J Med Syst (2016) 40: 140
DOI 10.1007/s10916-016-0497-2

PATIENT FACING SYSTEMS

Smartphone-Based Patients' Activity Recognition by Using a Self-Learning Scheme for Medical Monitoring

Junqi Guo [1,2] · Xi Zhou [2] · Yunchuan Sun [2] · Gong Ping [3] · Guoxing Zhao [2] · Zhuorong Li [2]

Received: 31 December 2015 / Accepted: 11 April 2016 / Published online: 22 April 2016

Abstract Smartphone based activity recognition has recently received remarkable attention in various applications of mobile health such as safety monitoring, fitness tracking, and disease prediction. To achieve more accurate and simplified medical monitoring, this paper proposes a self-learning scheme for patients' activity recognition, in which a patient only needs to carry an ordinary smartphone that contains common motion sensors. After the real-time data collection though this smartphone, we preprocess the data using coordinate system transformation to eliminate phone orientation influence. A set of robust and effective features are then extracted from the preprocessed data. Because a patient may inevitably perform various unpredictable activities that have no apriori knowledge in the training dataset, we propose a self-learning activity recognition scheme. The scheme determines whether there are apriori training samples and labeled categories in training pools that well match with unpredictable activity data. If not, it automatically assembles these unpredictable samples into different clusters and gives them new category labels. These clustered samples combined with the acquired new category labels are then merged into the training dataset to reinforce recognition ability of the self-learning model. In experiments, we evaluate our scheme using the data collected from two postoperative patient volunteers, including six labeled daily activities as the initial apriori categories in the training pool. Experimental results demonstrate that the proposed self-learning scheme for activity recognition works very well for most cases. When there exist several types of unseen activities without any apriori information, the accuracy reaches above 80 % after the self-learning process converges.

Keywords Smartphone · Activity recognition · Self-learning

This article is part of the Topical Collection on *Patient Facing Systems*

✉ Gong Ping
medical201611@gmail.com

1 Beijing Advanced Innovation Center for Future Education, Beijing Normal University, Beijing, People's Republic of China

2 College of Information Science and Technology, Beijing Normal University, Beijing, People's Republic of China

3 Peking University Third Hospital Geriatrics of Internal Department, Peking University Third Hospital, Beijing, People's Republic of China

Introduction

Activity recognition has recently attracted increasing attention in the field of mobile health, which generates effectively new solutions for safety monitoring, fitness tracking and disease prediction. For instance, when abnormal actions of solitary elders or postoperative patients such as falling down are detected, nursing staff or family members receive a warning message immediately so that sick people can receive prompt medical attention [1]. Besides, nutritionists can estimate a person's daily calorie consumption through continuous activity recognition and provide personalized suggestions on a healthy diet [2]. For the purpose of disease prediction, it is probable that the old people have a disease such as Alzheimer's disease, Parkinson's disease or epilepsy if their actions often deviate from normal habits and activities [3].

To the best of our knowledge, the existing activity recognition methods can be divided into two major categories: traditional methods [4, 5] and sensor-based methods [6–8]. Because of the rise of mobile health applications, sensor-based methods receive more attention than traditional methods. Regarding solution accuracy and simplification during practical use, smartphone-based methods [9, 10] have been gradually recommended as a substitute for sensor-based methods because the latter may deploy redundant and complicated sensor devices on human bodies. Modern smartphones usually have various built-in motion sensors such as accelerometers, gyroscopes and magnetometers, which makes it much easier to capture users' activity data and recognize their activities in real time using an ordinary smartphone.

However, the performance of smartphone-based activity recognition methods may be easily affected by two problems. The first problem is the variation of smartphone orientations and positions. The phone orientation in a pocket is usually not fixed, and one phone may be placed in different positions of the body. Thus, most of the current work needs to train different models for different phone orientations or different pocket positions through a large number of experiments, which greatly increases system complexity. The other problem is there is no aprior information (such as activity type label, type number and training samples) in the training dataset for "unknown activities". In this paper we define "known activities" and "unknown activities" as follows: "known activities" are those activities whose categories have already existed in the training dataset containing large training samples, whereas "unknown activities" are those activities whose categories have never been recorded in the training dataset, which also means there is no apriori information. Because there may exist a great number of unpredictable activities in a person's daily life, it is unrealistic to collect sample data for all types of activities for training or add all activity categories into the training dataset.

For the first problem mentioned above, researchers have proposed some feature extraction methods to manage it [11–15]. Sun et al. [16] considered six pocket positions and four phone orientations. They extracted several features, including the mean, variance, correlation, energy, and entropy, and then trained a support vector machine (SVM) classifier for activity identification. Apiwat et al. [17] used a mobile phone embedded with a tri-axial accelerometer and collected data for sixteen different phone orientations. Although their projection-based method could manage the variation of device orientations, they still suggested selecting different models for different phone positions.

For the second problem mentioned above, most existing methods can only recognize "known activities" with training data, whereas "unknown activities" cannot be classified correctly because there is no labeled category or data sample in the training sets. Considering the diversity of actual activities and the limitation of training sets, "unknown activity" recognition is of practical significance. Cheng et al. [18] proposed a "zero-shot learning" approach to recognize previously unseen activities. They used semantic attributes to represent patient activities, and then developed a two-layer attribute-based learning algorithm for recognition. User feedback was also employed to reinforce the recognition accuracy. Although the approach could recognize new activities by generalizing knowledge, there was a limitation that semantic attributes were manually defined, which greatly increased implementation complexity. Yin et al. [19] presented a novel two-phase approach to detect abnormal activities using one-class SVM and kernel nonlinear regression. However, sensor data were assumed to contain few abnormal actions in their approach, which may not be appropriate for real data collection.

In this paper, we propose a self-learning data analysis scheme for patients' activity recognition. For simplicity, a patient only needs to carry an ordinary smartphone that contains some common motion sensors such as an accelerometer, gyroscope and magnetometer, so that sensor data can be continuously collected in real time though this device. To solve the first problem mentioned above, sensor data are preprocessed using coordinate system transformation to eliminate phone orientation influence. A set of robust and effective features are then extracted from the preprocessed data. To solve the second problem mentioned above, we propose a self-learning activity recognition scheme that contains the following: 1) a novelty detection algorithm based on a kernel null Foley-Sammon transform (KNFST), to determine whether sensor data result from "known activities" or "unknown activities"; 2) a random forest algorithm for the classification of "known activities"; 3) a t-distributed stochastic neighbor embedding (t-SNE) algorithm for the dimension reduction of "unknown activity" data; 4) a density-based spatial clustering of applications with noise (DBSCAN) algorithm for "unknown activity" data clustering, where the clustered "unknown activities" can be labeled as new categories. These sample clusters with new category labels are then merged into the training dataset to retrain the KNFST-based novelty detector and the random forest classifier, so that the recognition ability of the self-learning model can be reinforced. By conducting experiments, we evaluate our scheme by using the data collected from two postoperative patient volunteers, including six labeled daily activities as the initial "known activity" categories. Experimental results prove that the proposed self-learning scheme for activity recognition works very well for most cases. The average accuracy converges to almost above 80 % when there exist various unknown activities in experiments.

To date, we have found few related studies on self-learning scheme for patients' activity recognition based on smartphones. Ho et al. [20] conducted some closely related research. They proposed a self-reconfigurable activity recognition algorithm based on active learning. However, their experiments were implemented within an intelligent home environment filled with many sensors for data collection. By contrast, the testee in our scheme only needs to carry an ordinary smartphone, which greatly simplifies system implementation.

The remainder of this paper is organized as follows. In "Data acquisition and preprocessing" section, we describe data acquisition based on a smartphone and data preprocessing using coordinate system transformation. "Feature extraction" section presents feature extraction, in which a set of robust and effective features are listed. In "The self-learning activity recognition scheme" section, we propose the entire self-learning framework which contains several algorithms for different purposes. In "Experiments" section, experiments are given for performance evaluation and comparison. Finally, we draw conclusions in "Conclusions" section.

Data acquisition and preprocessing

Smartphone-based data acquisition

An ordinary smartphone (e.g., Apple iPhone 6 and Samsung Galaxy Note 3) usually has abundant built-in motion sensors, such as an accelerometer, gyroscope, magnetometer, gravity accelerometer and linear accelerometer, which makes it much easier to capture a user's activity data in real time. The sensors involved in our scheme and the corresponding data definition are listed in Table 1.

Note that the sensor data above are measured in the smartphone coordinate system instead of the conventional earth coordinate system. As shown in Fig. 1, a smartphone coordinate is the coordinate system relative to the phone screen in its default orientation; thus, the directions of the smartphone coordinate axes change together with the change of screen orientation. Generally, the phone orientation in a pocket is usually not fixed. Some people carry a phone vertically, while others place it horizontally. Moreover, the device orientation may be unsteady all the time because of body movements. Therefore, values of sensor data measured in the smartphone coordinate system are inevitably and easily affected by orientation variation. All sensor data should be rotated into the earth coordinate system to eliminate differences in orientation variation, which will be further discussed in "Data preprocessing" section.

Another problem is the variation of the smartphone position on bodies. Different positions of a phone on human bodies can also significantly affect motion sensor measurements. For example, acceleration data collected by a smartphone in a coat pocket remain consistent with the whole body acceleration, whereas additional acceleration derived from swinging legs also exists in sensor data when the smartphone is placed in a trouser pocket. Therefore, it is necessary to select features that are robust and effective to prevent performance degradation due to position variation, which will be further discussed in "Feature extraction" section.

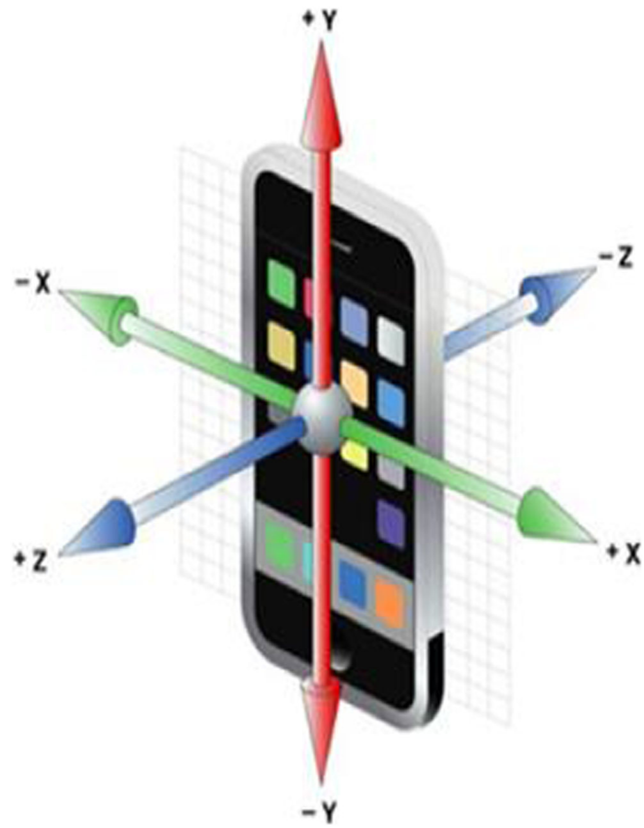


Fig. 1 Smartphone coordinate system

To collect and upload patient activity data in a real environment, we develop an application (APP) program based on the Android platform which contains several functions such as data acquisition, data uploading and information annotation including activity types, orientations and positions, as shown in Fig. 2. Detailed parameters for data acquisition, such as sampling frequency, duration, number and type of initial "known activities", will be provided for the experimental stage discussed in "Experiments" section.

Table 1 Motion sensors involved in our scheme

Sensor	Unit	Description
Accelerometer	m/s^2	Acceleration along the three axes (x, y, z)
Gyroscope	rad/s	Angular velocity around the three axes (x, y, z)
Magnetometer	μT	Geomagnetic field intensity along the three axes (x, y, z)
Gravity accelerometer	m/s^2	Gravitational acceleration along the three axes (x, y, z)
Linear accelerometer	m/s^2	Linear acceleration without gravity along the three axis (x, y, z)

Fig. 2 APP interface for data acquisition

Data preprocessing

Coordinate system transformation

We present a coordinate system transformation method, which rotates all sensor data into the earth coordinate system to manage the variation of device orientations. By doing this, data deviation derived from orientation variation can be eliminated.

Generally, coordinate system transformation is implemented by using a rotation matrix **R** from one coordinate system to another as follows:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R}\cdot\begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

Fortunately, the application programming interface (API) for Android application development has already provided a function "getRotationMatrix" which can directly calculate the rotation matrix **R** from the smartphone coordinate system to the earth coordinate system. We can find this function in the help document of Android developer. This function only requires sensor values of gravity accelerometer and magnetometer as two input parameters to compute the rotation matrix **R**. Then values of two parameters in the earth coordinate system can be obtained as follows:

$$linear\_acc\_earth = \mathbf{R}\cdot(acc - gravity) \tag{2}$$

$$gyro\_earth = \mathbf{R}\cdot gyro \tag{3}$$

where: *acc* and *gyro* denote the acceleration and angular velocity in the smartphone coordinate system, respectively, and *linear_acc_earth* and *gyro_earth* are the linear acceleration and angular velocity in the earth coordinate system, respectively. *gravity* denotes gravitational acceleration. Because *acc* contains a gravitational acceleration component, the linear acceleration created only by the patient's movement can be obtained through *acc* minus *gravity.*

Furthermore, because smartphone orientation information is helpful for identifying activities, we employ another function, "getOrientationto", which has also been defined in the Android API to calculate smartphone orientation parameters, as follows:

$$Orient = (azimuth, pitch, roll) \tag{4}$$

where: as shown in Fig. 3, *azimuth* represents the azimuth angle which rotates around the axis perpendicular to the earth ground; *pitch* represents the pitch angle which rotates around the axis along the east-west direction of the earth; *roll* represents the roll angle which rotates around the axis along the north-south direction of the earth.

We have five types of sensor data (accelerometer, gyroscope, magnetometer, gravity accelerometer and linear accelerometer) by direct measurement as described in "Smartphone-based data acquisition" section. After coordinate system transformation, three calculated values (*linear_acc_earth*, *gyro_earth* and *Orient*) are also acquired. Each of these eight physical quantities contains three-axis components. Considering time sampling, all the sensor data over the entire observation duration can be represented as 24 time series or a $24 \times N$ data matrix, where N denotes the sampling number.

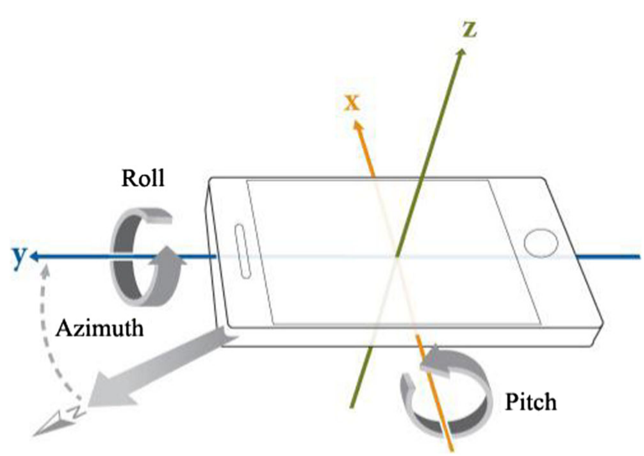


Fig. 3 Smartphone orientation defined by three attitude angles

Data segmentation

We use a fixed-size sliding window with 50 % overlapping to divide the $24 \times N$ data matrix mentioned above into slices. Assume that the activity type, device position and orientation are all constant within a sliding window. A data fragment can be denoted as a *Tetrad* as follows:

$$Tetrad = (slice,\ type,\ position,\ orientation) \tag{5}$$

where *slice* represents a $24 \times M$ data matrix and M denotes the sampling number within a window. By sliding the time window, we obtain a series of *Tetrad* which can be jointly represented as follows:

$$Preprocessed\_Data = \left[T_1,\ T_2,\ \ldots,\ T_{\lfloor(2N-M)/M\rfloor}\right] \tag{6}$$

where T_i denotes the i-th *Tetrad*. $\lfloor x \rfloor$ rounds x to the nearest integer towards minus infinity.

Feature extraction

The *Preprocessed_Data* in Eq. (6) contains meaningful information for recognizing patient activities, in addition to much noise. Selecting optimal features can not only extract more effective information from raw data contaminated by noise, but also considerably reduce the amount of data. For the purpose of comprehensively representing the characteristics of each activity and preventing performance degradation due to smartphone position variation, we present a set of robust and effective features based on previous studies [21–23]. The features chosen in this paper can be classified into three categories: descriptive statistics, correlation coefficients and zero-crossing rate.

Descriptive statistics

Descriptive statistics can reveal statistical characteristics of data very well, which may reflect statistical rules of human activities. Moreover, these characteristics are also very easy to calculate. We use seven common statistical indicators: the mean, standard deviation, maximum value, minimum value, 50 % quantile, skewness and excess kurtosis, among which skewness and excess kurtosis are given as follows:

$$skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^3}{\left(\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^{3/2}} \tag{7}$$

$$excess\ \ kurtosis = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^4}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^2} \tag{8}$$

where $\bar{x}$ denotes the arithmetic average of the data samples.

Correlation coefficients

Correlation coefficients describe the correlation between two random variables. We use two common correlation coefficients: Pearson's correlation coefficient and Spearman's rank correlation coefficient, which are given by:

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y} \tag{9}$$

$$\rho_{spearman} = \frac{\sum_i\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_i\left(x_i-\bar{x}\right)^2\sum_i\left(y_i-\bar{y}\right)^2}} \tag{10}$$

where: $E[\cdot]$ denotes mathematical expectation; μ_X and μ_Y represent the expectation of X and Y, respectively; σ_X and σ_Y represent the standard deviation of X and Y, respectively. As mentioned in "Data preprocessing" section, we obtain eight physical quantities after coordinate system transformation, each of which contains three-axis components. For each physical quantity, we calculate correlation coefficients between each two axes (e.g., X and Y, Y and Z, X and Z).

Zero-crossing rate

The zero-crossing rate has been widely used in the field of speech recognition and music retrieval. It can also be used for feature extraction of human activities, since people's activities usually have repeatability, and the zero-crossing rate is a good index for repeatability description. In addition, it is very easy to calculate. The zero-crossing rate is given by:

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1}\operatorname{sgn}\{s_t s_t{-}1 < 0\} \tag{11}$$

where: s is a signal of length T; the function $\operatorname{sgn}\{w\}$ is equal to 1 or 0 when the value of w is true or false, respectively. Here we calculate the zero-crossing rate of the data for each axis.

Finally, we extract all the above parameters from each *slice* in Eq. (5) and combine them to obtain a feature vector $\mathbf{x}$, while the activity type in accordance with this $\mathbf{x}$ is *type* in Eq. (5). Therefore, the entire *Preprocessed_Data* in Eq. (6) can be converted into a training dataset that is denoted as follows:

$$training\_dataset = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n\} \tag{12}$$

where $\mathbf{s}_i$ denotes the i-th training sample given by:

$$\mathbf{s}_i = (\mathbf{x}_i,\ type_i) \tag{13}$$

where $\mathbf{x}_i$ denotes the i-th feature vector and $type_i$ represents the activity type to which $\mathbf{x}_i$ belongs.

The self-learning activity recognition scheme

Framework of the self-learning scheme

We have obtained a training dataset in Eq. (12) for activity classification. However, it is unrealistic to collect all types of activity samples for training, since there exist a great number of unseen activities in a patient's daily life. As mentioned before, we assume that "known activities" represent those activities whose categories have already labeled in the training dataset with large training samples, while "unknown activities" mean those activities whose categories have never been recorded in the training dataset. Obviously, when it comes to "unknown activities", traditional data classifiers have no ability to identify them accurately. Therefore, it is of practical significance to recognize both the known and unknown activities. Here we propose a self-learning activity recognition scheme, which can adaptively distinguish whether sensor data result from "known activities" or "unknown activities", and then automatically learn new categories from the "unknown activities" to reinforce ability of activity recognition by itself. The framework of the self-learning scheme is shown as follows:

As shown in Fig. 4, the self-learning framework contains several main modules as follows:

- *Module 1*: Novelty detection. A kernel null Foley-Sammon transform (KNFST) [24, 25] based novelty detection algorithm is proposed to automatically determine whether test data belong to "known activities" or "unknown activities";
- *Module 2*: Classification of "known activities". A random forest algorithm [26, 27] is employed to generate a classifier. Sensor data belonging to "known activities" can be further classified into several known categories, which also means patients' activities are recognized.
- *Module 3*: Clustering of high-dimensional "unknown activity" data. Since "unknown activity" data are usually represented with high-dimensional feature vectors, we employ a t-distributed stochastic neighbor embedding (t-SNE) algorithm [28] for dimension reduction, combined with a density-based spatial clustering of applications with noise (DBSCAN) algorithm [29] for low-dimensional data clustering.

The proposed scheme creatively integrates novelty detection, classification of "known activity" data, clustering of high-dimensional "unknown activity" data and feedback into a comprehensive self-learning framework. The detailed working procedure can be described as follows:

- *Step 1*: Training. Train the KNFST-based novelty detector in *Module 1* and the random forest classifier in *Module 2* according to the training dataset in Eq. (12).
- *Step 2*: Recognition. Determine whether test data belong to the "known activity" type by using the KNFST-based novelty detector. If so, import the data into the random forest classifier for activity recognition; if not, import the data into a data pool of unknown activities for the following self-learning.
- *Step 3*: Self-learning. When the data of unknown activities accumulate enough in the pool, start up a self-learning

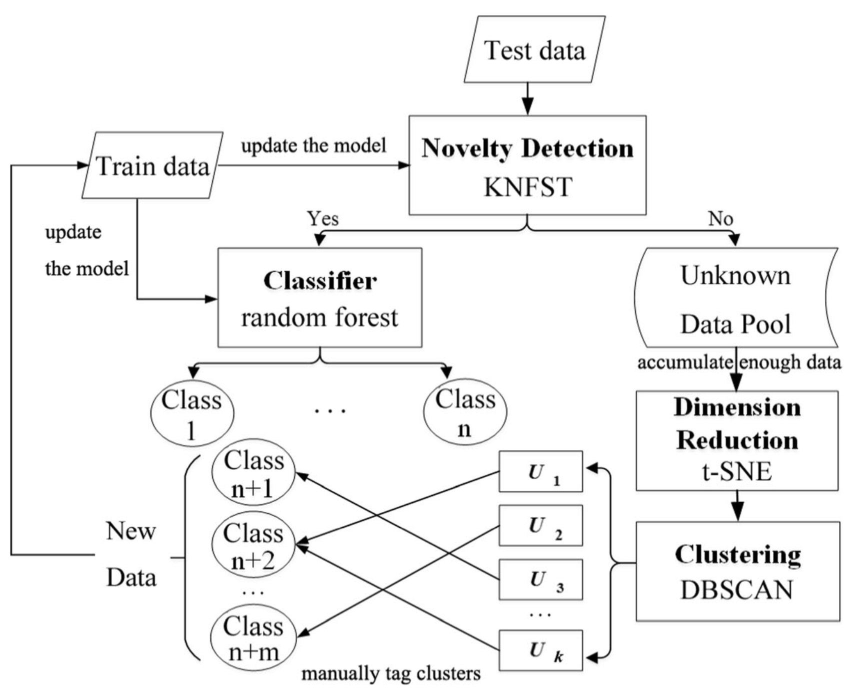


Fig. 4 Framework of the self-learning scheme

process. Firstly, reduce the dimension of the data represented with high-dimensional feature vectors by using the t-SNE algorithm. Next, cluster the data with the DBSCAN algorithm to put similar unknown activity data into the same category. Thirdly, label each cluster manually to give each category of unknown activity a class name. Note that artificial labelling is just to give each cluster a concept that people can understand. In fact, new classes have been automatically learned by the self-learning model. Furthermore, the self-learning process will converge to a stable state when all types of unknown activities are learned and there is no new labelling, which could be easily achieved under the condition that the activity type is limited for a patient after the surgery and there is no diverse type of activities.

- *Step 4*: Feedback and update. These sample clusters with new category labels are merged into the original training dataset to retraining the KNFST-based novelty detector and the random forest classifier, so that ability of the self-learning model can be reinforced.

Note that the framework of the proposed self-learning scheme contains four important algorithms: the KNFST-based novelty detection algorithm, the random forest algorithm, the t-SNE algorithm and the DBSCAN algorithm, among which the latter three algorithms are proposed by some previous studies, whereas the first one is proposed by us on the basis of the KNFST. Therefore, we will discuss the KNFST-based novelty detection algorithm in details below.

The KNFST-based novelty detection algorithm

As mentioned above, determining whether test data belong to "known activities" or "unknown activities" is vital for the subsequent self-learning process. In some scenarios such as nursing observation, we are not interested in recognizing all types of activities, but only pay close attention to several specific activity types. For example, we are only concerned about some basic rehabilitation activities (e.g., lying-down, and walking) and several dangerous activities (e.g., a sudden tumble) for elder care. As for other trivial activities, we just need to predict them to be "unknown" and don't care about their specific types. In the field of machine learning, the problem of identifying whether a test sample belongs to a known type or not is defined as "novelty detection". Novelty detection proves to be a complicated problem for high dimensional data [30–33]. Unfortunately, sensor data have been represented as high-dimensional feature vectors after feature extraction, as mentioned in "Feature extraction" section, which makes the design of novelty detection in our scheme more complicated. To solve this problem, we propose a novelty detection algorithm based on the kernel null Foley-Sammon transform (KNFST). Experiments in "Experiments" section demonstrate that it is better than some existing novelty detection algorithms in recognition accuracy.

KNFST [34, 35] is a mapping transformation which attempts to map the samples of the same class into a single point, while the samples of different classes are mapped into different points, respectively. Based on this transformation, we implement novelty detection as follows:

- *Step 1*: Training. Relying on the KNFST, find out an optimal transformation matrix $\mathbf{W}$ according to the training data. By using this transformation matrix, samples of the same class are mapped into a single point which is defined as the central point of this class. Optimization of $\mathbf{W}$ ensures that the within-class scatter is equal to 0, while the between-class scatter is as large as possible.
- *Step 2*: Definition of novelty score. If we obtain a new observation sample $\mathbf{y}$ whose class is unknown, implement the same transformation on $\mathbf{y}$ by using $\mathbf{W}$ to acquire its projection point in the mapping space. Define "novelty score" of $\mathbf{y}$ as the smallest distance from its projection point to central points of all classes, which is represented as follows:

$$NoveltyScore(\mathbf{y}) = \min_{1\leq i\leq C} distance\left\{t^{*}, t^{(i)}\right\} \tag{14}$$

As is shown in Fig. 5, samples of *Class 1*, *2* and *3* are respectively mapped to $t^{(1)}$, $t^{(2)}$ and $t^{(3)}$ after mapping transformation. Also, the test sample $\mathbf{y}$ is projected to the point t^*. Distances from t^* to $t^{(1)}$, $t^{(2)}$ and $t^{(3)}$ are d_1, d_2 and d_3, respectively. It is obvious that the smallest distance is d_2, which is regarded as the novelty score of the observation sample $\mathbf{y}$. If $\mathbf{y}$ belongs to a known class, its novelty score is inevitably small. Otherwise, $\mathbf{y}$ is mapped far away from central points of all classes, which leads to a large novelty score of $\mathbf{y}$. Therefore, the novelty score reveals possibility whether a test sample belongs to an unknown class. The larger the novelty score is, the more likely the sample is from an unknown class.

- *Step 3*: Threshold-based Decision. An appropriate threshold δ should be selected for decision. If the novelty score of a test sample is larger than the threshold, it can be regarded as the "unknown activity"; otherwise, the sample belongs to the "known activity". Threshold-based Decision is represented as follows:

$$Class(\mathbf{y}) = \begin{cases} unknown & NoveltyScore(\mathbf{y}) > \delta \\ known & NoveltyScore(\mathbf{y}) \leq \delta \end{cases} \tag{15}$$

Note that the value of the threshold δ plays a key role in decision accuracy. Generally, the optimal value of δ is often given by experiments. Here we give a practical method for threshold selection. Firstly, separate one third of training

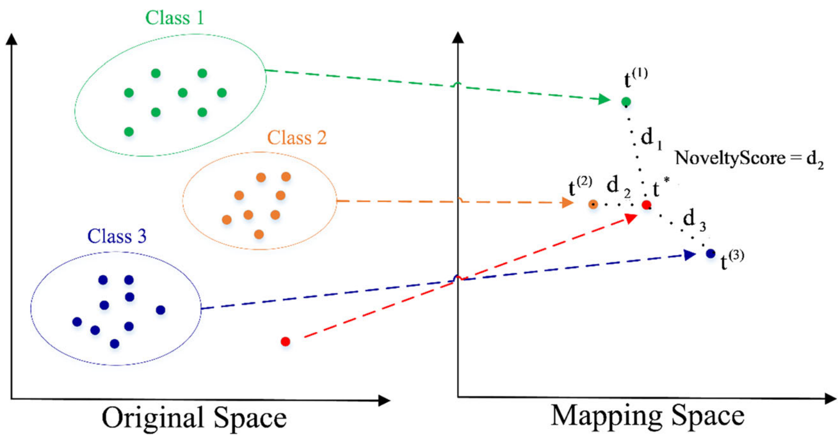


Fig. 5 Geometrical principle of the KNFST-based novelty detection algorithm

samples from the whole training dataset to be regarded as the testing dataset. Train the novelty detector by using the rest two third of samples in the training dataset. Next, acquire novelty scores of all samples in the testing dataset. Assume that these scores obey a normal distribution, estimate the standard deviation σ. Select $\delta = 2\sigma$ as the optimal value of the threshold.

Pseudo-codes of the KNFST-based novelty detection algorithm are given in Appendix.

Experiments

Data acquisition

In order to achieve data collection in a real environment, we develop an APP (application) program based on the Android platform which contains several functions, such as data acquisition, data uploading and information annotation including activity types, orientations and positions. Meanwhile, we employ a "Java + Spring" framework to develop a set of application programming interface (API) for the server, so that a smartphone can upload its sensor data to the server. Data are finally stored in the MySQL database for activity recognition.

To verify the robustness of the proposed scheme under the condition of different smartphone orientations and positions, we consider 4 different orientations (vertically inward, vertically outward, horizontally inward and horizontally outward) and 2 positions (coat pocket, and trouser pocket), as shown in Figs. 6 and 7, respectively. All training data in experiments are collected from two patient volunteers recruited by us. We select 6 types of activities (walking, running, going upstairs, going downstairs, standing and sitting) as the initial "known activity" types to build a training dataset. Note that these types of activities are common during a patient' postoperative and rehabilitation periods. Volunteers keep doing each type of activity with each orientation and each position for 5 min, so that two volunteer generate the training data with the total amount of 4 (orientations) × 2 (positions) × 6 (activity types) × 2 (volunteers) × 5 (minutes) = 480 min. In addition, the sampling frequency is set as 25Hz. Note that the sampling frequency can be adjustable according to the energy consumption of smart phones.

After data acquisition, we implement data preprocessing and feature extraction as described in "Data preprocessing" section and "Feature extraction" section. Finally, we build a training dataset including 24,925 labeled samples in total. Figure 8 shows data volume of each type of activity in the training dataset.

Evaluation criteria

For multivariate classification, accuracy is a typical and common index to evaluate classification performance. Here we employ "accuracy" to evaluate activity recognition performance of the proposed self-learning scheme. It is defined as the percentage of correctly predicted samples in the whole sample set, which is given by:

$$Accuracy\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathrm{sgn}\left\{\hat{y}_i = y_i\right\} \quad (16)$$

where: $\mathbf{y}$ and $\hat{\mathbf{y}}$ represent the real type vector and the predicted type vector, respectively; y_i and $\hat{y}_i$ denote the i-th element of $\mathbf{y}$ and $\hat{\mathbf{y}}$, respectively; n is the number of test samples; the function $\mathrm{sgn}\{w\}$ is equal to 1 or 0 when the value of w is true or false, respectively.

Experimental results

Performance of the proposed feature extraction method

As mentioned in "Feature extraction" section, we have selected three categories of features: descriptive statistics, correlation coefficients and zero-crossing rate. To verify the feasibility and efficiency of these feature parameters, we compare our method with other two feature extraction methods: the

Fig. 6 Four orientations of smartphones

Yunus's method [21] and the Jennifer's method [36]. To ensure fairness of the comparison, we employ the same training dataset, testing dataset and classifiers for all the three feature extraction methods. Both a random forest classifier and a support vector machine (SVM) classifier are employed for data classification.

Figure 9 shows accuracy performance of different feature extraction methods. Obviously, our feature extraction method is better than other two methods, no matter whether we use the random forest classifier or the SVM classifier for data classification.

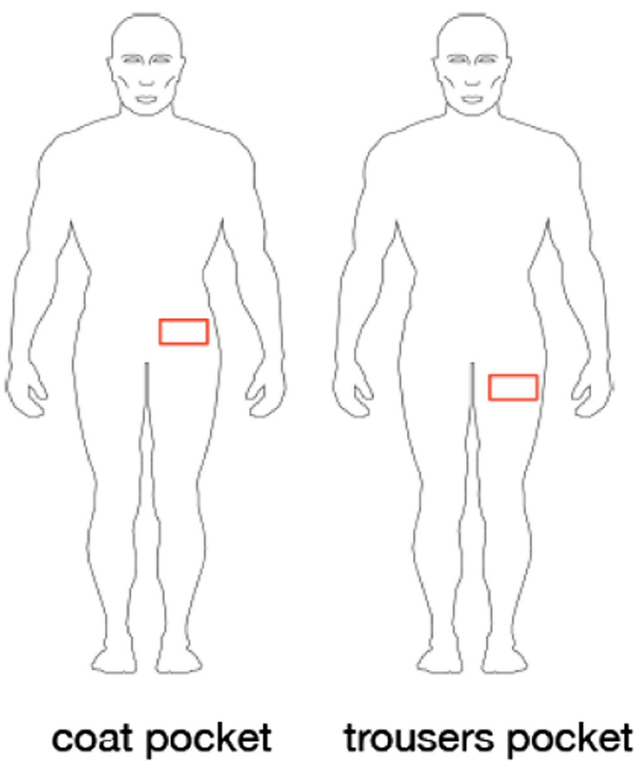


Fig. 7 Two positions of smartphones

Performance of the proposed KNFST-based novelty detection algorithm

To evaluate performance of the KNFST-based novelty detection algorithm proposed in "The KNFST-based novelty detection algorithm" section, we compare it with other two common novelty detection algorithms: One-class SVM [32] and Binary SVM [33]. In addition, we employ a "Random" method as the baseline of novelty detection performance. The principle of the "Random" method is to guess whether the test data belong to the known or unknown activities only by using the ratio of known and unknown samples in the training dataset, so it can be regarded as the simplest and worst novelty detection method with a little above 50 % detection accuracy.

Figure 10 shows accuracy performance of the four novelty detection algorithms when there exists only one type of unknown activity, in which "kernel" denotes our algorithm, and the label above each sub-figure stands for one type of unknown activity. From the figure, we can see that the detection

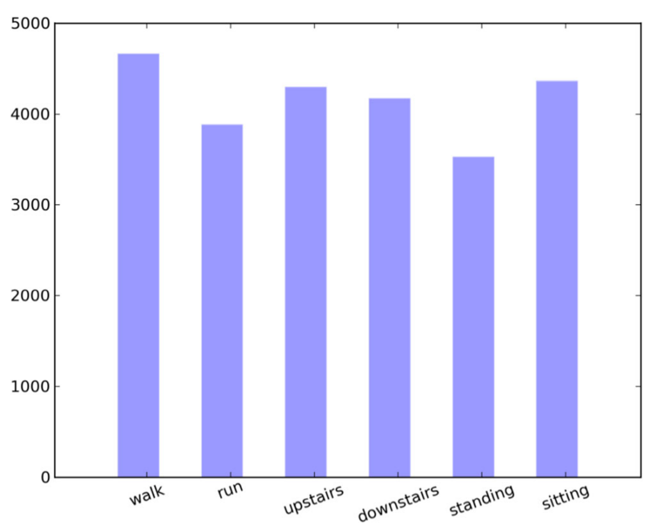


Fig. 8 Data volume of the training dataset

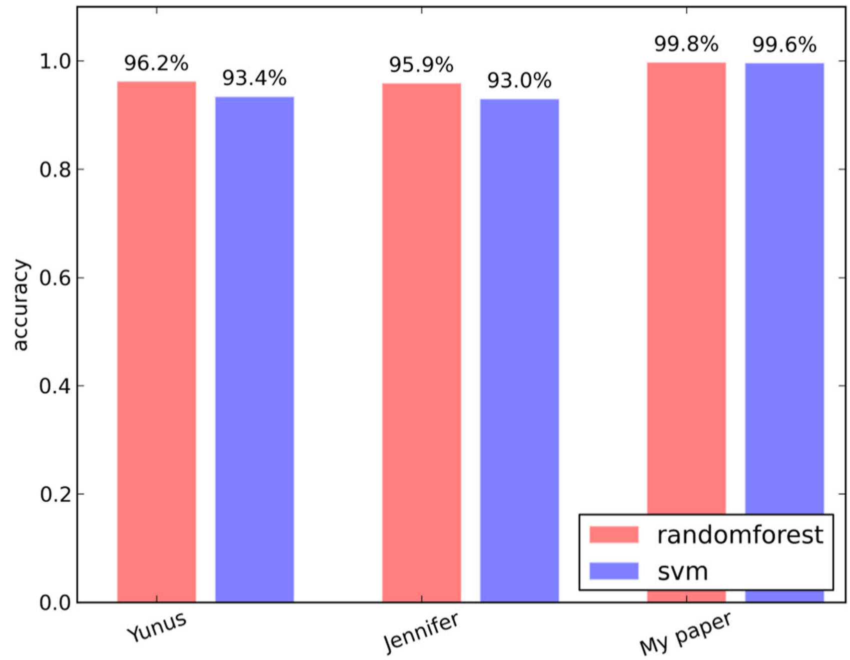


Fig. 9 Accuracy performance of the three feature extraction methods

accuracy of our algorithm is much higher than others in all situations. Besides, we find that the accuracy of our algorithm is always higher than 80 % except when the unknown category is "standing" or "sitting". The reason is that "standing" and "sitting" are two similar human activities for sensors. The body is quiescent under both "standing" and "sitting" states, which makes the algorithm difficult to distinguish them.

There are 15 different combinations when we simultaneously select 2 types as unknown activities from the total 6 activity types. Figure 11 shows the accuracy performance of

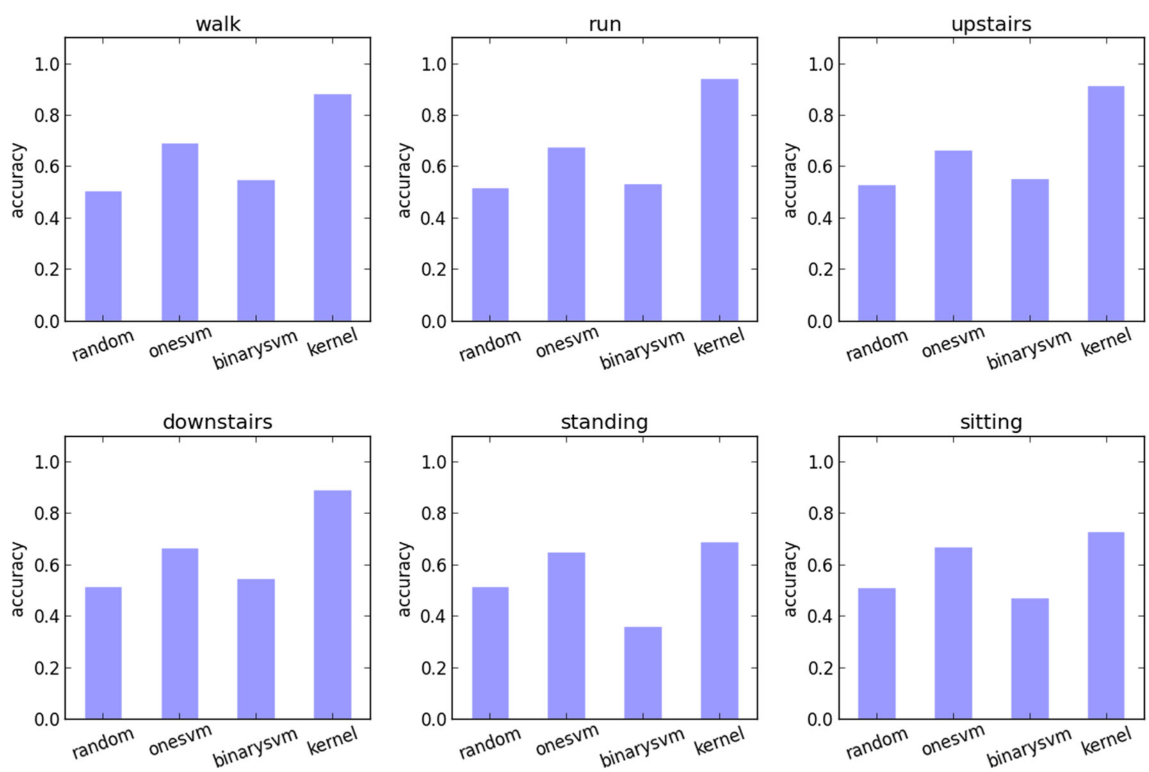


Fig. 10 Accuracy performance of the four novelty detection algorithms. (Under only one type of unknown activity)

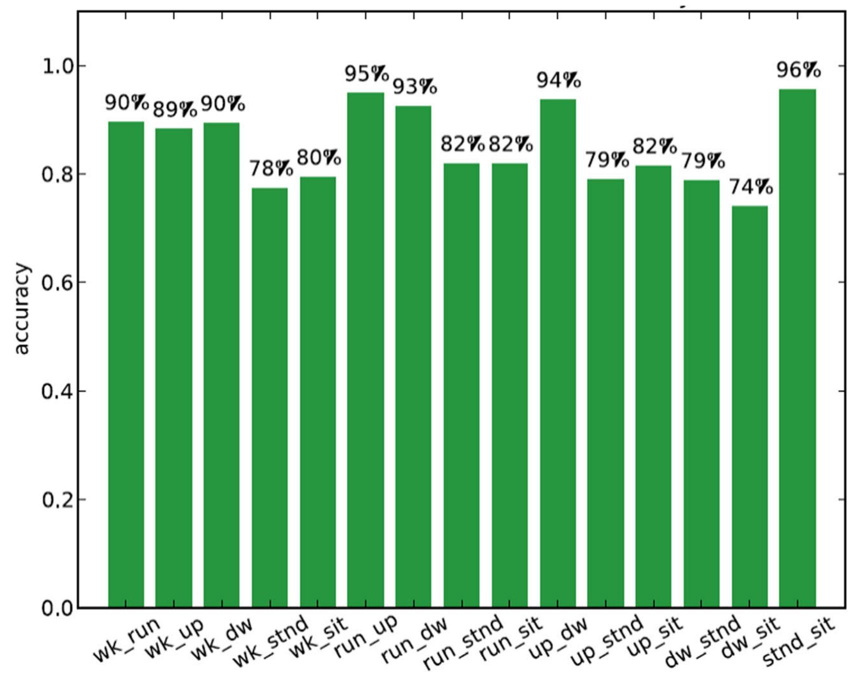


Fig. 11 Accuracy performance of the KNFST-based novelty detection algorithm. (Under different combinations of two types of unknown activities)

the proposed novelty detection algorithm when there are two types of unknown activities. Among the 15 combinations, accuracy of 7 combinations reaches about 90 %. The highest accuracy reaches 96 % and the corresponding combination is “standing + sitting”. Moreover, accuracy of the rest 8 combinations is 70 % ~ 80 %. Note that these 8 combinations have something in common: one of the “standing” and “sitting” is the unknown type and the other is the known type. As is mentioned before, it is difficult for the algorithm to distinguish “standing” and “sitting”. Therefore, when both of them belong to the unknown types, we obtain higher detection accuracy, since we needn’t distinguish them from each other. But when one of them belongs to the unknown type and the other belongs to the known type, we obtain lower accuracy.

Performance of the proposed self-learning scheme

Figures 12 and 13 present the distribution situation of the unknown activity data after the t-SNE dimension reduction

Fig. 12 Dimension reduction result for unknown activity data by using the t-SNE. (Unknown activities: Run and Upstairs)

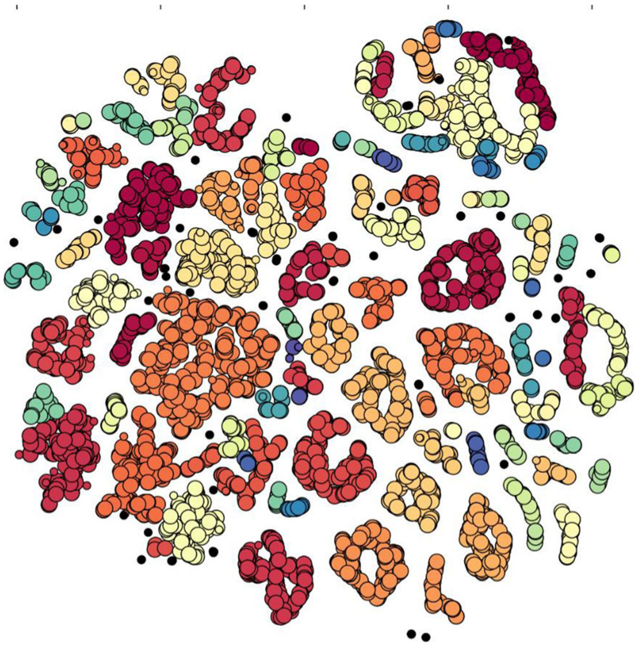


Fig. 13 Clustering result for unknown activity data by using the DBSCAN. (Unknown activities: Run and Upstairs)

and the DBSCAN clustering, respectively. Obviously, both algorithms work well in the proposed self-learning model.

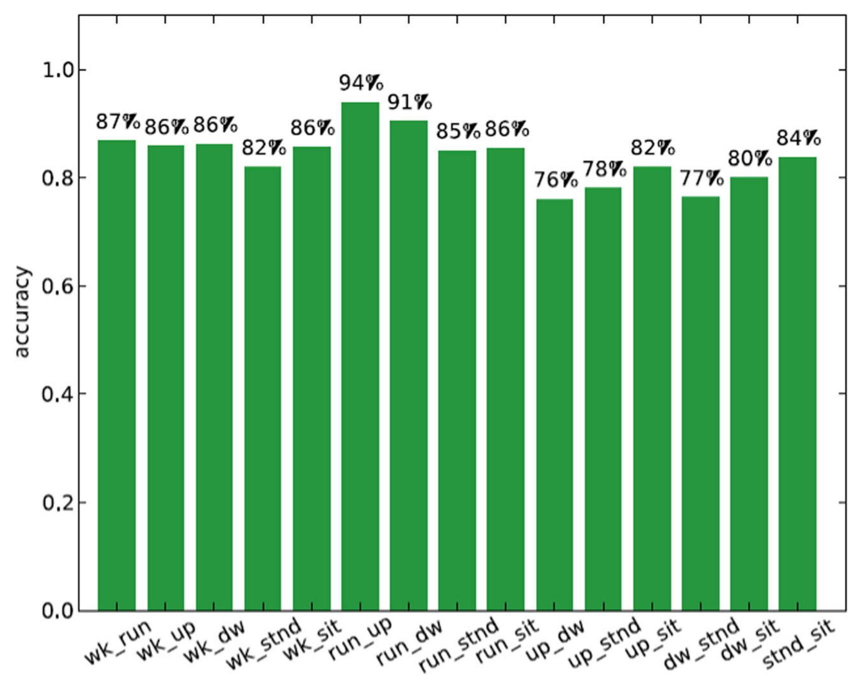
The samples labeled "5" and "6" in Fig. 12 denote the two types of unknown activities: "Run" and "Upstairs", while other samples labeled from "1" to "4" are the known activity samples. From the figure, we find that the t-SNE algorithm makes the samples belonging to the same class cluster together. Moreover, the t-SNE can also cluster some known activity samples together, or scatter them in the region with small sample density, which will be recognized as noise points (the solid black points in figures) by the DBSCAN algorithm. Therefore, the probability that these known activity samples are mistakenly labeled as the unknown activity is reduced.

Figure 14 shows the accuracy of the proposed self-learning scheme under the combination of different types of unknown activities. It is obvious that the accuracy reaches above 80 % in most cases after the self-learning process converges. Sometimes it reaches even above 90 %, which demonstrates the feasibility and efficiency of the self-learning process when there exist several types of unseen activities without any apriori information in training dataset.

Conclusions

We have proposed a self-learning data analysis scheme for patients' activity recognition. By using this scheme, a patient only needs to carry an ordinary smartphone which contains motion sensors for automatic data collection and uploading. The server preprocesses the sensor data to eliminate orientation influence, and then extracts a set of effective features from the data for further analysis. Moreover, a self-learning framework is proposed for recognizing unpredictable activities without any apriori knowledge in the training dataset. A key functional module in self-learning process is the proposed KNFST-based novelty detection algorithm, which determines whether there are apriori categories in the training dataset that

Fig. 14 Accuracy performance of the proposed self-learning scheme



well match with the unpredictable activity data. If not, these data are automatically assembled into clusters with new category labels. The clustered samples combined with the acquired new category labels are then merged into the training dataset to reinforce recognition ability of the self-learning model. Experiments have demonstrated the feasibility and efficiency of the self-learning scheme for unpredictable activity recognition. In the future work, we plan to recruit more volunteers for sensor data collection and performance evaluation of the proposed scheme.

Acknowledgments This research is sponsored by National Natural Science Foundation of China (No.61401029, 61171014, 61472044, 61472403, 61371185), Beijing Advanced Innovation Center for Future Education (BJAICFE2016IR-004), the Fundamental Research Funds for the Central Universities (No.2012LYB46), and Beijing Youth Excellence Program (YETP0296).

Appendix

The KNFST-based novelty detection algorithm

Input:

$X_{train} \in R^{m\times n}$: m samples, n features

$X_{test} \in R^{k\times n}$: k samples, n features

$Model$: model obtained by training

Output:

$scores$: $k\times 1$ column vector corresponds to k novelty values of k samples

1. Compute kernel matrix $K_t \in R^{m\times k}$, where all elements of every position satisfy $k\left(x_i, x_j\right): x_i \in X_{train}$ and $x_j \in X_{test}$
2. Compute projection of test samples $T_{proj} = K_t \cdot Model.proj$
3. Compute the distance of projections between test samples and each class $Dist = euclidean\left(T_t, Model.classCenter\right)$
4. Compute the novelty of i-th sample $F\left(i\right) = \min_{j\in[1,c]} Dist\left(i, j\right)$
5. Discriminate whether the sample is a novel sample or not according to formula 15

References

1. Győrbíró, N., Fábián, Á., and Hományi, G., An activity recognition system for mobile phones[J]. *Mobile Netw Appl* 14(1):82–91, 2009.
2. Arif, M., Bilal, M., Kattan, A., et al., Better physical activity classification using Smartphone acceleration sensor[J]. *J Med Syst* 38(9):1–10, 2014.
3. Moran, A. L., Ramrez-Fernandez, C., Meza-Kubo, V., Orihuela-Espina, F., Garca-Canseco, E., Grimaldo, A. I., and Sucar, E., On the e_ect ofprevious technological experience on the usability of a virtual rehabilitation tool forthe physical activation and cognitive stimulation of elders. *J Med Syst* 39(9):1–11, 2015.
4. Poppe, R., A survey on vision-based human action recognition[J]. *Image Vis Comput* 28(6):976–990, 2010.
5. Turaga, P., Chellappa, R., Subrahmanian, V. S., et al., Machine recognition of human activities: A survey[J]. *IEEE Trans Circ Syst Video Technol* 18(11):1473–1488, 2008.
6. Bao, L., and Intille, S. S., *Activity recognition from user-annotated acceleration data[M]//Pervasive computing*. Springer, Berlin Heidelberg, pp. 1–17, 2004.
7. Maurer, U., Rowe, A., Smailagic, A., et al., *Location and activity recognition using eWatch: A wearable sensor platform[M]// Ambient Intelligence in Everyday Life. Springer*: BerlinHeidelberg: 86–102, 2006.
8. Maurer, U., Smailagic, A., Siewiorek, D. P., et al., *Activity recognition and monitoring using multiple sensors on different body positions[C]//Wearable and Implantable Body Sensor Networks, 2006*. BSN 2006. International Workshop on. IEEE, 4 pp.-116, 2006.
9. Incel, O. D., Kose, M., and Ersoy, C., A review and taxonomy of activity recognition on mobile phones[J]. *Bio Nano Sci* 3(2):145–171, 2013.

10. Keally, M., Zhou, G., Xing, G., et al., *Pbn: towards practical activity recognition using smartphone-based body sensor networks[C]// Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. ACM*: 246–259, 2011.
11. Blanke, U., Schiele, B., Sensing location in the pocket[J]. *Ubicomp Poster Session*. 2, 2008.
12. Yang, J., *Toward physical activity diary: motion recognition using simple acceleration features with mobile phones[C]//Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics. ACM*: 1–10, 2009.
13. Khan, M., Ahamed, S. I., Rahman, M., et al., *A feature extraction method for realtime human activity recognition on cell phones[C]// Proceedings of 3rd International Symposium on Quality of Life Technology (isQoLT 2011)*. Toronto, Canada. 2011.
14. Thiemjarus, S., *A device-orientation independent method for activity recognition[C]//Body Sensor Networks (BSN), 2010 International Conference on. IEEE*. 19–23, 2010.
15. Theekakul, P., Thiemjarus, S., Nantajeewarawat, E., et al., *A rule-based approach to activity recognition[M]//Knowledge, Information, and Creativity Support Systems*. Springer, Berlin Heidelberg, pp. 204–215, 2011.
16. Sun, L., Zhang, D., Li, B., et al., *Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations[M]//Ubiquitous intelligence and computing*. Springer, Berlin Heidelberg, pp. 548–562, 2010.
17. Henpraserttae, A., Thiemjarus, S., Marukatat, S., *Accurate activity recognition using a mobile phone regardless of device orientation and location[C]//Body Sensor Networks (BSN), 2011 International Conference on. IEEE*. 41–46, 2011.
18. Cheng, H. T., Sun, F. T., Griss, M., et al., *Nuactiv: Recognizing unseen new activities using semantic attribute-based learning[C]// Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM*. 361–374, 2013.
19. Yin, J., Yang, Q., and Pan, J. J., Sensor-based abnormal human-activity detection[J]. *IEEE Trans Knowl Data Eng* 20(8):1082–1090, 2008.
20. Ho, Y., Lu, C., Chen, I., et al., *Active-learning assisted self-reconfigurable activity recognition in a dynamic environment[C]// Proceedings of the 2009 I.E. international conference on Robotics and Automation. IEEE Press*: 1567–1572, 2009.
21. Ustev, Y. E., Durmaz, I. O., Ersoy, C., *User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal[C]//Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM*: 1427–1436, 2013.
22. Anguita, D., Ghio, A., Oneto, L., et al., *A public domain dataset for human activity recognition using smartphones[C]//European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*. 2013.
23. Müller, M., Dynamic time warping[J]. Inform Retr Music Motion. 69–84, 2007.
24. Foley, D. H., and Sammon, J. W., Jr., An optimal set of discriminant vectors[J]. *IEEE Trans Comput* 100(3):281–289, 1975.
25. Guo, Y. F., Wu, L., Lu, H., et al., Null Foley–Sammon transform[J]. *Pattern Recogn* 39(11):2248–2251, 2006.
26. Breiman, L., Random forests[J]. *Mach Learn* 45(1):5–32, 2001.
27. Ho, T. K., The random subspace method for constructing decision forests[J]. *IEEE Trans Pattern Anal Mach Intell* 20(8):832–844, 1998.
28. Van der Maaten, L., and Hinton, G., Visualizing data using t-SNE[J]. *J Mach Learn Res* 9(2579–2605):85, 2008.
29. Ester, M., Kriegel, H. P., Sander, J., et al., A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Kdd. 96(34): 226–231, 1996.
30. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., et al., Estimating the support of a high-dimensional distribution[J]. *Neural Comput* 13(7):1443–1471, 2001.
31. Ko, H., Baran, R., Arozullah, M., *Neural network based novelty filtering for signal detection enhancement[C]//Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on. IEEE*: 252–255, 1992.
32. Bishop, C. M., *Novelty detection and neural network validation[C]//Vision, Image and Signal Processing, IEE Proceedings-. IET*. 141(4): 217–222, 1994.
33. Muñoz, A., and Muruzábal, J., Self-organizing maps for outlier detection[J]. *Neurocomputing* 18(1):33–60, 1998.
34. Zheng, W., Zhao, L., and Zou, C., Foley-Sammon optimal discriminant vectors using kernel approach[J]. *IEEE Trans Neural Netw* 16(1):1–9, 2005.
35. Lin, Y., Gu, G., Liu, H., et al., *Kernel null foley-sammon transform[C]//Computer Science and Software Engineering, 2008 International Conference on. IEEE*. 1: 981–984, 2008.
36. Kwapisz, J. R., Weiss, G. M., and Moore, S. A., Activity recognition using cell phone accelerometers[J]. *ACM SigKDD Explorations Newsl* 12(2):74–82, 2011.