

Received March 1, 2020, accepted March 22, 2020, date of publication March 30, 2020, date of current version April 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984456

W-Trans: A Weighted Transition Matrix Learning Algorithm for the Sensor-Based Human Activity Recognition

CHANGHAI WANG¹, BO WANG¹, HUI LIANG¹, JIANZHONG ZHANG², WANWEI HUANG¹, AND WANGWEI ZHANG¹

¹Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450002, China

²College of Cyber Science, Nankai University, Tianjin 300350, China

Corresponding author: Changhai Wang (chw@zzuli.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61975187, in part by the Key Science and Technology Program of Henan Province under Grant 192102210291, Grant 202102210174, Grant 202102210383, and Grant 202102210397, and in part by the Key Scientific Research Projects of Henan High School under Grant 19A520043 and Grant 20B520039.

ABSTRACT The sensor-based human activity recognition has been widely applied in behavior tracking, health monitoring, indoor localization etc. Using activity continuity to assist activity recognition is an important research issue, in which the activity transition matrix which describes the activity transformation in real scenarios is the most important parameter. Aiming at the problem that the current classic transition matrix learning algorithm cannot fuse weights of sample classification results, a weighted transition matrix learning algorithm is proposed in this paper. First, the basic definitions of an improved Hidden Markov Model (HMM) which fuses weights of classification results are given. Then, the recursive formula of transition matrix learning is derived, and the learning algorithm W-Trans is put forward. Finally, the proposed algorithm is simulated with the public data sets. The evaluation results show that the proposed algorithm outperforms the classical Baum-Welch algorithm under evaluation metrics of both the cosine similarity and the euler distance. By applying W-Trans to current activity recognition post-process methods, the advantage of our method is verified.

INDEX TERMS Activity recognition, Hidden Markov Model, parameter learning, transition matrix.

I. INTRODUCTION

A. BACKGROUND

In the past decade, the sensor-based human activity recognition [1], [2] has been a key research field in industry and academic field. In this research, the custom-made devices integrated with multiple inertial sensors (e.g., accelerometer, gyroscope) were bounded on human bodies. These devices can capture body movement and generate real-time sensor data as humans perform activities such as walking, running etc. This research has brought about many applications such as behavior tracking [3], [4], health monitoring [5], [6], indoor localization [7], [8] and so on.

Currently, the supervised learning is the commonest technique adopted for activity recognition [9], in which a classification model is first trained with labeled samples, and

then applied to classify the unlabeled ones [10]. However, there are still several challenges affecting the performance of sensor-based activity recognition [11], [12]. First, for the same activity, the sample distributions are different from person to person. This problem makes it difficult to train a universal recognition model for all users [13]. Second, for some mobile devices such as smartphones, the device locations are always unfixed. For the same activity, the sensing data obtained from different body locations are inconsistent [14]. It brings a great challenge to improve the performance by removing the influence of device locations [15]. To overcome these shortages, many researches try to incorporate supervised learning method with activity continuity which is independent of sample distribution.

In applying the activity continuity to the activity recognition, the transition matrix is the most important parameter. It describes the probabilities that a certain type of activity will hold itself and turn to others at the next time in a scene. This

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Xiang.

parameter is the basis for calculating the confidence of recognition results and smoothing the activity results sequence. Since this parameter depends on the application scenarios, it needs to be learned from the sequence of classification results of the activity in the scene. Currently, the Baum-Welch algorithm [16], [17] is the most commonly used algorithm, but it still has rooms for improvement.

B. RELATED WORKS

This section will briefly introduce current methods on applying activity continuity to activity recognition. Using activity continuity to assist activity recognition has always been a research issue in activity recognition researches. In early 2008, a computationally inexpensive methodology [18] for incorporating smoothing classification temporally was proposed, which can couple with any classifier with minimal training for classifying continuous sequences. The Hierarchical Support Vector Machine and Context-based Classification (HSVMCC) was proposed in [19] to recognize human activities when the sampling rate was less than the frequency of activities. These two methods utilize naive modification strategy, and do not consider the activity transition. The performance of these two methods are obviously worse as the activities change to another.

The other studies attempt to add this fact to sequence smoothing. The Activity Recognition Shell (ARShell) was proposed in [20], in which a Markov smoother was applied to post-process the results generated by the Google recognition service. The Lowest Cumulative Cost Activity Sequence (LCCAS) was put forward in [21], in which the similar policy was adopted to perform sequence smoothing. However, these two methods are all follow the straightforward schemes, which just obtain locally optimal solution [22]. Other researchers applied the HMM [23], [24] to provide global solutions. In [25], HMM was adopted to smooth out the accidental misclassifications generated by supervised learning schemes. The HMM and ensemble HMM were applied to recognize activities in [26] and [27] respectively. Because of its simplicity and effectiveness, the HMM is still the most popular model in current applications.

In the past several years, with the great success of deep learning in the fields of image processing and speech recognition, many researchers have applied deep learning model to activity recognition. To model the activity sequence, the sequence models of deep learning such as Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) were adopted in [28]–[31], and they got meaningful results. But the advantage of deep learning models depend on sufficient training data and powerful computing ability, which cannot be provided by the wearable devices [32]. Especially for the sequence models, as the activity transitions are different from scene to scene, these models always need to be trained independently for a specific scene. It is impractical for the wearable devices. Therefore, it is still a long way to go to apply sequence models of deep learning to activity recognition in practical applications [33]. On the other hand,

some applications such as the Google activity recognition service just provide the activity results but not the raw sensor data [20]. In these scenes, the deep learning models cannot take advantage of them. In summary, the classical models such as HMM are still the most commonly used models in current activity recognition systems, and the activity transition matrix is still the most crucial parameter for these classical models.

C. MOTIVATION

In the classical sequences models such as HMM, the activity transition matrix which describes the transition probabilities between different activities is the most important parameter. An accurate transition matrix can provide effective results on applying activity sequences to activity recognition. Currently, this parameter can be trained with the classical Baum-Welch algorithm [16], [17] which is a basic algorithm in HMM theory. But it still has its own shortage for sensor-based activity recognition. In Baum-Welch algorithm, the sequence of activity recognized labels should be the input. However, the recognized labels which acquired from most supervised learning algorithms always contain additional information which is the weights for these labels.

Taking a set containing two activities for example, one sample is recognized as (0.9,0.1) and the other one is (0.6, 0.4). The final labels of them are the same, but their classification weights are quite different. Low confidence indicates the low probability that the sample is correctly classified, while high confidence means the opposite. Thus, these two samples should provide different influences in training the transition matrix. But the classical Baum-Welch algorithm loses sight of this important feature. Fusing classification weights in learning transition matrix is a direction for improvement.

This paper proposes a weighted transition matrix learning algorithm named W-Trans which is the continuation of our previous works [22], [34]. The key concept of weighted observation probability described in the following section was first proposed in [34] to identify the confidence level of classification results. Then, this concept was applied to smooth the classification result sequence in order to improve the recognition accuracy in [22]. This paper focuses on the transition matrix learning issue on applying the concept of weighted observation probability. The main works of this paper are as follows.

- 1) The definitions of weighted observation probability are described, and the improved HMM named Weight Observation Hidden Markov Model (WOHMM) is introduced.
- 2) The recursive formula of transition matrix learning is derived, and the W-Trans for WOHMM is put forward.
- 3) The W-Trans is evaluated with metrics of cosine similarity and euler distance on two public data sets.
- 4) This algorithm is applied to the current post-process method and is verified to be effective.

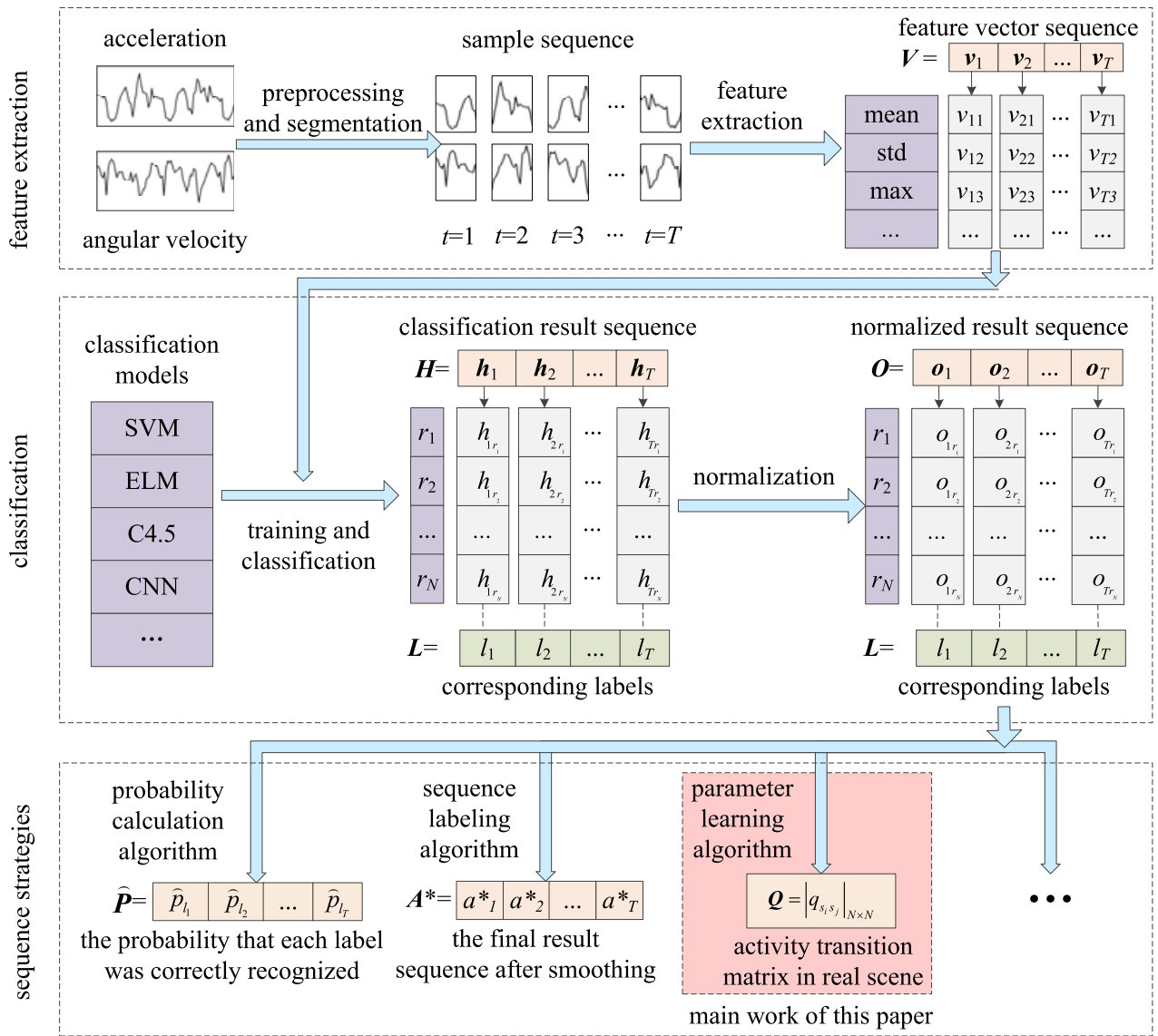


FIGURE 1. The activity recognition framework.

II. METHOD

This section first introduces a common framework of activity recognition, and point out the position of transition matrix learning in this framework. Then the formal description of WOHMM is given. Finally the W-Trans is put forward.

A. THE FRAMEWORK OF ACTIVITY RECOGNITION

To clearly introduce the purpose of this paper, we first introduce the common framework of activity recognition, shown in Fig. 1. As shown in Fig. 1, activity recognition is divided into three steps. In the first step, a variety of data including acceleration and angular velocity can be collected via the programming interfaces provided by wearable devices. These sensing data are divided into samples with fixed size, and the feature vectors are extracted from those samples. This step is summarized as feature extraction, shown as the upper

area in Fig. 1. In the second step, those feature vectors are identified by the trained classifier to generate a classification result sequence, which is H in the middle area. Generally, H is always normalized [35] with

$$o_{jr_k} = \frac{h_{jr_k} - \min(\mathbf{h}_j)}{\sum_{i=1}^N (h_{jr_i} - \min(\mathbf{h}_j))}, \tag{1}$$

where $\min(\mathbf{h}_j)$ is the minimum element in vector \mathbf{h}_j . Specially, some classification models such as Convolutional Neural Networks(CNN) would combine the first two steps into one step. After normalization, it can get the normalized result sequence O , and this sequence would be the input of sequence-based strategies shown as the lower area in Fig. 1.

In the sequence-based strategies, some sub-problems are concerned, such as identifying the probability that each label was correctly recognized, smoothing the result sequence to

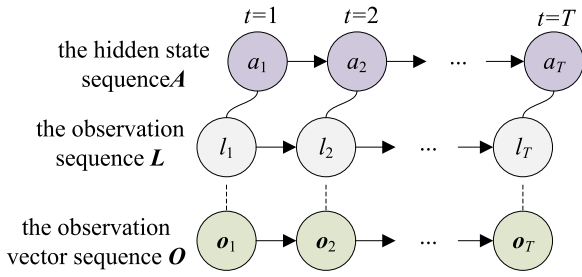


FIGURE 2. The WOHMM.

correct wrong labels, and so on. The bases of these sub-problems are obtaining the activity transition matrix in a specified scene. The classical Baum-Welch algorithm learns the transition matrix with the recognized label sequence L , and our proposed W-Trans will learn this parameter with the normalized result sequence O . The following subsections will give the basic definitions and detail our algorithm.

B. DEFINITIONS OF WOHMM

Before details, several basic definitions should be introduced. First, the state set and the observation set are denoted as $S = \{s_1, s_2, \dots, s_N\}$ and $R = \{r_1, r_2, \dots, r_M\}$ respectively, where N is the number of hidden states and M is the number of observations. Besides, the observation vector $\mathbf{o} = (o_{r_1}, o_{r_2}, \dots, o_{r_M})$ is the weight vector corresponding to R , where o_{r_i} is the weight that current hidden state is observed as r_i . For vector \mathbf{o} , the elements meet $0 \leq o_{r_i} \leq 1$ and $\sum_{i=1}^M o_{r_i} = 1$. Corresponding to the activity recognition, S and R are both the set of activities, and the observation vector \mathbf{o} illustrates the confidence values for all the possible activities which derived from classifications.

The WOHMM is shown in Fig. 2, which consists of three sequence $A = \{a_1, a_2, \dots, a_T\}$, $L = \{l_1, l_2, \dots, l_T\}$ and $O = \{o_1, o_2, \dots, o_T\}$. In these sequences, a_t , l_t and \mathbf{o}_t are the hidden state, observation and observation vector respectively at time t , and meet $a_t \in S$ and $l_t \in R$. For A , the hidden state may remain constant or transform to another as time goes on. The transition probabilities between different states constitute the transition matrix, written as

$$Q = |q_{s_i s_j}|_{N \times N}, \tag{2}$$

where $0 \leq q_{s_i s_j} \leq 1$ and $\sum_{j=1}^N q_{s_i s_j} = 1$. $q_{s_i s_j}$ represents the probability $P(a_{t+1} = s_j | a_t = s_i)$, and its subscript means the state at $t + 1$ is s_j on the condition that the state at t is s_i .

Corresponding to A , L is the observation sequence, in which l_i is the observation as $t = i$. Similar to HMM, the observation matrix is defined as

$$P = |p_{s_i r_j}|_{N \times M}, \tag{3}$$

where $0 \leq p_{s_i r_j} \leq 1$ and $\sum_{j=1}^M p_{s_i r_j} = 1$. $p_{s_i r_j}$ is the probability that the state s_i is observed as r_j , that is

$p_{s_i r_j} = P(l_t = r_j | a_t = s_i)$. In activity recognition, the observation matrix corresponds to the confusion matrix which can be obtained through experimental results.

According to the observation matrix, we define the weighted observation probability $\bar{p}_{s_i l_t} = P(l_t | a_t = s_i, \mathbf{o}_t)$, which is the probability that hidden state s_i observes l_t on the condition that given the corresponding observation vector $\mathbf{o}_t = (o_{tr_1}, o_{tr_2}, \dots, o_{tr_M})$. This probability can be calculated by the cosine similarity of $\mathbf{p}_{s_i} = (p_{s_i r_1}, p_{s_i r_2}, \dots, p_{s_i r_M})$ and \mathbf{o}_t , shown as

$$\bar{p}_{s_i l_t} = \frac{\mathbf{p}_{s_i} \cdot \mathbf{o}_t}{|\mathbf{p}_{s_i}| |\mathbf{o}_t|}. \tag{4}$$

The physical significance of $\bar{p}_{s_i l_t}$ is as follows. \mathbf{p}_{s_i} is the priori probability distribution, which describes the probabilities of different observations as the hidden state is s_i . But the actual observation probabilities are \mathbf{o}_t at time t . If \mathbf{p}_{s_i} and \mathbf{o}_t are highly similar, the probability that the hidden state is s_i would be high. In other words, as the actual observation vector is \mathbf{o}_t at time t , the probability that hidden state s_i observes l_t should be obviously high for this sample. Conversely, if the similarity between \mathbf{p}_{s_i} and \mathbf{o}_t is very low, the corresponding probability should be low. To fuse the observation vector, the $p_{s_i l_t}$ can be instead by $\bar{p}_{s_i l_t}$ in HMM, and we rename the new HMM as WOHMM.

Similar to the HMM, the initial vector $\boldsymbol{\pi}$ is defined as

$$\boldsymbol{\pi} = (\beta_{s_1}, \beta_{s_2}, \dots, \beta_{s_N})_{1 \times N}, \tag{5}$$

where $0 \leq \beta_{s_i} \leq 1$ and $\sum_{i=1}^N \beta_{s_i} = 1$. β_{s_i} is the probability that the hidden state is s_i at $t = 1$, and it equals to $P(a_1 = s_i)$.

To sum up, WOHMM is written as

$$\boldsymbol{\lambda} = [P, Q, \boldsymbol{\pi}], \tag{6}$$

which contains three parts of transition matrix Q , observation matrix P and initial state vector $\boldsymbol{\pi}$. As P and $\boldsymbol{\pi}$ can be obtained through experiments, our parameter learning problem is summarized as training the optimal transition matrix Q on given the observation sequence O and L . The following section will give the learning algorithm W-Trans.

C. OUR PARAMETER LEARNING ALGORITHM

Before the details of W-Trans, we introduce two variables.

1) The forward probability

$$\alpha_t(s_k) = P(\{l_1, l_2, \dots, l_t\}, a_t = s_k | \boldsymbol{\lambda}, \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t\})$$

which is the probability that the observation sequence between 1 and t is $\{l_1, l_2, \dots, l_t\}$ and the hidden state at time t is s_k on given the WOHMM parameter $\boldsymbol{\lambda}$ and observation vector sequence $\{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t\}$. This value can be obtained by

$$\alpha_t(s_k) = \begin{cases} \beta_{s_k} \bar{p}_{s_k l_t} & \text{if } t = 1 \\ \sum_{j=1}^N \alpha_{t-1}(s_j) q_{s_j s_k} \bar{p}_{s_k l_t} & \text{if } t > 1, \end{cases} \tag{7}$$

where $\bar{p}_{s_k l_t}$ is shown as Equation 4.

2) The backward probability

$$\beta_t(s_k) = P(\{l_{t+1}, \dots, l_T\} | a_t = s_k, \lambda, \{\mathbf{o}_{t+1}, \dots, \mathbf{o}_T\})$$

which is the probability that the observation sequence between $t + 1$ and T is $\{l_{t+1}, l_{t+2}, \dots, l_T\}$ on given λ , the hidden state s_k at time t , and the observation vector sequence $\{\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \dots, \mathbf{o}_T\}$. This value can be obtained by

$$\beta_t(s_k) = \begin{cases} 1 & \text{if } t \geq T \\ \sum_{j=1}^N q_{s_k s_j} \bar{p}_{s_j l_{t+1}} \beta_{t+1}(s_j) & \text{if } t < T, \end{cases} \quad (8)$$

where $\bar{p}_{s_j l_{t+1}}$ is shown as Equation 4.

Following the Baum-Welch algorithm, the logarithmic likelihood function of complete-data is $\log P(\mathbf{L}, \mathbf{A} | \lambda, \mathbf{O})$, which denoted as

$$\log P(\mathbf{L}, \mathbf{A} | \lambda, \mathbf{O}) = \beta_{a_1} \bar{p}_{a_1 l_1} q_{a_1 a_2} \bar{p}_{a_2 l_2} \dots q_{a_{T-1} a_T} \bar{p}_{a_T l_T}, \quad (9)$$

where \mathbf{A} is a possible hidden state sequence corresponding to \mathbf{L} and \mathbf{O} , λ is the WOHMM parameters. Construct the Q function of EM algorithm, shown as

$$Q(\lambda, \hat{\lambda}) = \sum_{\mathbf{A}} P(\mathbf{L}, \mathbf{A} | \hat{\lambda}, \mathbf{O}) \log P(\mathbf{L}, \mathbf{A} | \lambda, \mathbf{O}), \quad (10)$$

where λ is the parameters to be maximized, and $\hat{\lambda}$ is the current values. Substituting 9 into 10, we can get

$$Q(\lambda, \hat{\lambda}) = \sum_{\mathbf{A}} P(\mathbf{L}, \mathbf{A} | \hat{\lambda}, \mathbf{O}) \log \beta_{a_1} + \sum_{\mathbf{A}} P(\mathbf{L}, \mathbf{A} | \hat{\lambda}, \mathbf{O}) \left(\sum_{t=1}^T \log p_{a_t l_t} \right) + \sum_{\mathbf{A}} P(\mathbf{L}, \mathbf{A} | \hat{\lambda}, \mathbf{O}) \left(\sum_{t=1}^{T-1} \log q_{a_t a_{t+1}} \right), \quad (11)$$

which composes of three parts, the summation of initial vector, observation probability and transition probability respectively. As the initial vector and the observation matrix do not need be updated, the third part of Equation 11 should be maximized in order to maximize the Q function. Expand the third part and construct the lagrange function, expressed as

$$La = \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} P(\mathbf{L}, a_t = s_i, a_{t+1} = s_j | \hat{\lambda}, \mathbf{O}) \log q_{s_i s_j} + \gamma \left(\sum_{j=1}^N q_{s_i s_j} - 1 \right). \quad (12)$$

Calculate the partial derivative for $q_{s_i s_j}$, and let results be 0.

$$\sum_{t=1}^{T-1} P(\mathbf{L}, a_t = s_i, a_{t+1} = s_j | \hat{\lambda}, \mathbf{O}) + q_{s_i s_j} \gamma = 0 \quad (13)$$

Accumulate j and apply $\sum_{j=1}^N q_{s_i s_j} = 1$, we can get

$$\gamma = - \sum_{t=1}^{T-1} P(\mathbf{L}, a_t = s_i | \hat{\lambda}, \mathbf{O}). \quad (14)$$

Substitute 14 into 13.

$$q_{s_i s_j} = \frac{\sum_{t=1}^{T-1} P(\mathbf{L}, a_t = s_i, a_{t+1} = s_j | \hat{\lambda}, \mathbf{O})}{\sum_{t=1}^{T-1} P(\mathbf{L}, a_t = s_i | \hat{\lambda}, \mathbf{O})} \quad (15)$$

Decompose the numerator and denominator, shown as

$$\begin{aligned} P(\mathbf{L}, a_t = s_i, a_{t+1} = s_j | \hat{\lambda}, \mathbf{O}) &= P(\{l_1, \dots, l_t\}, a_t = s_i | \hat{\lambda}, \{\mathbf{o}_1, \dots, \mathbf{o}_t\}) \\ &\quad \times \hat{q}_{s_i s_j} \times \bar{p}_{s_j l_{t+1}} \\ &\quad \times P(\{l_{t+2}, \dots, l_T\} | a_{t+1} = s_j, \hat{\lambda}, \{\mathbf{o}_{t+2}, \dots, \mathbf{o}_T\}) \end{aligned} \quad (16)$$

and

$$\begin{aligned} P(\mathbf{L}, a_t = s_i | \hat{\lambda}, \mathbf{O}) &= P(\{l_1, \dots, l_t\}, a_t = s_i | \hat{\lambda}, \{\mathbf{o}_1, \dots, \mathbf{o}_t\}) \\ &\quad \times P(\{l_{t+1}, \dots, l_T\} | a_t = s_i, \hat{\lambda}, \{\mathbf{o}_{t+2}, \dots, \mathbf{o}_T\}). \end{aligned} \quad (17)$$

Substitute 7, 8, 16 and 17 into 15, we can get

$$q_{s_i s_j} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(s_i) \hat{q}_{s_i s_j} \hat{p}_{s_j l_{t+1}} \hat{\beta}_{t+1}(s_j)}{\sum_{t=1}^{T-1} \hat{\alpha}_t(s_i) \hat{\beta}_t(s_j)}, \quad (18)$$

which is the recurrence formula for transition probability learning. According to this formula, the W-Trans is shown as Algorithm 1.

In Algorithm 1, the first loop indicates iteration times, which depends on specific scenarios. For each iteration, a new Q is calculated based on the previous iteration. In this step,

Algorithm 1 W-Trans: The Weighted Transition Matrix Learning Algorithm for WOHMM

Input:

The observation sequence $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T\}$ and $\mathbf{L} = \{l_1, l_2, \dots, l_T\}$;

The initial transition matrix \hat{Q} ;

Output:

The updated transition matrix Q ;

- 1: **for** $n = 1, 2, \dots$ **do**
 - 2: **for** $i, j \in [1, N]$ **do**
 - 3: **for** $t = 1 : T - 1$ **do**
 - 4: $\hat{\alpha}_t(s_i) = \text{cal_alpha}(t, i, \hat{Q})$; //Equation 7
 - 5: $\hat{\beta}_t(s_j) = \text{cal_btea}(t, j, \hat{Q})$; //Equation 8
 - 6: **end for**
 - 7: $q_{s_i s_j} = \text{cal_new}(\hat{\alpha}_t(s_i), \hat{\beta}_t(s_j), \hat{Q})$; //Equation 18
 - 8: **end for**
 - 9: $\hat{Q} = Q$
 - 10: **end for**
 - 11: **return** Q
-

the forward probability and backward probability are first obtained with Equation 7 and 8. Then the updated transition probability is calculated by Equation 18. All the transition probabilities compose the new transition matrix \hat{Q} . After iterating many times, the final transition matrix is returned.

III. EXPERIMENTS

In this section, we compare W-Trans with the classic Baum-Welch algorithm on two public data sets. First, it introduces the design of experiments and then gives the final experimental results for different classifier on two data sets. Finally, it discusses the influences of algorithm parameters.

A. EXPERIMENTAL DESIGN

First of all, the SARD [36] and HAPT [37] are selected as the data sets in our experiments. Both of them contain the acceleration and gyroscope data provided by the smartphones. SARD is the product of Twente University, while HAPT is shared by UCI Machine Learning Repository. Five popular daily activities including walking, running, upstairs, downstairs and standing were chosen for SARD and six daily activities such as walking, upstairs, downstairs, sitting, standing and lying were considered for HAPT. Secondly, we divided those sensing data into fixed-size samples by the half-overlapping sliding window. As the window size was set to 1 second, we obtained 27000 and 11000 samples respectively from SARD and HAPT. Finally, we extracted features with different methods for each data set. For SARD, we extracted the features such as mean, variance, average cross rate, maximum, minimum and 10 low frequency coefficients of Fast Fourier Transformation (FFT) [38]. Since HAPT has provided 561 features, we only selected the top 30 ones according to their scores by ReliefF algorithm [39].

Before simulating Algorithm 1, we recognized the activities with the supervised learning method [14]. The samples of each user were selected as the test set in turn, and the samples of other users constitute the training set. We selected two popular classifiers of SVM and ELM to get the recognized label and the corresponding vector for each sample. Then we randomly arranged the sample orders to reconstruct different

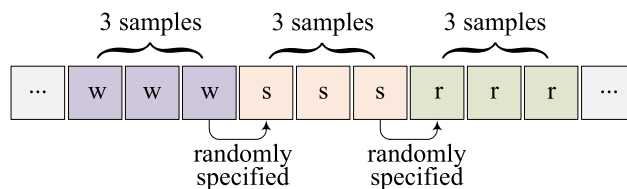


FIGURE 3. An example of activity sequence reconstruction.

activity sequences. In this step, the duration of each activity was set to a specified value n , which denotes the activity will last n samples as it occurs. Besides, the transitions between different activities were randomly specified. An example is shown as Fig. 3, in which the activity duration is set to 3. In this example, w denotes walking, s indicates standing, and r represents running. As shown in Fig. 3, as activity of walking occur, it lasts 3 sample. Then, the activity randomly changes to standing, and it also lasts 3 samples. Next, it randomly changes to running.

According to the reconstruction, we could simulate the activity sequences in different scenarios. After reconstruction, the diagonal elements of actual transition matrix would be $\frac{n-1}{n}$ and the other elements would be close to $\frac{1}{n(n-1)}$, where m is the number of activities. The reconstructed sequence would be the input of parameter learning methods. In simulating Algorithm 1, the initial state vector was set to follow the uniform distribution. The observation matrices corresponding to SARD and HAPT are described in Table 1 and 2 respectively, which were built according to the confusion matrices for activity recognition. Due to different numbers of default activities in SARD and HAPT, the sizes of matrices are unequal.

To prove the advantages of our algorithm, we should choose metrics to evaluate the similarity between the learned transition matrix and the actual one. As the cosine similarity and the euler distance are the most popular similarity tools, we use these two methods as our metrics in the following experiments. The cosine similarity is defined as

$$d_{cosine} = \frac{\sum_{ij} q_{ij} \hat{q}_{ij}}{\|Q\|_F \|\hat{Q}\|_F}, \tag{19}$$

TABLE 1. Observation matrix of SARD.

	Walking	Standing	Running	Going Upstairs	Going Downstairs
Walking	0.82	0	0	0.11	0.07
Standing	0	0.99	0	0.01	0
Running	0	0	0.97	0	0.03
Going Upstairs	0.1	0	0	0.84	0.06
Going Downstairs	0.1	0	0.02	0.11	0.77

TABLE 2. Observation matrix of HAPT.

	Walking	Going Upstairs	Going Downstairs	Sitting	Standing	Lying
Walking	0.94	0.02	0.05	0	0	0
Going Upstairs	0.09	0.89	0.02	0	0	0
Going Dostairs	0.05	0.13	0.82	0	0	0
Sitting	0	0	0	0.66	0.34	0
Standing	0	0	0	0.19	0.81	0
Lying	0	0	0	0.01	0	0.99

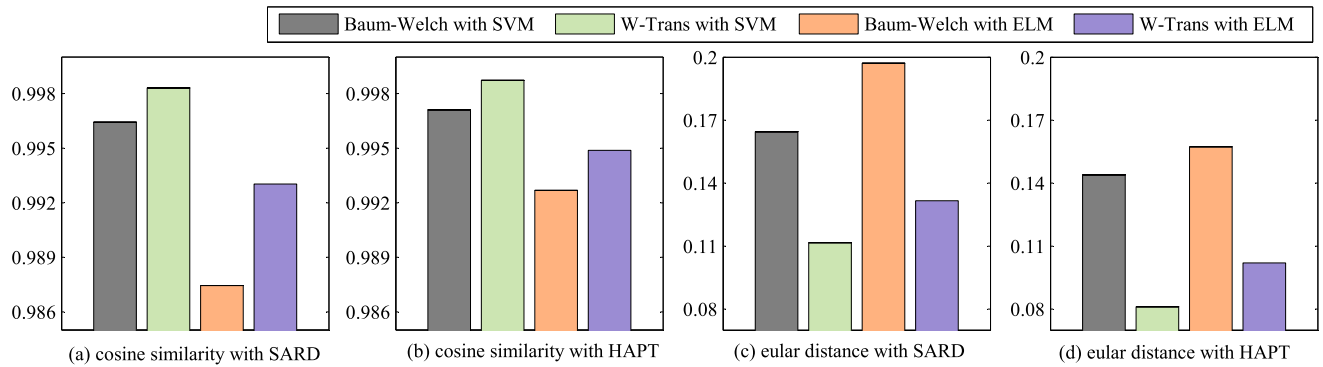


FIGURE 4. Mean results with different metrics and data sets.

where \hat{q}_{ij} and q_{ij} are the learned and actual transition probabilities respectively, $\|Q\|_F$ is the Frobenius norm of matrix Q . The euler distance is defined as

$$d_{euler} = \sqrt{\sum_{i,j} (\hat{q}_{ij} - q_{ij})^2}, \quad (20)$$

These two metrics evaluate the results in different views. For the cosine similarity, high values indicate good results, and low values instead for the euler distance.

For example, Q is the actual transition matrix which derived from sequence reconstruction, Q_1 and Q_2 are learned by different algorithms respectively, which denoted by Alg1 and Alg2. The $d_{cosine}(Q, Q_1)$, $d_{cosine}(Q, Q_2)$, $d_{eular}(Q, Q_1)$ and $d_{eular}(Q, Q_2)$ can be calculated. If $d_{cosine}(Q, Q_1) > d_{cosine}(Q, Q_2)$, Alg1 would be better. But if $d_{eular}(Q, Q_1) > d_{eular}(Q, Q_2)$, Alg2 would be better. The following subsections will show the comparison results based on these two indicators.

B. COMPARISON RESULTS

As the transition matrix is a parameter of HMM, the experiments just compared Algorithm 1 with Baum-Welch algorithm [16]. Fig. 4 shows mean results with different metrics and data sets, where W-Trans is our method and Baum-Welch is the original algorithm. The iteration time was set to 10, and the activity duration was set to 5 for these results. The selection of these two parameters will be introduced in the following sections. Fig. 4 includes four figures. The first two figures show comparison results of the cosine similarity with different data sets, and the following two figures illustrate results of the euler distance. For each figure, it shows four results, which are derived from different parameter learning algorithms and classifiers.

As shown in Fig. 4(a) and (b), the W-Trans provides obviously higher cosine similarity than Baum-Welch algorithm whatever the classifiers. It means the transition matrix learned by our method is closer to the actual one. For the following two subfigures, the W-Trans gives obviously lower euler distance. As lower euler distance means better result, our method gives better performance.

Comparing the results derived from different classifiers, we can know that SVM performs better than ELM whatever the parameter learning algorithms. According to analyzing the classification results, we find that the classification accuracy of SVM is slightly better than ELM. It causes that the incorrect results in classification result sequence of SVM are less than ELM. In parameter learning, these misclassified results offer more obvious influences to the final transition matrix. Thus, the performance of ELM would be worse. This comparison reveals that the activity classification accuracy is an import influencing factor to the final transition matrix.

Comparing the results evaluated on different data sets, we can find that the data set of HAPT provide better performance. The reasons are described as follows. By observing Table 2, we can know activities included in HAPT can be divided into two groups. The first group contains Walking, Upstairs and Downstairs, and the second group consists of the rest activities. The activity confusion probabilities between these two groups are 0, which causes the activities in one group would not be recognized as activities of another group in the recognition result sequence. In parameter learning, the activity transition probability between groups would be easy to be determined. The factors that affect the final transition matrix are just the confusion matrix within the group, which is a simplified problem. It also reveals that simplifying the confusion matrix is an important direction to improve the algorithm in practical applications.

C. INFLUENCE OF ITERATION TIME

In Algorithm 1, the iteration time is an import parameter affecting performance. Section III-B set this parameter to 10. This subsection will discuss how this value is obtained. In the process of simulation, we calculate the cosine similarity and the euler distance for each iterate. As the SVM and ELM provide similar results, this subsection just introduces the results classified by SVM. Fig. 5 shows the experimental results as the iteration time changes from 1 to 15. The activity duration is 5 for these results. As shown in Fig. 5, the euler distance declines and the cosine similarity improves for these two

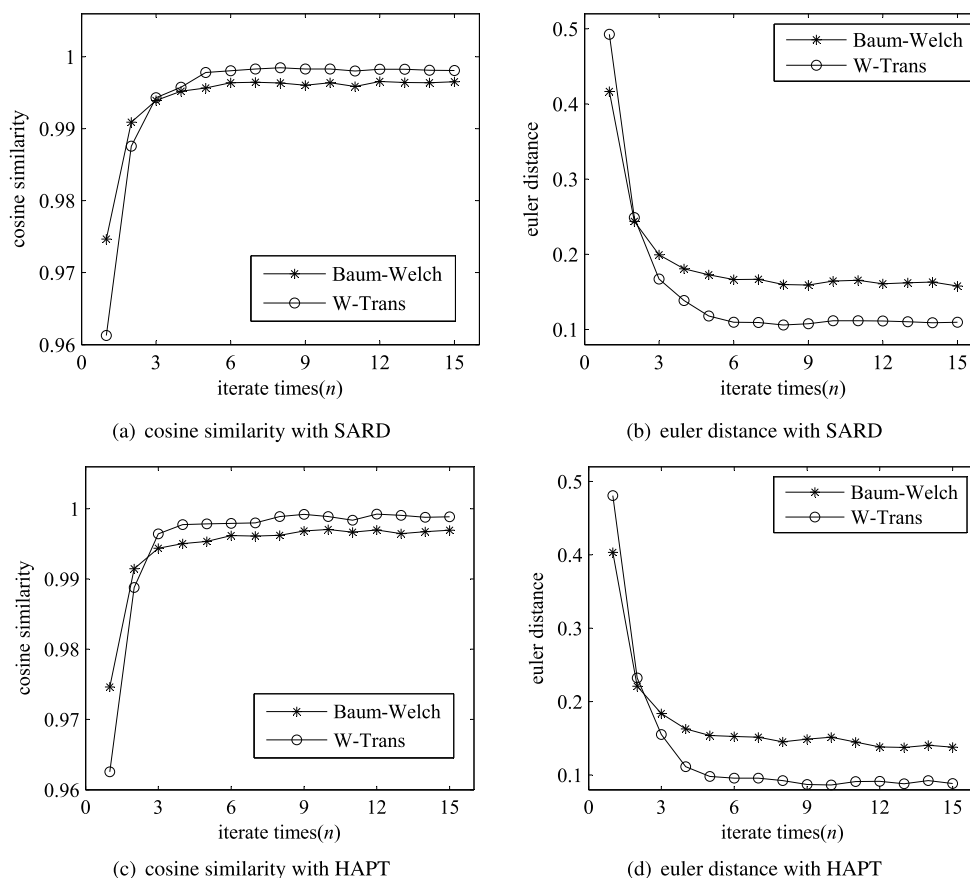


FIGURE 5. Comparison results as the iteration time changes from 1 to 15.

methods as the iteration time increases. When the iteration time is smaller than 5, these curves change obviously, and then become gentle. It is because that the learning process becomes convergent and the transition matrix is gradually approaching the optimal matrix.

When the iteration time is 1 or 2, the W-Trans performs worse than the Baum-Welch algorithm, and then it becomes better. The reasons are as follows. After replacing the original observation probability with the weighted observation probability, the differences between probabilities which observed by the same hidden state become small. This feature leads to the convergence rate becomes slow. So the performance is worse when iterating few times. It is because of slow convergence rate, the W-Trans can get more accurate result.

Comparing the results simulated on different data sets, we can find that the results on HAPT converge faster. For the SARD, the transition matrix converges after about 6 iterate times, and it is 4 for HAPT. The reasons are also simple. As the observation matrix of HAPT is divided into two groups, the transition matrix learning of HAPT is actually divided into two sub-problems. Each sub-problem contains three activities. Obviously, the parameter learning problem of 3 activities is faster than 6 activities.

D. INFLUENCE OF ACTIVITY DURATION

In the above sections, activity sequences were randomly reconstructed, in which the duration of each activity was set to a specified value and the transition between different activities was randomly specified. In this step, activity duration is an important parameter affecting the final results. This section will discuss the influence of activity duration. Fig. 6 shows the comparison results as the activity duration changes from 2 to 15 on SARD data set. As the results derived from two classifiers are similar, these figures just show the results of SVM.

As shown in Fig. 6, the learned transition matrix of activity sequence is approaching the optimal value as the activity duration increases. The main reasons are as follows. When the activity duration is small, the activity transitions frequently. Learning the transition matrix in the result sequence is susceptible to the incorrect recognition results. As the number of continuous samples increases, the transition law between activities becomes obvious, and it is also easy to be captured from the activity result sequence. In contrast, the influence of the error classification results become small.

Comparing the results of the two algorithms in Fig. 6, it can be found that when the activity duration is 2, the performance of W-Trans is slightly worse than Baum-Welch. With the

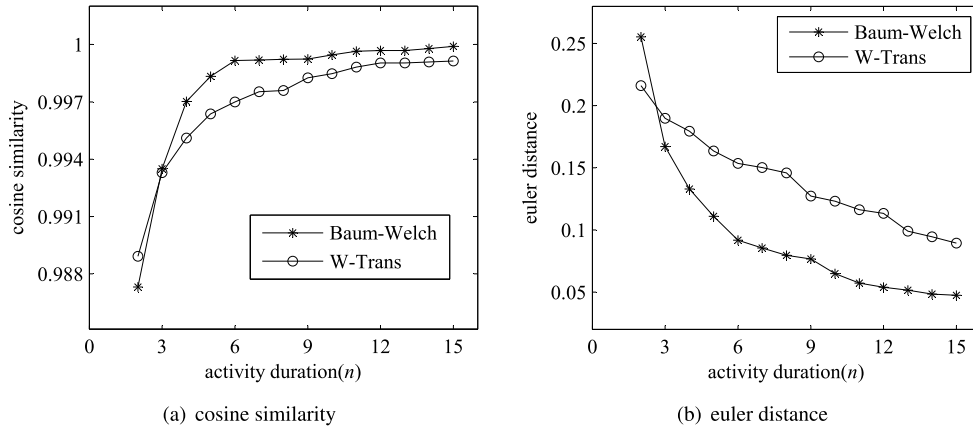


FIGURE 6. Comparison results as the activity duration changes from 2 to 15.

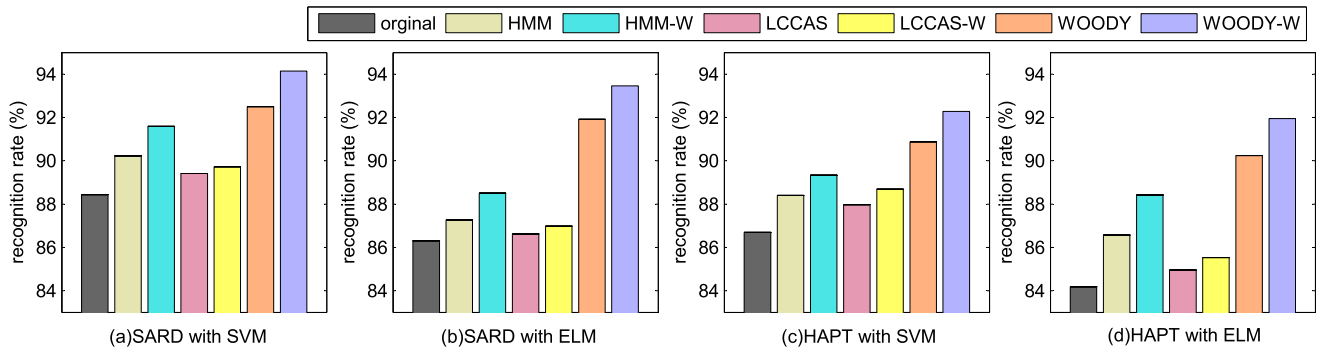


FIGURE 7. The recognition rates of post-process methods with different data sets and classifiers.

increase of activity duration, the performance of W-Trans is gradually better than that of Baum-Welch. The main reasons are as follows. Since the difference of weighted observation probability for different hidden states is small, W-Trans is more susceptible to misclassified results when the number of continuous samples is small. It causes that the W-Trans performs worse. As the activity duration increases, the effect of misclassified results becomes small, and the results of W-Trans also become better. However, parameter learning algorithm always requires a long sequence of recognition results. For a long activity sequence, activity duration is 2 means one person changes his activity in about 2 seconds. Obviously, it is impossible in practical applications. Therefore, it has little effect on the activity recognition in real scenes.

IV. APPLICATION VERIFICATION

The performance of W-Trans has been evaluated in the previous section, which compared to the classical transition matrix learning algorithm under metrics of cosine similarity and euler distance. To verify the validity of the method further, this section will apply W-Trans to activity post-process which is an import issue in sensor-based activity recognition. Current popular post-process methods include WOODY [22],

HMM [40], CRF [41] and LCCAS [21] etc. In these methods, three methods of WOODY, HMM, and LCCAS are all regard the activity transition matrix as a basic parameter, which is learned by the classical Baum-Welch algorithm in these methods. This section will evaluate these post-process methods in replacing the Baum-Welch algorithm with W-Trans.

In this section, the SARD and HAPT are also selected as the experimental data sets. The experimental designs of data preprocessing, feature extraction, training and classification, sequence reconstruction are the same as Section III-A. In this section, activity duration is set to 5. After getting the normalized recognition result sequence, we first learn the activity transition matrix with Baum-Welch algorithm and W-Trans respectively. Then the learned transition matrix and the activity result sequence are as inputs of post-process algorithms. Fig. 7 shows the average recognition rates of seven post-process methods with respect to different classifiers and data sets.

Fig. 7 shows four sub-figures. Each sub-figure plots the recognition rates with different datasets and classifiers. For each sub-figure, it contains seven methods. Original represents the classification without any post-process. HMM, LCCAS and WOODY are the methods in [21], [40] and [22] respectively. These three methods are included in our

experiments as baselines, in which the Baum-Welch algorithm is applied as the transition matrix learning algorithm. The following three methods of HMM-W, LCCAS-W, WODDY-W are the improved post-process methods, in which the transition matrix learning algorithm replaced by W-Trans.

Comparing each post-process method and the corresponding improved method, we can find that the activity recognition rates are all improved. The best result is WOODY-W in Fig. 7(d), which increases the recognition rate by 1.85%. Even for the worst result of LCCAS-W in Fig. 7(a), it also improves the recognition rate by 0.31%. The reasons are obvious. The transition matrix is a most important parameter in post-process. It describes the transition relationship between activities. Accurate activity transition matrix can decrease the modifying error than inaccurate one in estimating the result probability. It is bound to increase the recognition rates. These results reveal that the activity recognition rate improves steadily after replacing Baum-Welch algorithm with our approach in post-process methods.

V. CONCLUSION

The sensor-based human activity recognition has been widely applied in behavior tracking, health monitoring, indoor localization etc. Applying the activity continuity to assist activity recognition is an important research issue, in which the HMM is a commonly used theoretical tool because of its simplicity and effectiveness. As the most important parameter, the transition matrix should be learned from the classification results sequence dynamically.

Aiming at the problem that the classical transition matrix learning algorithm cannot utilize the weights of classification results, an improved HMM named WOHMM is introduced in this paper. Based on the definitions of WOHMM, the weighted transition matrix learning algorithm W-Trans is proposed. In the evaluation, the public data sets are applied to activity recognition and transition matrix learning, and the metrics of cosine similarity and eular distance are adopted for measuring the performance of different methods. The experimental results show that our algorithm outperforms the classical Baum-Welch algorithm.

There are still some future research directions based on our method. In the actual scene, the sample distribution in the activity sequence is often unbalanced. For example, a certain type of activity such as sitting always lasts a long time, while other activities such as running and upstairs are relatively short. The transition matrix learning algorithm in this paper does not consider this subject, whose effect to the proposed algorithm need to be studied in the future researches. On the other hand, the proposed W-Trans in this paper is a universal algorithm, which can be applied not only to sensor-based activity recognition. Other sequence related problems with observation weight can also use this algorithm to learn the transition matrix. Applying this algorithm to other fields is also worth studying.

REFERENCES

- [1] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.
- [2] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, "A survey on approaches of motion mode recognition using sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1662–1686, Jul. 2017.
- [3] S. Scheurer, S. Tedesco, O. Manzano, K. N. Brown, and B. O'Flynn, "Monitoring emergency first responders' activities via gradient boosting and inertial sensor data," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Dublin, Ireland: Springer, 2018, pp. 691–694.
- [4] V. Varadharajan, U. Tupakula, and K. Karmakar, "Secure monitoring of patients with wandering behavior in hospital environments," *IEEE Access*, vol. 6, pp. 11523–11533, 2018.
- [5] L. Li, R. Bai, B. Xie, Y. Peng, A. Wang, W. Wang, B. Jiang, J. Liang, and X. Chen, "R&P: An low-cost device-free activity recognition for E-health," *IEEE Access*, vol. 6, pp. 81–90, 2018.
- [6] F. D. Guillén-Gámez and M. J. Mayorga-Fernández, "Empirical study based on the perceptions of patients and relatives about the acceptance of wearable devices to improve their health and prevent possible diseases," *Mobile Inf. Syst.*, vol. 2019, pp. 1–12, Apr. 2019.
- [7] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.
- [8] A. Poullose, O. S. Eyobu, and D. S. Han, "An indoor position-estimation algorithm using smartphone IMU sensor data," *IEEE Access*, vol. 7, pp. 11165–11177, 2019.
- [9] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, Jan. 2014.
- [10] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Gener. Comput. Syst.*, vol. 81, pp. 307–313, Apr. 2018.
- [11] Y. Zhang, M. Haghdan, and K. S. Xu, "Unsupervised motion artifact detection in wrist-measured electrodermal activity data," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2017, pp. 54–57.
- [12] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognit.*, vol. 78, pp. 277–290, Jun. 2018.
- [13] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervas. Mobile Comput.*, vol. 57, pp. 1–13, Jul. 2019.
- [14] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, and K. Tang, "Kernel fusion based extreme learning machine for cross-location activity recognition," *Inf. Fusion*, vol. 37, pp. 1–9, Sep. 2017.
- [15] M. U. S. Khan, A. Abbas, M. Ali, M. Jawad, S. U. Khan, K. Li, and A. Y. Zomaya, "On the correlation of sensor location and human activity recognition in body area networks (BANs)," *IEEE Syst. J.*, vol. 12, no. 1, pp. 82–91, Mar. 2018.
- [16] L. R. Welch, "Hidden Markov models and the Baum-Welch algorithm," *IEEE Inf. Theory Soc. Newslett.*, vol. 53, no. 4, pp. 10–13, Dec. 2003.
- [17] F. Yang, S. Balakrishnan, and M. J. Wainwright, "Statistical and computational guarantees for the baum-welch algorithm," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 4528–4580, 2017.
- [18] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 3337–3340.
- [19] L. Zheng, D. Wu, X. Ruan, S. Weng, A. Peng, B. Tang, H. Lu, H. Shi, and H. Zheng, "A novel energy-efficient approach for human activity recognition," *Sensors*, vol. 17, no. 9, p. 2064, 2017.
- [20] M. Zhong, J. Wen, P. Hu, and J. Indulska, "Advancing Android activity recognition service with Markov smoother: Practical solutions," *Pervas. Mobile Comput.*, vol. 38, pp. 60–76, Jul. 2017.
- [21] R. Younes, T. L. Martin, and M. Jones, "Activity classification at a higher level: What to do after the classifier does its best?" in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2015, pp. 83–86.
- [22] C. Wang, Y. Xu, H. Liang, W. Huang, and L. Zhang, "WOODY: A post-process method for smartphone-based activity recognition," *IEEE Access*, vol. 6, pp. 49611–49625, 2018.
- [23] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

- [24] S. M. E. Sahraeian and B.-J. Yoon, "A novel low-complexity HMM similarity measure," *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 87–90, Feb. 2011.
- [25] J. Wen and Z. Wang, "Learning general model for activity recognition with limited labelled data," *Expert Syst. Appl.*, vol. 74, pp. 19–28, May 2017.
- [26] C. Wang and X. Wang, "Daily activity recognition based on triaxial accelerometer of elderly people," *Acta Electronica Sinica*, vol. 45, no. 3, pp. 570–576, 2017.
- [27] Y.-J. Kim, B.-N. Kang, and D. Kim, "Hidden Markov model ensemble for activity recognition using tri-axis accelerometer," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 3036–3041.
- [28] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognit.*, vol. 76, pp. 80–94, Apr. 2018.
- [29] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artif. Life Robot.*, vol. 23, no. 2, pp. 173–185, Jun. 2018.
- [30] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, Jun. 2017.
- [31] A. Ullah, K. Muhammad, J. Del Ser, S. W. Baik, and V. H. C. de Albuquerque, "Activity recognition using temporal optical flow convolutional features and multilayer LSTM," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9692–9702, Dec. 2019.
- [32] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, Sep. 2018.
- [33] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, 2019.
- [34] C. Wang, Y. Xu, J. Zhang, and W. Yu, "SW-HMM: A method for evaluating confidence of smartphone-based activity recognition," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 2086–2091.
- [35] Y. Chen, Z. Zhao, S. Wang, and Z. Chen, "Extreme learning machine-based device displacement free activity recognition model," *Soft Comput.*, vol. 16, no. 9, pp. 1617–1625, Sep. 2012.
- [36] M. Shoaitb, H. Scholten, and P. J. M. Havinga, "Towards physical activity recognition using smartphone sensors," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput. IEEE 10th Int. Conf. Autonomic Trusted Comput.*, Dec. 2013, pp. 80–87.
- [37] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. Esann*, 2013, pp. 1–6.
- [38] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. Joint Conf. Smart Objects Ambient Intell. Innov. Context-Aware Services, Usages Technol. (sOc-EUSAI)*, 2005, pp. 159–163.
- [39] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.
- [40] J. Wen, M. Zhong, and J. Indulska, "Creating general model for activity recognition with minimum labelled data," in *Proc. ACM Int. Symp. Wearable Comput.*, 2015, pp. 87–90.
- [41] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2014, pp. 51–59.



BO WANG was born in Zhoukou, Henan, China, in 1987. He received the Ph.D. degree from Xi'an Jiaotong University, in 2017. He has been working with Zhengzhou University of Light Industry, since 2017. His research interests include mobile computing, activity recognition, and machine learning.



HUI LIANG received the bachelor's and master's degrees in computer science from the China University of Mining and Technology and the Ph.D. degree from the Communication University of China. He is currently a Professor with the Software Engineer College, Zhengzhou University of Light Industry. He had been a Marie Curie Research Fellow with the National Centre for Computer Animation, Bournemouth University, U.K. His current research interests include wearable computing, computer animation, intelligent data management, and software engineering.



JIANZHONG ZHANG was born in Shijiazhuang, Hebei, China. He received the Ph.D. degree with Nankai University, Tianjin, China, in 2008. He is currently a Professor of the College of Cyber Science, Nankai University. His research interests include mobile computing, activity recognition, and network security.



WANWEI HUANG was born in Yancheng, Jiangsu, China. He received the Ph.D. degree from the National Digital Switching System Engineering and Technological Research and Development Center, in 2009, Zhengzhou, China. He is currently working with the Zhengzhou University of Light Industry. His research interests include next-generation network architecture, network security, and reconfigurable computing.



CHANGHAI WANG was born in Liaocheng, Shandong, China, in 1987. He received the M.S. degree in computer science and technology from Jiangsu University, in 2012, and the Ph.D. degree from Nankai University, in 2016. He has been working with Zhengzhou University of Light Industry, since 2017. In the last five years, he has published nine articles related to activity recognition, and received the Best Paper Award of the 10th IEEE Asia-Pacific Services Computing Conference, and the Excellent Paper Award of *Journal on Communications*, in 2016. His research interests include activity recognition, wearable computing, and mobile computing.



WANGWEI ZHANG was born in Yuncheng, Shanxi, China. He received the Ph.D. degree from the Army Engineering University, Shijiazhuang, China. He has been working with the Zhengzhou University of Light Industry, since 2018. His research interests include data simulation, missile control, and launch theory and technology.

...