Chapter 41
Activity Recognition Based on Latent Knowledge Mining in Smart Home

Yu Tong, Rong Chen, and Bo Yu

Abstract Activity recognition in smart home is one pattern recognition problem. Many activity recognition algorithms have appeared so far to recognize activities in smart home. Past researches have proved that dynamic and deep knowledge mining algorithms will help improve the accuracy. But because of the uncertainty of sensors and the complexity of the user activities, existing activity recognition methods still have a lot of room for improvement. Considering there is some latent knowledge existed in sensors or user activities, this paper proposes to recognize activities by exploring latent knowledge. Firstly, this paper improves activity recognition by extracting latent knowledge between sensors and activities, thereby proposed one feature preprocessing method. Then, it proves one new multi-resident activity recognition method based on latent knowledge in multi-resident activities. Simulations conclude that extracting latent knowledge can greatly enhance activity recognition.

41.1 Introduction

Non-invasive activity recognition (NAR) is an ambient intelligence technology which can recognize activities based on non-invasive sensors without affecting the living conditions of residents. NAR has an important application in the field of smart home that can understand individual behavior, group behavior, and the interaction between people and the environment. Over the past decade, most of the previous NAR methods are based on pattern recognition methods and have evolved from static algorithms to

Y. Tong (✉)
School of Computer Science and Technology, Hefei Normal University, 230601 Hefei, China
e-mail: tongyu24@126.com

R. Chen
College of Information Science and Technology, Dalian Maritime University, 116026 Dalian, China

B. Yu
School of Civil Engineering, Hefei University of Technology, 230009 Hefei, China

S.-L. Peng et al. (eds.), *Sensor Networks and Signal Processing*, Smart Innovation, Systems and Technologies 176, https://doi.org/10.1007/978-981-15-4917-5_41

dynamic algorithms, from simple feature representation to multi-level deep knowledge mining algorithms [1], such as support vector machine [2], Naive Bayes [3], hidden Markov models [4], latent-dynamic conditional random fields [5] and deep learning [6]. Past research has proved dynamic activity recognition algorithms and deep knowledge mining algorithms will help improve the accuracy of NAR [5, 7]. Based on these considerations, this paper will enhance NAR based on dynamic activity recognition algorithms by mining latent knowledge that exists in sensors and activities.

On one hand, with low-cost sensors and wireless sensor networks development, various passive sensors have recently been used to recognize activities [8, 9] in non-invasive smart home. For example, motion sensors which are installed in the floor can capture human motion and the RFID tags which are attached to the object can capture human-to-environment interaction. However, observed feature dimension increases with the increase of sensors and causes the higher computational complexity. The more features are used, the higher computational complexity it will cause, while modeling fewer features is often insufficient to ensure recognition accuracy. Besides, activity observation feature which composed of sensor observation is often abundant and sometime redundant. Thus, the feature selection directly contributes to the performance of the recognition model. Principal component analysis (PCA) [10] is tested for feature generation, but the algorithms need to choose appropriate principal component number and number selection is an impact on the result. Fortunately, there is a lot of latent knowledge in sensor networks, such as multiple sensors that are often related to only one activity. It will help NAR if we can mine the latent knowledge between sensors and activities.

On the other hand, the activities in activity recognition are not only the activities of a single person, but also the activities of multiple residents. Multi-resident activity recognition (MRAR) is more difficult due to user activity interfering with each other. In a smart home with non-obtrusive sensors, MRAR often uses data association [4, 11] which associates sensor data to the person who triggered the sensor or changed the value. To improve the MRAR accuracy, dynamic Bayesian networks such as CHMM and FCRF often used to model interacting process [12, 13]. However, data associations are often unknown and hard to obtain in ubiquitous sensor environment. Beside, for multiple residents in smart home with non-obtrusive sensors, who triggered the sensor is often ambiguous and there are not strong underlying data associations to use. If the data association is incorrect, the MRAR will be correspondingly inaccurate. So, it would be interesting to find a method for MRAR that does not rely on data association. Fortunately, there is some latent knowledge which is often invariant in multi-resident environment. For instance, there are some global features and trends, playing chess collaboratively, only one person can use computer at the same time since there is only one computer. The latent knowledge is often easy to represent in multi-resident environment. If we can mine the latent knowledge well, multi-resident activity recognition will be improved.

The paper is organized like this, in Sect. 41.2, it will introduce one activity recognition method by extracting latent knowledge between sensors and activities. Then, Sect. 41.3 will prove one new multi-resident activity recognition method based on latent knowledge in multi-resident activities. Section 41.4 is validation. Finally, the article concludes with some conclusions.

41.2 Latent Knowledge Between Sensors and Activities

Motivated by the relationship between sensors and activities, this section will combine the multiple features that related only one activity as one feature and use CRF to recognize activities in smart homes. To describe our method, we start from analyzing the relationship between sensor data and activities.

The activity observation feature vector at time t is often denoted as $\mathbf{x}_t = \left(x_t^1, x_t^2, \ldots, x_t^N\right)$, where N is the dimension of observation feature. By considering one sensor as one observation feature, the dimension of observation feature vector will equal to the total sensor number. However, it is common that some sensors are related to only one activity and some sensor states are related to more than one activity. Figure 41.1 is the relationship between sensors and activities. In the figure, the sensors that related only one activity are denoted as "●," the sensors that related two activities are denoted as "■," and the sensors that related more than two activities are denoted as "▲." When the state of sensor data that relates only one activity and does not relate other activities changed, it is easy to deduce that the related activity rather than other activities is being carried out. In addition, it is also common that several sensors related only one activity and does not relate other activities ("●" that in dashed circle). When one or several states of those sensors changed, we can deduce the related activity is being carried out.

If we regard the relationship between sensors and activities as latent knowledge and the sensors observation that related only one activity and do not relate other activities as one combined observation feature, we can deduce the observation feature vector $\left(x_t^1, x_t^2, \ldots, x_t^N\right)$ to $\left(x_t^1, x_t^2, \ldots, x_t^L\right)$,where N and L is the observation feature

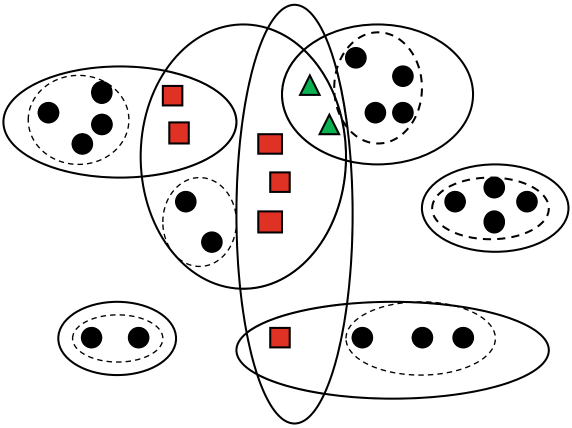


Fig. 41.1 Relationship between sensors and activities

dimension before and after feature combining. The feature combining method is shown in Algorithm 1.

Algorithm 1. Feature combining method

```
Input: Observation feature $x0$:
$\{\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^{\mathrm{N}}),\ t = 1, \ldots, \mathrm{T0}\}$,
activity labels $y0$:
$\{y_t, t = 1, \ldots, \mathrm{T0}\}$
Output: Combined feature $X0$:
$\{\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^{\mathrm{L}}), t = 1, \ldots, \mathrm{T0}\}$
1. Find the number of activities $\mathrm{Na} = \max\{\boldsymbol{y}0\}$ and the number of sensors $\mathrm{Ns} = \mathrm{N}$;
2. Find the sensors that related to activity i and put them in sensor set $S_i = \{s_{\mathrm{i}}\}$, $i = 1, \ldots, \mathrm{Na}$;
3. For every $s_i$ in $S_i$, put it to set $C_i$ if $s_i$ does not appear in other set $S_j, j \neq i, i, j = 1, \ldots, \mathrm{Na}$;
4. Combined the sensors in $C_i$ as one combined feature;
5. For every observation feature $\boldsymbol{x}_t$ in $\boldsymbol{x}0$, update the features that corresponded sensors in $C_i$
   with the combined feature, $i = 1, \ldots, \mathrm{Na}$;
6. Denote the updated $\boldsymbol{x}0$ as $\boldsymbol{X}0$;
7. Return $\boldsymbol{X}0$.
```

41.3 Latent Knowledge in Multi-resident Activities

MRAR is to infer multi-resident activities form observations. Multi-resident activity sequence is often denoted as $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{\mathrm{T}}\}$ and observation is often denoted as$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{\mathrm{T}}\}$. For $t = 1, 2, \ldots, \mathrm{T}$, $\boldsymbol{y}_t$ represents multi-resident activities at time t, and $\boldsymbol{x}_t$ represents sensor observation vectors at time t. Both $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ are multi-dimensional variables, where the dimension of $\boldsymbol{y}_t$ is the number of residents, and the dimension of $\boldsymbol{x}_t$ is the number of observation feature. MRAR with machine learning method often needs some empirical data to train a recognition model, where empirical data are often used as training samples $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_{\mathrm{T0}}, \boldsymbol{y}_{\mathrm{T0}})\}$.To better illustrate the problem, this paper will give a multi-resident scenario below.

Scenario: Two residents (ID = 1 and ID = 2) randomly perform three daily activities in one smart home. The three activities are labeled 1, 2, 3, and 0 if the user does not perform any activity. Assume that $A = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_7, \boldsymbol{y}_7)\}$ is the collect empirical data when the two residents perform activities, where $\boldsymbol{y}_1 = (0, 0)$, $\boldsymbol{y}_2 = (1, 0)$, $\boldsymbol{y}_3 = (1, 0)$, $\boldsymbol{y}_4 = (3, 0)$, $\boldsymbol{y}_5 = (1, 1)$, $\boldsymbol{y}_6 = (1, 1)$, and $\boldsymbol{y}_7 = (3, 3)$. In this case,$y_t = (y_t^1, y_t^2)$ is two-dimension, where $y_t^i,\ i = 1,\ 2$ represent the activity ID that the ith resident performed at time t.

For the four activity labels (1, 2, 3, and 0) for two residents, theoretically we can get 4 × 4 = 16 different vectors {(0,0), (0,1), …, (3,3)}. However, due to resident preferences, some exclusive and independent activities occurred, some states we cannot observe in fact. So, only seven label vectors are obtained.

To represent prior knowledge, some terms will be introduced below.

A single label y often has multiple possible values, which we often denote as state and all the possible value sets as state set. MRAR can be seen as a multi-label state labeling problem. For m residents, there are many possible values for $(y_t^1, y_t^2, \ldots, y_t^{\mathrm{m}})$, since different residents may perform different activities.

Here, we use the **State Event** $(y_t^1, y_t^2, \ldots, y_t^{\mathrm{m}})$ to represent the activities of multiple users at the same time, use **State Event Set A** to represent various values of State Events, and use **State Event Matrix M** to represent the values of State Events at time 1 to T. Then, State Event Set denotes as

$$A = \{\,(y_1^1, y_1^2, \ldots, y_1^m), (y_2^1, y_2^2, \ldots, y_2^m), \ldots, (y_K^1, y_K^2, \ldots, y_K^m)\,\}$$

where K is the State Events number.

State Event Matrix is given by

$$\mathbf{M} = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^m \\ y_2^1 & y_2^2 & \cdots & y_2^m \\ \vdots & \vdots & \ddots & \vdots \\ y_T^1 & y_T^2 & \cdots & y_T^m \end{bmatrix}$$

which includes T State Events.

For above multi-resident scenario, State Event Set can be denoted as

$$\mathbf{A1} = \{(0,\ 0),\ (1,\ 0),\ (3,\ 0),\ (1,\ 1),\ (3,\ 3)\}$$

The State Event Matrix can be denoted as

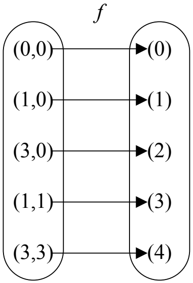
$$\mathbf{M1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 3 & 0 \\ 1 & 1 \\ 1 & 1 \\ 3 & 3 \end{bmatrix}$$

Note that two State Events in M may be the same, but any two State Events in A are different, and all State Events in M could be found in **A**. From **M1** we can see that the State Event 2 and 3 are the same, State Event 5 and 6 are the same, and all State Event can be found in **A1**.

Represented $(y_t^1, y_t^2, \ldots, y_t^m)$ with one uniquely combined label C, it can get **combined label states set B** $= \{0, 1, \ldots, K-1\}$. The map between State Event $(y_k^1, y_k^2, \ldots, y_k^m)$ and combined label state $C_k \in \mathbf{B}$ is defined as

$$(y_k^1, y_k^2, \ldots, y_k^m) \longrightarrow fC_k$$

For the State Event Set $\mathbf{A}1 = \{(0,\ 0), (1,\ 0), (3,\ 0), (1,\ 1), (3,\ 3)\}$, there a recombined label states set $\mathbf{B}1 = \{0, 1, 2, 3, 4\}$. The mapping is defined as



Similarly, there are mapping between **M1** and B1 as follows

$$\begin{bmatrix} 0\ 0 \\ 1\ 0 \\ 1\ 0 \\ 3\ 0 \\ 1\ 1 \\ 1\ 1 \\ 3\ 3 \end{bmatrix} \overset{f}{\rightarrow} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

The states of single activity y_t^i, $i \in \{0,\ 1,\ \ldots,\ K-1\}$ can be obtained by inverse mapping. In the two-resident scenario, for $C = 1$, it can get $\boldsymbol{y}_1 = 1, \boldsymbol{y}_2 = 0$ by inverse mapping, and for $\boldsymbol{C} = 3$, it can get $\boldsymbol{y}_1 = 1, \boldsymbol{y}_2 = 1$ by inverse mapping.

The algorithm of extracting latent knowledge in multi-resident activities is given in Figs. 41.2 and 41.3. The former is the model building flowchart, with which we can get State Event Set A, mapping f, combined label states set B, and combined label recognition model. The latter is activity recognizing flowchart from which it can see that new testing multi-resident activities are recognized with two steps: firstly, recognize the states of combined label C, then inverse map C to State Event by f^{-1}. Finally, figure out multi-resident activities based on the State Event.

It can be seen that the extracting latent knowledge algorithm did not use data association when recognizing multi-resident activities. But, if there is a need (i.e., tracking the resident), it can also find out data association. For $\boldsymbol{C} = 1$, if figure out $\boldsymbol{y}_1 = 1, \boldsymbol{y}_2 = 0$, we say the data is get by the first resident, since $\boldsymbol{y}_2 = 0$ represents the second resident does not carry out any activity and considered not trigger any sensors.

The algorithm can also handle some uncertain multi-resident activity patterns. For the two residents activity label $(A1, A2)$, where $A1$ is the activity that the first resident performed and $A2$ is the activity that the second resident performed. If $A1$ or $A2$ is equal '0,' it means the resident is performing one unknown activity and can be any one activity. For two residents with N total activities, there are

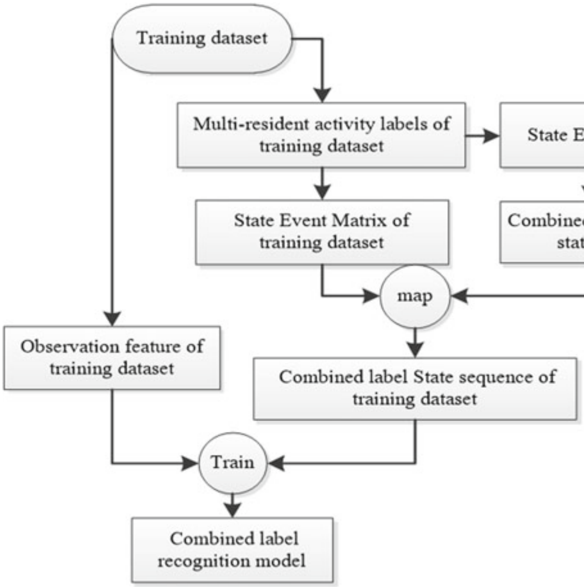
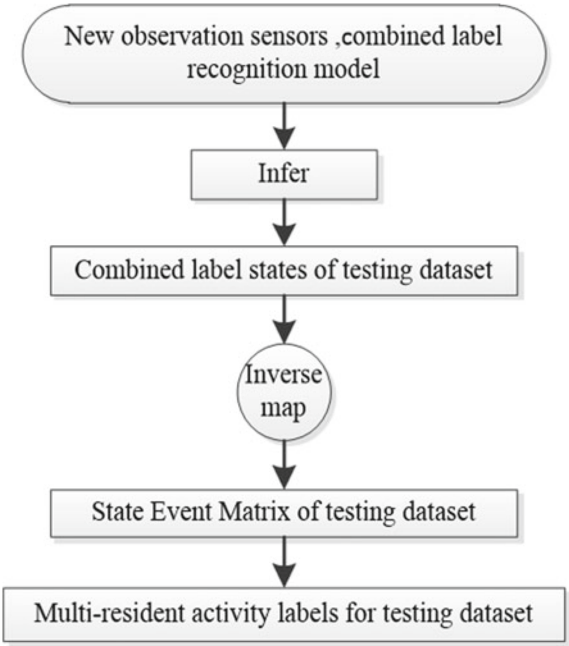

Training dataset
Multi-resident activity labels of training dataset
State Event Set
State Event Matrix of training dataset
Combined label and state set
map
Observation feature of training dataset
Combined label State sequence of training dataset
Train
Combined label recognition model

Fig. 41.2 Model building flowchart

Fig. 41.3 Activity recognizing flowchart


New observation sensors ,combined label recognition model
Infer
Combined label states of testing dataset
Inverse map
State Event Matrix of testing dataset
Multi-resident activity labels for testing dataset

$$\begin{aligned}
&(0, A2) = (1, A2) \vee (2, A2), \ldots, \vee(N, A2)\\
&(A1, 0) = (A1, 1) \vee (A1, 2), \ldots, \vee(A1, N)\\
&(0, 0) = (0, A2) \vee (A1, 0)
\end{aligned}$$

The unknown state is actually a union of all possible activities, thus the algorithm can handle some uncertainties and can improve activity recognition.

41.4 Validation

41.4.1 Validation 1

To validate our feature combining method, two experiments will be given. For every experiment, we will introduce the datasets, their activities, and sensor features, and then carry out our experiments. To measure the percentage of correctly classified testing samples, we define the recognition accuracy of the class as

$$\text{Accuracy} = \frac{\sum_{n=1}^{N}[\inf\text{erred}(n) = \text{true}(n)]}{N} \tag{1}$$

where N is total testing samples.

In addition, to verify the recognition of a single class, we also give the recognition accuracy of individual activities as

$$\frac{\text{inferred}_c(n) = \text{true}_c(n)}{N_c} \tag{2}$$

where N_c is the total samples that contained in class c.

Experiment 1
The first experiment is based on the "ADL adlnormal" dataset that is collected in WSU Apartment Test bed [14]. There are five daily activities in the dataset. The apartment is installed with various non-invasive sensors.

The raw sensor number, cleaned sensor number, and the finally sensor number after sensor combining are shown in Table 41.1. As it was shown, the sensor number decreases obviously after sensor cleaning whereas the finally sensor number after

Table 41.1 Sensor number changes after cleaning and combining

Feature	Sensor number
All feature	39
Clean feature	23
Finally feature	19

Table 41.2 Involved sensor number changes before and after sensor combining for five individual activities

	A1	A2	A3	A4	A5
Before	13	14	16	15	17
After	9	14	16	15	17

sensor combining does not decrease much. The involved sensor numbers before and after sensor combining for the five activities can be seen in Table 41.2.

After the sensors cleaning and combining, we take every sensor as a feature and recognize activities in this dataset using CRF with threefold cross-validations. To validate our sensors combining solution, we compared the results with the result before sensor cleaning and sensor combining. Table 41.3 is the recognition accuracy and the total time that used for both training and testing with raw sensor and finally sensor. From the table, we can see that the recognition accuracy is increased after sensor cleaning and sensor combining whereas the time used is reduced.

The recognition accuracies for five individual activities with the raw sensor and with the sensor after cleaning and combining are shown in Fig. 41.4. We can see that the recognition accuracies of all the activities are increased after sensor cleaning and sensor combining for that our method not only can reduce parameter in count, but also can avoid the error caused by redundant information.

Experiment 2 The second experiment focuses on routine morning activities collected in kitchen outfitted with 60 RFID tags [8]. In the kitchen, 11 routine morning

Table 41.3 Recognition accuracy and the total time changes

	Accuracy	Time
Raw sensor	0.8308	77.1716
Senor after combining	0.8460	58.1504

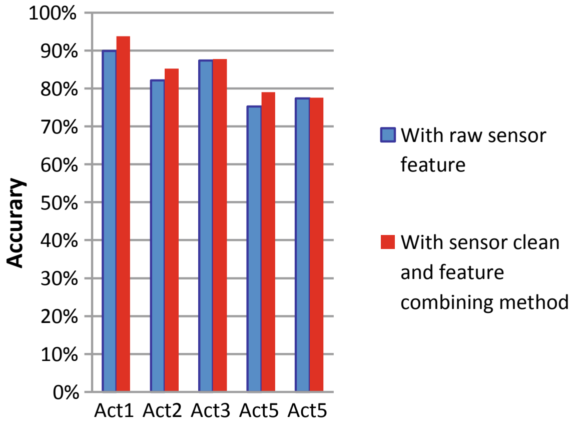


Fig. 41.4 Recognition accuracies for five individual activities

activities are performed by user in different ways and use the RFID tags to collect sensor data. The objects that attached tags include bowl, coffee container, cupboard, dishwasher, door, drawer, egg carton, hand soap, kettle, cooking spoon, stove control, telephone, and so on.

Before combining the sensors, we first clean the uninvolved sensors, since some sensors may not involve in all the activities. The raw sensor number, cleaned sensor number, and the finally sensor number after combining are shown in Table 41.4. From the table, we can see that all of the 60 sensors are involved in the activities and the total sensor number decreases obviously after combining.

Also, we give the involved sensor number before and after combining for 11 individual activities in Table 41.5, where A_i, $i = 1, \ldots, 11$, presents the ith activity. The table shows that the sensors in activities 4, 5, 6, 8, 10 are combined, and thus the involved sensor numbers are decreased.

After sensors combining, we take every sensor as a feature. To validate the algorithm of extracting latent knowledge between sensors and activities, we recognize activities based on CRF with leave-one-out cross-validation. Also, we compare the results with PCA method that extracting 35 principal components as features. Table 41.6 is the recognition accuracy and the total time that used for both training and testing with different method. From the table, we can see that the recognition accuracy is increased after sensor combining whereas the time used is reduced. Although based on the feature with same dimension, PCA gets worse result than the algorithm

Table 41.4 Sensor number changes after cleaning and combining

Sensor type	Sensor number
Raw sensor number	60
Sensor number after cleaning	60
Sensor number after combining	35

Table 41.5 Involved sensor number before and after combining for 11 individual activities

Activity	Before	After
$A1$	13	13
$A2$	10	10
$A3$	1	1
$A4$	17	7
$A5$	6	4
$A6$	15	9
$A7$	6	6
$A8$	13	10
$A9$	14	14
$A10$	5	1
$A11$	1	1

Table 41.6 Recognition accuracy and the total time changes

	Accuracy	Time (s)
Raw feature	0.9355	1245.75
Combined feature with 35 feature	0.9388	1043.30
PCA with 35 feature	0.9228	1234.03

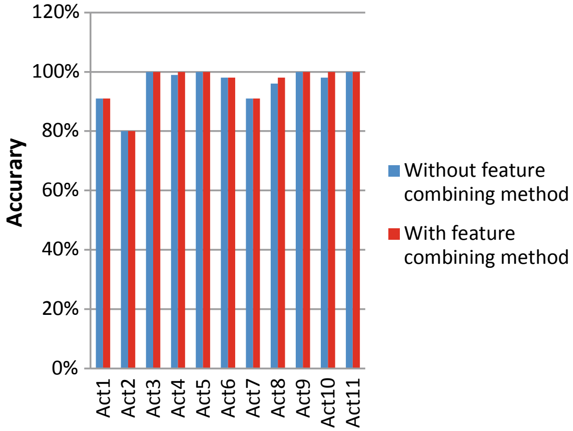


Fig. 41.5 Recognition accuracies for 11 individual activities

that extracting latent knowledge between sensors and activities with much time that used for both training and testing.

The recognition accuracies for 11 individual activities are shown in Fig. 41.5. We can see that the recognition accuracies of activity 4, 8, 10 are increased after sensor combining. This is because our method not only can reduce parameter in count, but also can avoid the error caused by redundant information.

41.4.2 Validation 2

We will validate our algorithm exploiting latent knowledge of multi-resident activities based on multi-resident activities dataset [4] collected in the CASAS project. In the dataset, there are two residents and 15 activities.

The multi-resident activity State Event and their frequency F are shown in Table 41.7.

As Table 41.7 shows that some State Events occur frequently, while some appear rarely, does not happen actually. For two residents with 16 (activity 0 represent the resident performed unknown activity), there are 16 × 16 = 256 State Events theoretically, but in this case, there are only 27 State Events, since some State Event do not occur at all actually. To validate our algorithm, one experiment will be given.

Table 41.7 Occurrence counts of different State Events

$(A1, A2)$	F	$(A1, A2)$	F	$(A1, A2)$	F
(0, 0)	3	(0, 15)	748	(10, 11)	284
(0, 2)	1568	(1, 0)	1175	(10, 15)	2
(0, 3)	668	(4, 0)	864	(12, 0)	1179
(0, 4)	1	(6, 0)	1505	(12, 13)	272
(0, 5)	545	(6, 7)	350	(13, 0)	4
(0, 7)	1529	(9, 0)	866	(13, 13)	865
(0, 8)	432	(9, 8)	280	(14, 0)	387
(0, 11)	891	(10, 0)	660	(14, 15)	845
(0, 13)	1309	(10, 1)	1	(15, 0)	1

Experiment 3 This experiment is carried out with three fold cross-validations. In the training stage, firstly, it will build mapping f *and* inverse mapping f^{-1}, and map State Event Matrix of training data as combined label state sequence. Then, dynamic activity recognition algorithm, such as HMM, CRF, and latent-dynamic conditional random fields (LDCRF) [4], is trained with observation sequences and combined label states sequence.

In the testing stage, we estimate combined label state firstly with the trained model and observation sequences in the test dataset. The average accuracy of combined label state for HMM with latent knowledge (LK-HMM) and CRF with latent knowledge (LK-CRF), and LDCRF with latent knowledge (LK-LDCRF) are 65.46, 67.61, and 63.87% correspondingly.

It is important to note that the above is not the ultimate accuracy of MRAR. To get multi-resident activity of test dataset, we need to map combined label states to State Event Matrix with f^{-1}. Figure 41.6 is the average MRAR accuracies of five models. From it, we can see that LK-HMM gets 75.77%, LK-CRF gets 75.38%, and LK-LDCRF gets 72.69% which all get higher average MRAR accuracies than single

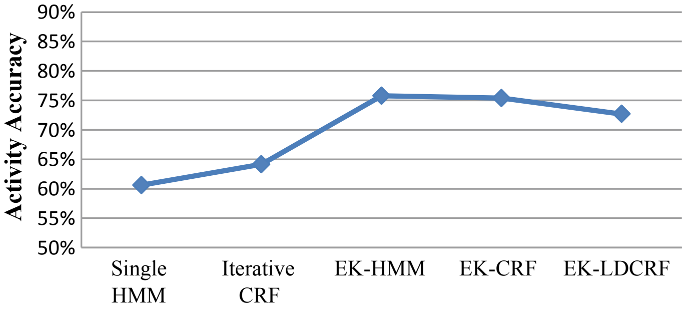


Fig. 41.6 Average recognition accuracies of five models in recognizing activities for multi-residents

HMM and iterative CRF which did not give data association when MRAR. Thus, mining latent knowledge can help MRAR.

There are some reasons for LK-HMM, LK-CRF, and LK-LDCRF to outperform single HMM and iterative CRF. Firstly, when single HMM model is implemented for both residents, it cannot well represent transitions between activities and multi-resident at the same time. Secondly, although iterative CRF does not need to give data associations, it still needs to learn data associations for MRAR. When data association is learned badly, MRAR accuracy will be low. In addition, LK-HMM, LK-CRF, and LK-LDCRF can mine knowledge in the multi-resident environment and can capture global features and trends of multi-resident activities.

In the LK-LDCRF case, it gets lower accuracy than LK-CRF which gets higher accuracy than CRF in single-user activity recognition [4]. This is because there are many combined label states with different internal structure, fixed hidden state for LK-LDCRF is difficult to adapt to all combined label states. Thus, the number of hidden states is difficult to determine, and it is not easy to determine. If we chose not suitable hidden states number, LK-LDCRF will get lower accuracy. In future, we will study the chosen suitable hidden states number for LDCRF in multi-resident environment and compare the result to our method.

41.5 Conclusions

This paper recognizes activities with some latent knowledge that exists in training samples. Firstly, this paper gives one new pretreatment method for activity recognition by extracting latent knowledge between sensors and activities, and then it proves one new multi-resident activity recognition algorithm by extracting latent knowledge in multi-resident activities. From the simulation, we conclude that extracting latent knowledge can greatly enhance activity recognition.

Acknowledgements This work was supported by the Anhui Provincial Natural Science Research Project of Colleges and Universities (No. KJ2017A933), the National Natural Science Foundation of China (No. 11872166) and China Postdoctoral Science Foundation (No. 2016M592042).

References

1. Inoue, M., Inoue, S., Nishida, T.: Deep recurrent neural network for mobile human activity recognition with high throughput. Artif Life Robot **23**(2), 173–185 (2018)
2. Kadouche R, Pigot H, Abdulrazak B et al (2011) User's behavior classification model for smart houses occupant prediction. In: Activity recognition in pervasive intelligent environments. Atlantis Press, pp 149–164
3. van Kasteren T, Krose B (2007) Bayesian activity recognition in residence for elders

4. Singla, G., Cook, D.J., Schmitter-Edgecombe, M.: Recognizing independent and joint activities among multiple residents in smart environments. J Ambient Intell Humaniz Comput **1**(1), 57–63 (2010)
5. Tong, Y., Chen, R.: Latent-dynamic conditional random fields for recognizing activities in smart homes. J Ambient Intell Smart Environ **6**(1), 39–55 (2014)
6. Wang, J., Chen, Y., Hao, S., et al.: Deep learning for sensor-based activity recognition: a survey. Pattern Recogn Lett **119**, 3–11 (2019)
7. Ma, C.Y., Chen, M.H., Kira, Z., et al.: Ts-lstm and temporal-inception: exploiting spatiotemporal dynamics for activity recognition. Sig Process Image Commun **71**, 76–87 (2019)
8. Patterson DJ, Fox D, Kautz H et al (2005) Fine-grained activity recognition by aggregating abstract object usage. In: Ninth IEEE international symposium on wearable computers (ISWC'05). IEEE, pp 44–51
9. Kasteren TV, Noulas A et al (2008) Accurate activity recognition in a home setting. In: Proceedings of the 10th international conference on Ubiquitous computing, Seoul, Korea
10. Mantyjarvi J, Himberg J, Seppanen T (2001) Recognizing human motion with multiple acceleration sensors. In: 2001 IEEE international conference on systems, man and cybernetics. e-Systems and e-Man for cybernetics in cyberspace (Cat. No. 01CH37236). IEEE, vol 2, pp 747–752
11. Hsu KC, Chiang YT et al (2010) Strategies for inference mechanism of conditional random fields for multi-resident activity recognition in a smart home. In: The twenty third international conference on industrial, engineering & other applications of applied intelligent systems (IEA-AIE), Córdoba, Spain, pp 417–426
12. Chiang YT, Hsu KC (2010) Interaction models for multi-resident activity recognition in a smart home. Intelligent robots and systems (IROS), pp 3753–3758
13. Wang, L., Gu, T., et al.: Recognizing multi-user activities using wearable sensors in a smart home. Pervasive Mob Comput **7**, 287–298 (2011)
14. Cook, D.J., Schmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. Methods Inf Med **48**(05), 480–485 (2009)