Bicycling", "Climbing stairs", "Taking elevator"), accuracy exceeds 50% for all the 3 users, which means the SS-ARTMAP-AR model has robust recognition ability on the ADLs. Besides, SS-ARTMAP-AR model has an accuracy of more than 50% on 3 high-level semantic activities ("Taking a nap", "Shopping", "Watching a movie") of 2 users, which is amazing since those activities are quite similar in nature but different in environmental circumstances. For instance, an activity of "Shopping" is mixed with some "Walking" and special gestures of picking up/put down some products. An accuracy of 99% is achieved on "Riding on the back of a bicycle" class of user "A", which is even better than the Bicycling class. However, SS-ARTMAP-AR also shows poor predictive ability on 3 classes ("Meeting", "Watching table tennis game", "Having fun in KTV") owning to their ambiguous characteristics on existing dataset.

TABLE IV
Testing accuracy of SS-ARTMAP-AR

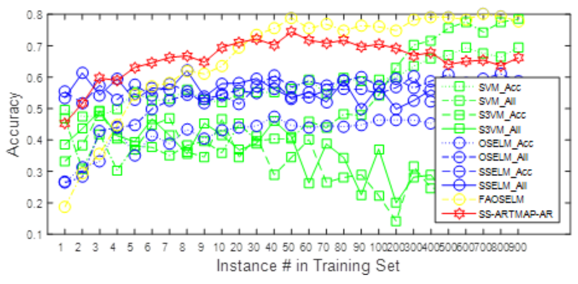
ID	Activity	Accuracy		
		A	B	C
1	Walking	0.73	0.57	0.78
2	Bicycling	0.97	0.94	0.99
3	Climbing stairs	0.83	0.83	0.76
4	Taking elevator	0.72	0.65	0.53
5	Taking the subway	0.51	0.45	0.62
6	Having a meal	0.81	0.29	0.63
7	Taking a nap	0.87	0.85	/
8	Meeting	0.37	0.25	/
9	Shopping	0.78	/	0.70
10	Watching movie	0.90	/	0.67
11	Taking the bus	/	0.39	0.56
12	Riding on the back of a bicycle	0.99	/	/
13	Watching table tennis game	0.29	/	/
14	Taking a taxi	/	0.55	/
15	Fitness training	/	0.62	/
16	Having fun in KTV	/	0.43	/

B. Comparison with other approaches

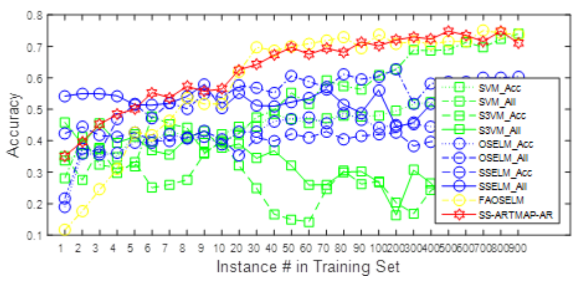
In this section, we evaluate the performance of SS-ARTMAP-AR model and compare it with several representative supervised and semi-supervised methods (Fig. 6). Comparison methods are divided into two groups:"SVM group" and "ELM group". Four methods are contained in "SVM group" and five in "ELM group". Among them, "SVM_*" [24] and "OSELM_*" [28] are supervised incremental learning methods; "S3VM_*" [26] and "SSELM_*" [27] are semi-supervised incremental learning methods. "_Acc" and "_All" represents instances for training and testing the model contains only acceleration features and all kinds of features (features from acceleration, Bluetooth, WiFi, GPS and GSM) respectively.

For all the supervised incremental methods introduced above,the feature sets of initial model generation and model update are the same("_Acc" or "_All"). Differently, "FAOSELM" [29] is a specialized supervised incremental method in which an initial model is trained based on acceleration features, and updated based on data with all features (acceleration, Bluetooth, Wi-Fi, GPS and GSM). "FAOSELM" is very similar to SS-ARTMAP-AR model except that data for model update are with labels for the former, and no labels is needed for the latter. To compare the results of four methods (Fig. 6) in "SVM group","ELM group" on 3 users' data we can conclude that: First, the performance of supervised methods are better than semi-supervised methods, which means user annotations are beneficial in constructing efficient AR models. Second, "_All" methods attain higher accuracy than "_Acc" methods, which illustrates the significance of context features in classifying ADLs. Third, the performance of methods in "SVM group" is much clearer and more distinguishable than "ELM group".

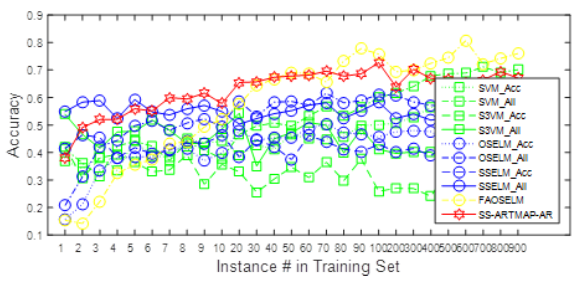
Among all the methods in comparison, SS-ARTMAP-AR, "FAOSELM" and "SVM_All" gain highest predictive ability on the dataset of User "B" (Fig. 6(b)), which is significant achievement for SS-ARTMAP-AR since large amounts of annotations are avoided. For datasets of User "A" and "C",



(a) User A



(b) User B



(c) User C

Fig. 6. Performance comparison of SS-ARTMAP-AR and supervised, semi-supervised methods in “SVM group” and “ELM group”

“FAOSELM” and “SVM_All” are a little bit superior to SS-ARTMAP-AR (Fig. 6(a),(c)). Still, these three methods are much better than the rest.

C. Contribution of environmental features

SS-ARTMAP-AR model has shown powerful recognition ability in predicting the ADLs according to the results in “Accuracy”. In this section, we want to evaluate the contribution of environmental features (Bluetooth, Wi-Fi, GPS, GSM features) to the overall accuracy of SS-ARTMAP-AR model. As mentioned above, the conventional SS-ARTMAP-AR model (Self-supervised ARTMAP) utilizes labeled instances with selected acceleration features as the training data in Model Initialization Stage, and unlabeled instances with selected features of all five sensors as the training data in Model Update Stage. For the sake of comparison, we also train an SS-ARTMAP-AR model with labeled instances in Model Initialization Stage, and unlabeled instances in Model Update Stage, both of which contain acceleration features only. We call it “Semi-supervised ARTMAP”, since it is generated

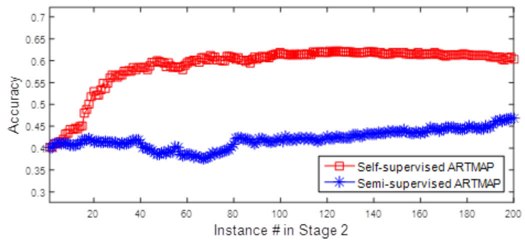


Fig. 7. Performance comparison of self-supervised ARTMAP-AR and semi-supervised ARTMAP-AR

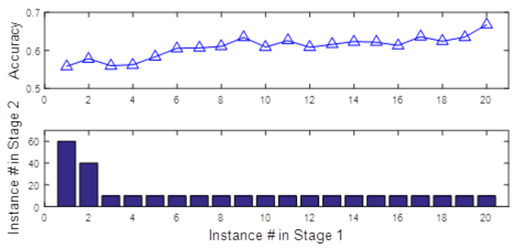


Fig. 8. Performance comparison of SS-ARTMAP-AR with different number of labeled instances in Model Initialization Stage

in a Semi-supervised learning framework. Given the same set of labeled instances in Model Initialization Stage, their predictive accuracy on the same testing set are compared. It should be noted that numbers in X-axis of Fig. 7 represents the amount of instances in each class, which is the same in Fig. 8. The accuracy increases for both. Obviously, the accuracy of Self-supervised ARTMAP goes faster and more stable than Semi-supervised ARTMAP model owing to the environmental Bluetooth, Wi-Fi, GPS, GSM features. Therefore, we can conclude that: environmental features, which contains a great deal of personalized characteristics to the data of on-going activities, can increase classification performance of an activity recognition model.

D. Effect of number of labeled instance

It is necessary to analyze the effect of the number of labeled instances on the recognition ability of SS-ARTMAP-AR, which is really important in practical application since the label of instances comes from costly and intrusive user annotation. For the whole dataset of User “A”, we randomly select 100 instances from each class to construct the testing set. The number of labeled instances for training from each class varies from 1 to 20. For each labeled instance number, we record the least number of unlabeled instances (see the histogram in Fig. 8) it needs in Model Update Stage to achieve an testing accuracy of more than 50% on the testing set. We look for the “least number” from 10, 20, 30,... until the proper one is found. The actual accuracy of SS-ARTMAP-AR model learned on the basis of different size of labeled instances are shown in the upper part of Fig. 8. From the results we can conclude: if there are only 2 labeled instances (data of more than 15 seconds) from each class are in Model Initialization Stage, it needs at least 10 unlabeled instances in Model Update

Stage to achieve an accuracy of large than 50%. It shows the fact that SS-ARTMAP-AR does not rely on massive user annotations. Performance of SS-ARTMAP-AR does benefit from more labeled data, but only a few labeled data, e.g. with 15 seconds user annotations for each activity ("activity-level one-off annotations"), are still enough for high recognition performance.

VI. Conclusion

In this paper, we construct SS-ARTMAP-AR, a self-supervised incremental learning model to recognize ADLs of users with mini-wearable devices. SS-ARTMAP-AR is firstly initialized using a few motion data with relevant annotations, then updated incrementally by the motion and context data without any user annotation effort. According to the consistent discovery of regular patterns in daily life, SS-ARTMAP-AR gradually adapts the user and make the device more intelligent.

In the future, we plan to actively select informative instances in personalizing the generic activity recognition models for mini-wearable device users.

Acknowledgment

This work is supported in part by Natural Science Foundation of China (NSFC) under Grant No.61572471, CAS Research Equipment Development Project under Grant No.YZ201527, Science and Technology Planning Project of Guangdong Province, China, under Grant No.2015B010105001, and Innovation Project of Institute of Computing Technology under Grant No.20156010.

References

[1] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing*. Springer, 2004, pp. 1–17.

[2] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE, 2014, pp. 51–59.

[3] https://jawbone.com/up.

[4] https://getpebble.com/steel.

[5] http://www.bodymedia.com/Support-Help/BodyMedia-FIT-BW.

[6] F. Moiz, P. Natoo, R. Derakhshani, and W. D. Leon-Salas, "A comparative study of classification methods for gesture recognition using a 3-axis accelerometer," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 2479–2486.

[7] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.

[8] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Networks*, vol. 53, pp. 1–7, 2014.

[9] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Semi-supervised network traffic classification," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1. ACM, 2007, pp. 369–370.

[10] G. P. Amis and G. A. Carpenter, "Self-supervised artmap," *Neural Networks*, vol. 23, no. 2, pp. 265–282, 2010.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[12] D. H. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, 2011, p. 1962.

[13] Y. Chen, Z. Chen, J. Liu, D. H. Hu, and Q. Yang, "Surrounding context and episode awareness using dynamic bluetooth data," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 629–630.

[14] Z. Chen, Y. Chen, S. Wang, J. Liu, X. Gao, and A. T. Campbell, "Inferring social contextual behavior from bluetooth traces," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 267–270.

[15] G. S. Chambers, S. Venkatesh, G. A. West, and H. H. Bui, "Hierarchical recognition of intentional human gestures for sports video annotation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2. IEEE, 2002, pp. 1082–1085.

[16] M. Fetter and T. Gross, "Lilolea framework for lifelong learning from sensor data streams for predictive user modelling," in *Human-Centered Software Engineering*. Springer, 2014, pp. 126–143.

[17] J. W. Lee and C. Giraud-Carrier, "Transfer learning in decision trees," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*. IEEE, 2007, pp. 726–731.

[18] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 10–19.

[19] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1553–1567, 2006.

[20] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *Wearable computers, 2008. ISWC 2008. 12th IEEE international symposium on*. IEEE, 2008, pp. 81–88.

[21] M. Mahdaviani and T. Choudhury, "Fast and scalable training of semi-supervised crfs with application to activity recognition," in *Advances in Neural Information Processing Systems*, 2008, pp. 977–984.

[22] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *IJCAI*, vol. 11. Citeseer, 2011, pp. 2545–250.

[23] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[25] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 879–892, 2006.

[26] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, vol. 99, 1999, pp. 200–209.

[27] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *Cybernetics, IEEE Transactions on*, vol. 44, no. 12, pp. 2405–2417, 2014.

[28] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1411–1423, 2006.

[29] X. Jiang, J. Liu, Y. Chen, D. Liu, Y. Gu, and Z. Chen, "Feature adaptive online sequential extreme learning machine for lifelong indoor localization," *Neural Computing and Applications*, vol. 27, no. 1, pp. 215–225, 2016.

[30] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural networks*, vol. 4, no. 6, pp. 759–771, 1991.

[31] J. R. Williamson, "Gaussian artmap: A neural network for fast incremental learning of noisy multidimensional maps," *Neural networks*, vol. 9, no. 5, pp. 881–897, 1996.

[32] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural networks*, vol. 4, no. 5, pp. 565–588, 1991.

[33] http://www.shimmer-research.com/.

[34] http://developer.android.com/guide/topics/connectivity/bluetooth.html.

[35] http://developer.android.com/reference/android/net/wifi/ScanResult.html.

[36] http://developer.android.com/guide/topics/location/strategies.html.

[37] http://developer.android.com/reference/android/telephony/gsm/GsmCellLocation.html.

[38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.