Personalized Smartphone Wearing Behavior Analysis

Yen-Hsuan Lin and Yi-Ta Chuang(✉)

Department of Computer Science, National Chiao-Tung University,
1001 University Road, Hsinchu City 30010, Taiwan
izeki.cs96g@nctu.edu.tw

Abstract. Next generation smartphones have the ability to sense user contexts such as mobility, device wearing position, location, activity, emotion, health condition. Many apps utilize user contexts to provide innovative services, e.g., pedometer, advanced navigation and location based services. Two of the most important user contexts are mobility patterns (still and walk) and device wearing positions (hand, arm, chest, waist and thigh). We call these two user contexts "wearing behavior". In this paper, we propose a 3-stage framework to recognize smartphone wearing behaviors by utilizing sensor data from smartphones. The framework starts with data preprocessing to extract sensor features and generate ground truths. After the data preprocessing, a threshold based finite state machine utilizes the sensor features to determine whether the smartphone is attached or not. Finally, a decision tree model is built based on the ground truth to determine the wearing behaviors. The experiment results show that our approach can achieve 94 % accuracy in average.

Keywords: Mobility · Next generation smartphone · User context · Wearing position · Wearing behavior

1 Introduction

Next generation smartphones have the ability to sense user contexts such as mobility, device wearing position, location, activity, emotion, health condition. Many apps utilize user contexts to provide innovative services, e.g., pedometer, advanced navigation and location based services. Two of the most important user contexts are mobility patterns and device wearing positions. For example, an adaptively models could be developed for the pedometer to detect footsteps more accurately; a better positioning method could be established for advanced navigation and location based services. In addition, an adaptive sensor duty cycle policy could be designed based on mobility patterns and wearing positions to support continuously user context sensing. In this work, mobility patterns and wearing positions are called wearing behaviors, and we want to use various sensors embedded in smartphones to detect the wearing behaviors.

There are many embedded sensors in a smartphone, e.g., accelerometer, gyroscope and light sensors. Recently, there are many works using the embedded sensors in a smartphone to detect user contexts [1] and utilize the user context to provide innovative services [2–4]. In [5] and [6], the applications utilize acceleration to detect human

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 318–328, 2014.
DOI: 10.1007/978-3-319-13186-3_30

mobility by the embedded accelerometer in the smartphone. However, these studies only consider the devices wearing in a fixed body position. This constraint restricts the applicability of their methods in the smartphone application since the smartphone users may not place their phones in fixed positions. In this work, we propose to consider both mobility and wearing position as a new user context called the wearing behavior and propose a framework to detect this new user context so that these applications requiring wearing behavior information can be adopted in the smartphone applications.

In this work, the wearing behavior is defined as a set of ordered pairs of mobility pattern set and wearing position set. We consider human mobility such as walking and stationary states, and the smartphone wearing positions including hand, arm, chest, waist and thigh. We develop a 3-stage approach to detect the smartphone wearing behavior. A data collection tool was developed to help us collect sensor data and label wearing behaviors. In the first stage, these labeled data are preprocessed to extract features and generate ground truth. In the second stage, the extracted features are used to build a threshold based finite state machine (FSM) to detect whether a smartphone is attached or not. This stage is reasonable because it is meaningless to detect the wearing behavior if the smartphone is not worn in the body. In the final stage, the extracted features are used to train a decision tree (DT) model for wearing behavior recognition. If it is detected that a smartphone is attached, the decision tree model is used to determine the wearing behavior.

The contributions of this work are several folds. Firstly, we proposed a threshold based FSM by using embedded sensors such as gyroscopes and accelerometers to detect the attachment of smartphones. By this mean, it is possible to improve user experience. For example, if it is detected that the smartphone is not attached, it can automatically tune the volume up so that users may not miss a phone call. Secondly, we developed a framework for personalized wearing behavior recognition. Based on the relationship of sensor readings to wearing positions and mobility, a personalized model is established by data mining techniques for personalized smartphone wearing behavior recognition. With this wearing behavior information, it is very useful to assist in applications, such as pedometer, advanced navigation and location based services, which require personalized usage behaviors. Last but not the least; the extracted features may not be only useful in the wearing behavior analysis but also in other user context related studies.

The rest of this paper is organized as follows. Section 2 reviews some related works on user context applications. In Sect. 3, the overview of our approach is introduced. Section 4 presents the data preprocessing process that extracts features and ground truth. The attachment recognition process that use a threshold based FSM is introduced in Sect. 5. The smartphone wearing behavior recognition process that applied DT is described in Sect. 6. Conclusions are drawn in Sect. 7.

2 Related Works

In this section, we give a briefly review on the related works to the user context related applications and research works.

In [1], a mechanism is proposed to detect user context for mobile and social networking applications. In [2, 3], the authors utilized the user context in the app usage to design energy saving mechanisms. In [4], a fast app launching mechanism is proposed based on the user context. In [5], this application turns the phone screen off automatically when it detects the user puts the phone into a pocket or onto a table and turns the screen on automatically when it detects the user takes the phone out or up. In [6], this application uses the accelerometer to detect whether the phone is on the hand. If it detects the phone is on the hand, it keeps the screen on.

In this work, we consider both mobility and wearing position as a new user context called the wearing behavior and propose a framework to detect this new user context. The detected wearing behavior could be utilized in these applications requiring wearing behavior information so that the user can enjoy these innovative services in smartphones.

3 Proposed Approach

To analyze the wearing behaviors, we propose a two-phase framework. The phases are training phase and inference phase. Each phase is divided into three stages: data preprocessing, attachment recognition and wearing behavior recognition as illustrated in Fig. 1.

For the training phase, in the data preprocessing stage, the sensor data with wearing behavior labels are collected. In this stage, sensor features are extracted from sensor data and the sensor features are input to the attachment recognition stage. In addition, the sensor features with wearing behavior labels are the ground truths and the ground truths are input to the wearing behavior recognition stage. In the attachment recognition stage, the sensor features are used to determine the threshold in the FSM. In the wearing behavior recognition stage, the ground truths are used to build the DT model. We will introduce the sensors features used in our approach in Sect. 4.

For the inference phase, in the data preprocessing stage, sensor features are extracted from sensor data and the sensor features are input to the attachment recognition stage. In the attachment recognition stage, the threshold based FSM determines the attachment state based on the sensor features. The attachment state along with the sensor features are input to the wearing behavior recognition stage. In the wearing behavior recognition stage, if the attachment state is attached, the DT model detects the wearing behaviors. Otherwise, the wearing behavior recognition finishes. We will introduce the threshold based FSM for the attachment recognition and the DT for the wearing behavior recognition in our approach in Sects. 5 and 6, respectively.

4 Data Preprocessing

Recall that our approach comprises the training phase and the inference phase. In order to collect training data for the subsequent model training for the training phase. We develop an application to help us collect sensors data with wearing behavior labels. In this application, users can label the mobility and the wearing positions so that the

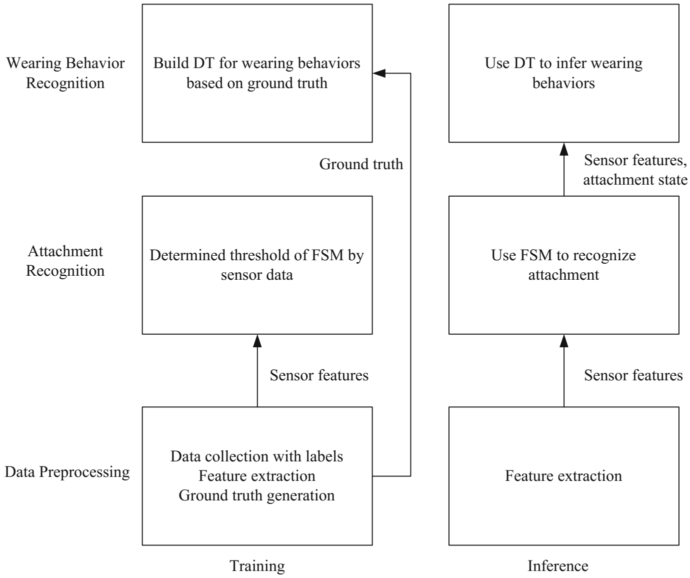

Fig. 1. Personalized wearing behavior analysis framework.

collected sensor data are with wearing behavior labels. The collected sensor data are preprocessed to extract sensor features. The sensor features are used to determine the threshold for the FSM in the attachment recognition stage. The sensor features with wearing behavior labels are the ground truths and are used to build DT in the wearing behavior recognition stage. For the inference phase, the sensor features are extracted in the data preprocessing stage. After that, the sensor features are input to the attachment recognition stage to determine the attachment state by the FSM. Finally, the sensor features and attachment state are input to the wearing behavior stage to detect the wearing behaviors by the DT model. In our experiment, we use hTC one model to collect sensor data and the sampling rate is 100 Hz.

The wearing behavior considered in this work is an order pair of the mobility patterns and the wearing positions. The mobility patterns in this work include still and walk states. We say the mobility is in still state if the footsteps are no more than two per second. Otherwise, we say the mobility is in walk state. The wearing positions in this work are hand, arm, chest, waist and thigh. The hand position is self-explained; the arm position is that the smartphone is strapped in an armband; the chest position is that the smartphone is placed in a breast pocket; the waist position is that the smartphone is placed in the purse near the waist; the thigh position is that the smartphone is placed in the pocket of trousers.

The sensors and features used in this work are depicted in Table 1. We collect sensors data from the embedded accelerometer, gyroscope and light sensor. The accelerometer and gyroscope are used to detect the movement and direction of the smartphone. The light sensor is used to detect the intensity of light. The intensity of light could be used to determine whether a smartphone is in a pocket. Since the raw sensor data are diverse across different kind of domains, we use several feature extraction technique such as statistic and frequency domain transformation on the sensors data.

Table 1. Sensors and features used in this work.

Hardware sensor	Feature	Unit
Accelerometer	Avg intensity	m/s^2
	Avg intensity of vertical component	m/s^2
	Avg abs intensity of vertical component	m/s^2
	Avg intensity of horizontal component	m/s^2
	Standard deviation	m/s^2
	DFT of vertical components	–
	Phone direction	–
Gyroscope	Avg intensity	rad/s
	Standard deviation	rad/s
Light	Avg intensity	lux

4.1 Sensor Features

This section introduces the sensor features used in our framework. There are 7 features extracted from the accelerometer, 2 features from the gyroscope and 1 feature from the light sensor.

The features extracted from the accelerometer are average intensity of acceleration, average intensity of vertical components of acceleration, average intensity of horizontal components of acceleration, standard deviation of acceleration, Discrete Fourier Transformation (DFT) of vertical components of acceleration and phone direction.

Average intensity of accelerometer ($\overline{\|A\|}$) is calculated by

$$\overline{\|A\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \|\boldsymbol{a}_i\|, \tag{1}$$

where $\boldsymbol{a_i}$ is the ith acceleration and n is the number of data in a minute. $\|\ \|$ denotes the norm operation on a vector.

Average intensity of vertical components of acceleration ($\overline{\|A^{\perp}\|}$) is calculated by

$$\overline{\|A^{\perp}\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \|\boldsymbol{a}_i^{\perp}\|, \tag{2}$$

where $\boldsymbol{G}_i$ is the gravity of the ith data, $\boldsymbol{a}_i^{\perp}$ is the vertical component of the ith acceleration which is calculated by $\frac{\boldsymbol{a}_i \cdot \boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}\frac{\boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}$ and $\cdot$ denotes the dot product operation on vectors.

Average absolute intensity of vertical components of acceleration ($\overline{|A^{\perp}|}$) is calculated by

$$\overline{|A|^{\perp}} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{\boldsymbol{a}_i \cdot \boldsymbol{G}_i}{\|\boldsymbol{G}_i\|}\right| \tag{3}$$

Average intensity of horizontal components of acceleration ($\overline{\|A^{=}\|}$) is calculated by

$$\overline{\|A^{=}\|} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left\|\boldsymbol{a}_i - \boldsymbol{a}_i^{\perp}\right\| \tag{4}$$

Standard deviation of acceleration (A_σ) is calculated by

$$A_\sigma = \left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\|\boldsymbol{a}_i\| - \overline{\|\boldsymbol{A}\|}\right)^2\right)^{1/2} \tag{5}$$

The magnitude of vertical components of acceleration after DFT at 1 Hz (A_1) is calculated by

$$A_1 = \left\|\sum\nolimits_{j=0}^{2n}\left\|\boldsymbol{a}_j^{\perp}\right\| * e^{-i2\pi j/n}\right\| \tag{6}$$

Phone direction (P) is determined by

$$\mathrm{P} = \left\{\begin{array}{l} 0,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_x \\ 1,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_y \\ 2,\ \text{if}\ \max\{\bar{\boldsymbol{A}}_x, \bar{\boldsymbol{A}}_y, \bar{\boldsymbol{A}}_z\} == \bar{\boldsymbol{A}}_z \end{array}\right\}, \tag{7}$$

where $\bar{\boldsymbol{A}}_x$, $\bar{\boldsymbol{A}}_y$ and $\bar{\boldsymbol{A}}_z$ denote the average intensity of acceleration in x, y and z directions, respectively.

Average intensity of gyroscope ($\bar{G}$) is calculated by

$$\bar{G} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\|g_i\|, \tag{8}$$

where g_i is the ith angular velocity.

Standard deviation of gyroscope (G_σ) is calculated by

$$G_\sigma = \left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\|\boldsymbol{g}_i\| - \|\bar{\boldsymbol{G}}\|\right)^2\right)^{1/2} \tag{9}$$

Average intensity of light ($\bar{\boldsymbol{L}}$) is calculated by

$$\bar{L} = \log\left(\frac{1}{n}\sum\nolimits_{i=1}^{n}\|L_i\|\right), \tag{10}$$

where L_i is the ith light luminance.

5 Attachment Recognition

In the attachment recognition stage, for the training phase, the sensors features are used to determine the threshold of the FSM. For the inference phase, the threshold based FSM is designed to determine whether the smartphone is attached.

Since the accelerometer and gyroscope are able to measure the human mobility, it is possible to use the long-term variations of acceleration and rotation to determine whether the smartphone is attached. We use A_{th} and G_{th} as the acceleration and rotation thresholds, respectively. A_{th} is determined by

$$A_{th} = \max_{1<i<n} A_{\sigma i}, \tag{11}$$

where $A_{\sigma i}$ is the ith A_{σ} in the unattached data and n is the number of unattached data. G_{th} is determined by

$$G_{th} = \max_{1<i<n} G_{\sigma i}, \tag{12}$$

where $G_{\sigma i}$ is the ith G_{σ} in the unattached data.

Intuitively, we use the maximum of A_{σ} and the maximum of G_{σ} from the training data labeled with unattached state as the thresholds. Based on the thresholds, the attachment state can be determined by

$$\text{State} = \left\{ \begin{matrix} \text{unattached}, & \text{if } A_{\sigma} < T_A \textit{ and } G_{\sigma} < T_G \\ \text{attached}, & \text{otherwise.} \end{matrix} \right\} \tag{13}$$

However, this method may cause false detection. To resolve the false detection and improve the detection accuracy, a two-bit FSM is designed.

There are four states in the FSM: A/A, U/A, A/U and U/U. Each state uses two bits to record the previous and current attachment state. The character 'A' means "attached" and the character 'U' means "unattached". For example, U/A state means the previous attachment state is unattached and the current attachment state is attached. The state transition moves as follows. On the A/A state, when the attachment state is unattached, the state transits to the A/U state; when the attachment state is attached, no state transition occurs. On the U/A state, when the attachment state is unattached, the state transits to the A/U state; when the attachment state is attached, the state transits to the A/A state and the FSM reports the smartphone is attached. On the A/U state, when the attachment state is unattached, the state transits to the U/U state and the FSM reports the smartphone is unattached; when the attachment state is attached, the state transits to the U/A state. On the U/U state, when the attachment state is unattached, no state transition occurs; when the attachment state is attached, the state transits to the U/A state. Figure 2 illustrates the state transition of the FSM.

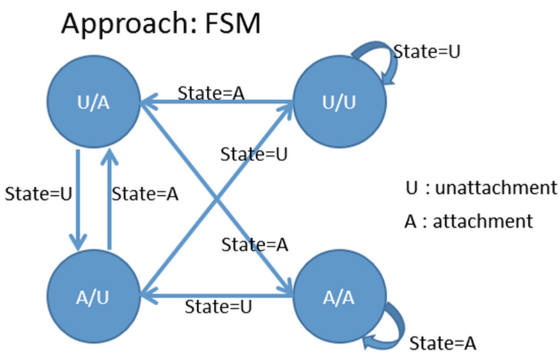

Fig. 2. The state transition diagram of the finite state machine.

6 Wearing Behavior Recognition

The final stage in our framework is the wearing behavior recognition stage. For the training phase, the ground truths generated in the data preprocessing stage are used to train the DT model. We use the REPTree algorithm [7] to build the DT model by the Weka data mining tool [8]. For the inference phase, the feature sets and attachment states from the attachment recognition stage are input to the DT model, and the wearing behavior is determined by the DT model. Figure 3 shows the process concept in the wearing behavior recognition stage.

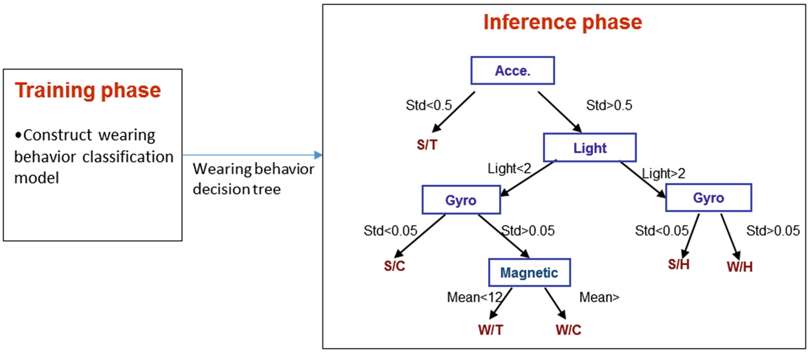

Fig. 3. The process concept in the wearing behavior recognition stage.

In this work, the REPTree algorithm is used to build the DT model for wearing behavior classification. The REPTree algorithm utilizes information gain/variance to build a decision/regression tree and prunes the tree by using reduced-error pruning (with backfitting). The REPTree algorithm uses the information gain [9] to create nodes in the decision tree. Let T denote the set of training data. The training data, i.e., the ground truths in this work, is in the form $X = (x_1, x_2, x_3, \ldots, x_k, y)$ where $x_i = val(a_i)$ is the value of the ith attribute and y is the wearing behavior label (class label). The entropy of the training data set T is calculated by

$$H(T) = -\sum_{i=1}^{n} P(c_i) log_2(P(c_i)) \tag{14}$$

where n is the number of classes and $P(c_i) = \frac{|\{X \in T | y = c_i\}|}{|T|}$ is the probability of the class c_i.

The conditional entropy of training data set T given a value of a feature $x_a = v$ is calculated by

$$H(\{X \in T | x_a = v\}) = -\sum_{i=1}^{n} P\{c_i | x_a = v\} log_2(P\{c_i | x_a = v\}) \tag{15}$$

The information gain of the training data set T for an attribute a is defined in terms of entropy.

$$IG(T, a) = H(T) - H(T|a), \tag{16}$$

where $H(T|a) = \sum_{v \in val(a)} \frac{|\{X \in T | x_a = v\}|}{|T|} \cdot H(\{X \in T | x_a = v\})$ is the average conditional entropy given the feature a.

The pseudo code of the REPTree algorithm is depicted in Fig. 4. Note that in line 4 of the pseudo code. If the attribute is numeric type, we need to find the split point to facilitate the calculation of information gain.

```
Build_Tree(T, a_split){
  Calculate IG(T, a) for each attribute a.
  Find the split point if the attribute is numeric type.
  Find the split attribute a_max with the maximum IG among the attributes.
  if IG(T, a_max) > IG(T, a_split) {
    For all v ∈ val( a_max) {
       T= {X ∈ T | x_a_max = v}.
       Build_Tree(T, a_max).
    }
  }
}
```

Fig. 4. Pseudo code of building decision tree

Two users use the data collection application to collect around 500 training data for each wearing behaviors. Figure 5 is the DT model built from a user dataset by Weka. The maximum depth of tree was set to five to avoid overfitting. From the result of the DT model, we observed that three features: phone direction, light intensity and average intensity of acceleration were the three significant features in the classification.

The phone direction and light intensity were very useful to detect the wearing positions. The phone direction is usually the same for different wearing positions. For example, the phone direction is usually perpendicular to the ground when the smartphone is placed in the armband; the phone direction is usually parallel to the ground when the smartphone is placed in the purse near the waist. The light intensity is useful to detect the position where the difference in luminosity is obvious, e.g., the purse near the waist and the pocket of trousers. The average intensity of acceleration was useful to detect the mobility.

```
Phone_direction = 2.000
|   light < 1.81 : Still/Thigh
|   light >= 1.81
|   |   A_mean < 1.43 : Still/Hand
|   |   A_mean >= 1.43 : Walk/Hand
Phone_direction = 0.000
|   A_mean < 1.58
|   |   light < 2.62
|   |   |   V_mean < -0.02
|   |   |   |   light < 1.47 : Still/Arm
|   |   |   |   light >= 1.47 : Still/Waist
|   |   |   V_mean >= -0.02 : Still/Thigh
|   |   light >= 2.62 : Still/Hand
|   A_mean >= 1.58
|   |   light < 1.76 : Walk/Waist
|   |   light >= 1.76 : Walk/Hand
Phone_direction = 1.000
|   G_mean < 0.98
|   |   A_mean < 1.54
|   |   |   light < 2.78 : Still/Chest
|   |   |   light >= 2.78
|   |   |   |   V_mean < -0.05 : Still/Chest
|   |   |   |   V_mean >= -0.05 : Still/Hand
|   |   A_mean >= 1.54
|   |   |   Aσ < 2.88 : Walk/Chest
|   |   |   Aσ >= 2.88 : Walk/Thigh
|   G_mean >= 0.98
|   |   Aσ < 1.71
|   |   |   light < 1.6
|   |   |   |   Gσ < 0.94 : Walk/Arm
|   |   |   |   Gσ >= 0.94 : Walk/Thigh
|   |   |   light >= 1.6
|   |   |   |   A_mean < 3.23 : Walk/Chest
|   |   |   |   A_mean >= 3.23 : Walk/Hand
|   |   Aσ >= 1.71
|   |   |   light < 1.79 : Walk/Thigh
|   |   |   light >= 1.79 : Walk/Hand
```

Fig. 5. The result of the decision tree build by the Weka REPTree algorithm.

We used the same data collected from the users to evaluate the DT model by Weka. In the experiment, 80 percentage of the dataset was used as training set to train the DT model, and the rest 20 percentage of the dataset was used to test the DT model. Table 2. is the evaluation result. The accuracy was evaluated in terms of true positive (TP) rate, false positive (FP) rate and precision. The true positive is defined as the number of correct classification and the false positive is defined as the number of incorrect classification. The precision is defined as the true positive over the sum of the true positive and the false positive. The Precision, TP rate and FP rate were 93.9 %, 93 % and 1.3% in average, respectively. We can observe that the TP rates of still/waist and walk/arm were not good (34.5 % and 62.5 %, respectively). The still/waist was usually detected to the still/thigh. This is because the phone direction was usually the same when the user is sitting and the smartphone is placed in the purse near the waist or the pocket of trousers. Similar observation could be discovered in the walk/arm and walk/thigh. The phone direction is similar when the user is walking and the smartphone is strapped in the armband or the pocket of trousers. Nevertheless, the proposed approach is able to detect the wearing behaviors in most cases.

Table 2. The accuracy of the wearing behavior recognition.

Class	TP rate	FP rate	Precision
S/H	0.86	0	1
S/A	1	0.004	0.818
S/C	1	0.02	0.864
S/W	0.345	0	1
S/T	1	0.045	0.827
W/H	1	0	1
W/A	0.625	0.002	0.833
W/C	1	0.002	0.978
W/W	1	0	1
W/T	0.981	0.009	0.972
Weighted Avg	0,93	0,013	0.939

7 Conclusion

In this paper, we proposed a 3-stage framework to detect the wearing behaviors. The wearing behavior is a combination of human mobility and wearing positions. The framework consists of the training phase and the inference phase. Our wearing behavior detection framework starts with the data preprocessing to collect sensor data, extract sensor features and generate ground truths. The sensor features and ground truths are used to train thresholds and classification model in the attachment recognition stage and the wearing behavior recognition stage. After the models are ready, the framework can detect the wearing behaviors based on the sensor features. We evaluate our approach by the weka data mining tool on sensor data from 2 users. The experiment result shows that our approach can achieve the precision at 94 % in average. In the future, we will extend our work on more mobility patterns and wearing positions.

References

1. Santos, A.C., Cardoso, J.M.P., Ferreira, D.R., Diniz, P.C., Chaínho, P.: Providing user context for mobile and social networking applications. Perv. Mobile Comput. **6**(3), 324–341 (2010)
2. Pathak, A., Jindal, A., Hu, Y., Midkiff, S.: What is keeping my phone awake? Characterizing and detecting no-sleep energy bugs in smartphone apps. In: Mobisys (2012)
3. Nath, S.: ACE: exploiting correlation for energy-efficient and continuous context sensing. In: Mobisys (2012)
4. Yan, T., Chu, D., Ganesan, D., Kansal, A., Liu, J.: Fast app launching for mobile devices using predictive user context. In: Mobisys (2012)
5. https://play.google.com/store/apps/details?id=com.plexnor.gravityscreenoffpro
6. https://play.google.com/store/apps/details?id=net.vmid.bettersleep&hl=zh_TW
7. http://wiki.pentaho.com/display/DATAMINING/REPTree
8. http://www.cs.waikato.ac.nz/ml/weka/
9. http://en.wikipedia.org/wiki/Information_gain_in_decision_trees