TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 12/14 pp952–964
DOI: 10.26599/TST.2022.9010072
Volume 28, Number 5, October 2023

Generating Markov Logic Networks Rulebase Based on Probabilistic Latent Semantics Analysis

Shan Cui, Tao Zhu, Xiao Zhang, Liming Chen, Lingfeng Mao, and Huansheng Ning*

Abstract: Human Activity Recognition (HAR) has become a subject of concern and plays an important role in daily life. HAR uses sensor devices to collect user behavior data, obtain human activity information and identify them. Markov Logic Networks (MLN) are widely used in HAR as an effective combination of knowledge and data. MLN can solve the problems of complexity and uncertainty, and has good knowledge expression ability. However, MLN structure learning is relatively weak and requires a lot of computing and storage resources. Essentially, the MLN structure is derived from sensor data in the current scene. Assuming that the sensor data can be effectively sliced and the sliced data can be converted into semantic rules, MLN structure can be obtained. To this end, we propose a rulebase building scheme based on probabilistic latent semantic analysis to provide a semantic rulebase for MLN learning. Such a rulebase can reduce the time required for MLN structure learning. We apply the rulebase building scheme to single-person indoor activity recognition and prove that the scheme can effectively reduce the MLN learning time. In addition, we evaluate the parameters of the rulebase building scheme to check its stability.

Key words: Markov Logic Network (MLN); structure learning; rulebase construction; probabilistic latent semantics

1 Introduction

As a typical algorithm combining data and drive, Markov Logic Network (MLN) is also used by many people in Human Activity Recognition (HAR) to deal with some uncertain or complex problems for activity recognition models. However, MLN structure learning requires a large amount of memory and computational resources. It also has a slightly lower efficiency than some advanced neural network models. Besides, MLN rarely uses real-time data streams to construct activity recognition models[1]. Effective construction of a real-time MLN structure is, therefore, an urgent problem to be solved[2]. The MLN structure is composed of a set of first-order logical rules[3]. If we build the first-order logical rulebase required by MLN in the current scenario, we can save the time required for learning the ruleset for MLN and improve the MLN learning efficiency.

Essentially, many environmental sensors are deployed in the indoor environment, and the sensors present activities performed by people in a continuous flow. In the current scenario, the MLN structure is derived from the environment sensor data. It is a first-order logic representation of all acquired indoor activities. Each activity comprises a series of sequential actions captured using sensors deployed in the room, so that a piece of data over a period represents an activity. This way, we can obtain an MLN structure by assuming we can effectively slice sensor data and translate the fragments into semantic rules. However, there is no clear boundary between activities; there is even some overlapping.

• Shan Cui, Xiao Zhang, Lingfeng Mao, and Huansheng Ning are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: cuishan@xs.ustb.edu.cn; zhxustb@163.com; lingfengmao@ustb.edu.cn; ninghuansheng@ustb.edu.cn.
• Tao Zhu is with the Computer School, University of South China, Hengyang 421001, China. E-mail: tzhu@usc.edu.cn.
• Liming Chen is with the School of Computing, Ulster University, Belfast, BT15 1AP, UK. E-mail: liming.chen@dmu.ac.uk.
∗ To whom correspondence should be addressed.
Manuscript received: 2022-05-10; revised: 2022-08-10; accepted: 2022-11-27

Therefore, we need to segment the continuous sensor data stream into information fragments.

Accurate representation of a unique activity based on dense environment sensors is a critical issue for segmenting sensor data. We therefore propose a semantic rulebase building scheme based on probabilistic latent semantic analysis (see Fig. 1). Probabilistic latent semantic analysis can convert text collections into text-word co-occurrence data. Intuitively, words with similar semantics and texts with similar semantics will be clustered into the same soft category, and such a soft category represents the topic. The semantic rulebase building scheme is to construct a ruleset that conforms to the current indoor characteristics based on the sensor dataset. The semanticized text of the sensor dataset contains multiple activity classes, and each activity class is composed of various activity atoms. Active atoms can be observed from text collections, while active categories are latent and cannot be found in the text. Therefore, we adopt probabilistic latent semantic analysis to build a semantic rulebase. This scheme solves three critical problems of sensor data segmentation and semantic rulebase building:

- Cutting the data stream into active segments according to sensor deployment location and activity characteristics;
- Converting semantics based on sensor features, and improving semantic segments using latent probabilistic semantic algorithms;
- Using prior knowledge to establish the mapping between semantic fragments and activity categories and constructing the semantic rulebase.

Crucially, the semantic rulebase is not only widely applicable to the MLN structure learning in different activity scenarios, but can also be used in other probabilistic semantic algorithms.

The rest of this paper is organized as follows: Section 2 briefly reviews data segmentation techniques and algorithms for activity identification. Section 3 proposes a semantic rulebase building scheme based on probabilistic latent semantic analysis. Section 4 presents activity recognition as an example and verifies the feasibility and stability of the semantic rulebase scheme. Section 5 summarizes the whole paper.

2 Related Work

2.1 MLN structure learning

In 2009, Domingos and Lowd[3] proposed a probabilistic semantic algorithm, MLN, a data-knowledge-driven approach. MLN learning mainly includes parameter learning and structure learning. Under the premise of the network structure, MLN needs to further learn and optimize the model parameters; this process is called parameter learning. MLN structure learning refers to learning the optimal or suboptimal network structure from the database. Specifically, MLN structure is the set of the knowledge network's first-order logic rules F_i. And structure learning refers to the generation of ruleset F for any world x given some constants. For any possible world x, a probability distribution of the structure can be expressed as

$$P(X=x)=\frac{1}{Z}\exp(\sum_i w_i n_i(x))=\frac{1}{Z}\prod_i \phi_i(x_i)^{n_i(x)} \tag{1}$$

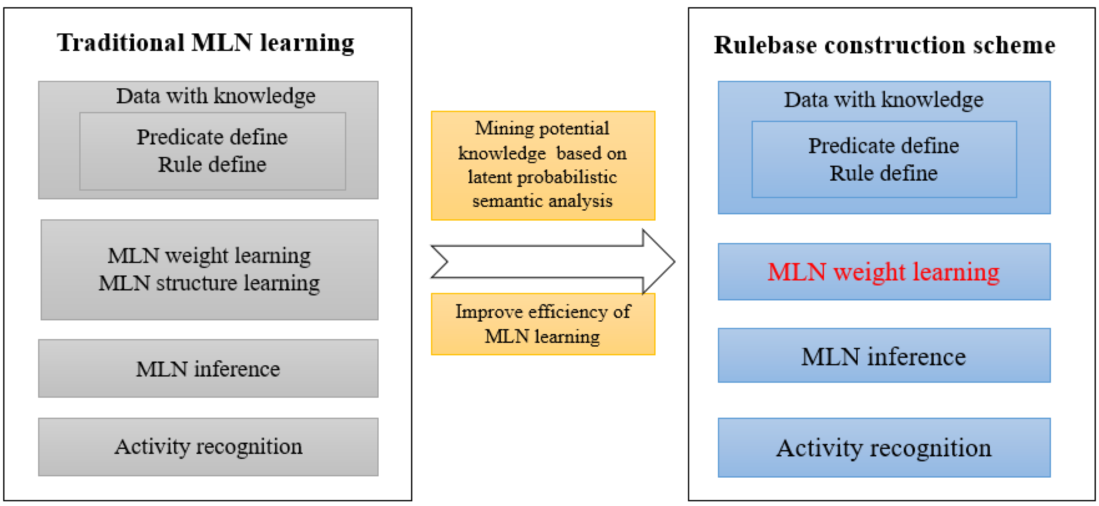

Fig. 1 Semantic rulebase building scheme based on probabilistic latent semantic analysis.

where $n_i(x)$ is the number of true groundings of F_i in x, x_i is the state (truth value) of the predicate appearing in F_i, Z is the partition function, $Z = \sum_{x \in X} \prod_k \phi_k(x_{\{k\}})$, and $\phi_k(x_k) = \mathrm{e}^{w_k}$, the calculation of $P(X = x)$ is based on the current vector weighted set w.

MLN is widely used in activity recognition because it can deal with uncertain information and clearly express the model's knowledge[3]. Chahuara et al.[4] used sensor data to extract user attributes, such as residents' voice, location, and mood, to set sliding windows according to user attributes. They then used MLN to create a rule classification model for probabilistic reasoning. In 2013, Chakraborty et al.[5] established a set of logical models of domain-specific knowledge through MLN knowledge representability. Gayathri et al.[6] proposed an MLN-based layering method that depends on action-based factors, such as object, location, time, and duration. Activity identification is based on the priority of factors associated with each layer. Next, Gayathri et al.[7] combined probabilistic reasoning with representation domain ontology to enhance ontology-based activity recognition using MLN probabilistic reasoning. They argued that the data obtained from sensors are inherently uncertain and that the uncertainty of the mapping ontology will not produce good accuracy within an augmented reality environment. Honda et al.[8] took the sensor value as input and used MLN to express the association rules between sensors and activities as soft logic expressions. MLN estimation activities and user feedback were then used to achieve better recognition accuracy. Cui et al.[9] proposed an MLN activity recognition model based on continuous learning to improve the learning ability of MLN in 2022.

2.2 Data segmentation

A lot of research has been done on data segmentation (user's state and environment at a certain point in time) to map active labels to sensor data[1]. Data segmentation refers to splitting the logically unified data into smaller and independently managed physical units for storage to facilitate reconstruction, reorganization, and recovery. There are many segmentation methods, roughly divided into horizontal segmentation and vertical segmentation. Horizontal splitting entails splitting tuples of global relationships into subsets called data fragments. In data fragmentation, data may be aggregated due to some common properties (e.g., geography and affiliation). Typically, data fragments in a relationship are disjoint. Vertical splitting divides global connections into data fragments based on attribute groups (longitudinal). Data in data fragments may need aggregation due to the ease of use or common access. In general, vertical data fragments in a relationship overlap only in some fundamental values, while other attributes are disjoint[10]. Sensor data cutting is horizontal cutting in activity recognition. Many activity recognition studies mostly use a sliding window to cut data.

In the sliding window concept, the time window is used to perform the segmentation of sensor data flow[1]. The segmentation methods can be roughly divided into static sliding[11] and dynamic sliding[12] time windows. Chua et al.[13] proposed a knowledge-driven model based on hidden Markov to segment sensor data streams. They use variable window lengths to move through a series of observations. The main disadvantage of this approach is that it requires a preexisting data set to determine the optimal size of the time window for splitting the data and constructing activity rules. Riboni et al.[14] proposed a fixed time window of one minute for splitting the data stream and selecting several sets of sensor data streams per minute. The model activity rules are based on the current dataset and have poor universality. Additionally, the use of a fixed-size time window may result in significant computational overhead.

In addition to considering the static or dynamic nature of the time window, researchers also need to analyze the probability of time windows overlapping. In case of overlapping time windows, sensor data can be shared by two or more time windows. In the absence of overlapping time windows, sensor data of each time window is exclusive to itself[15]. Tapia et al.[16] set different values for the time windows during initialization. Once a time window is active, its length cannot be dynamically modified. Fixed sliding does not divide between sensors, resulting in overlapping time windows. Hong et al.[17] used the concept of time continuity in sensor data and location context for data segmentation. This method works well when continuous activity takes place in different locations. Okeyo et al.[12] proposed a tree hierarchical activity model in which leaf activity is the most specific description. The time window is dynamically adjusted until the capture leaf activity satisfies the activity description. Segmentation fails if the window cannot determine the duration of the activity.

To use the processed data for HAR, researchers use the current mainstream methods for learning activity rules, building activity models, and completing activity identification. Yang et al.[18] used the divide-and-conquer strategy to divide dynamic and static activities and adopted a neural network as the classifier of activity recognition. They proposed an effective feature subset selection method to ensure the ability to learn and recognize complex activities. Anguita et al.[19] proposed a multiclass activity recognition model based on a Support Vector Machine (SVM). Compared to traditional SVM, this model develops a more sustainable activity construction model while maintaining similar accuracy. Qi et al.[20] proposed a recurrent neural network model based on semantics. The model can extract different spatiotemporal features and capture relationships between populations. Wang et al.[21] proposed a sensor-based incremental learning approach, which is a fuzzy clustering algorithm based on a probabilistic neural network. The algorithm balances incremental learning ability and recognition accuracy. Xie et al.[22] proposed a spectrum-sensing deep learning algorithm based on a convolutional neural network. Compared with model-based spectrum sensing algorithms, their proposed deep learning method is data-driven, combining both current and historical data.

3 Rulebase Building Scheme

This section proposes a semantic rulebase building scheme based on probabilistic latent semantic analysis, including dataset segmentation, semantization, and rule modification, creating matches between activity categories with rulesets.

3.1 Dataset segmentation

Sensor dataset $S = \{t_1, t_2, \dots, t_l\}$ is collected by environmental sensors deployed indoors, and the sensor set is $s = \{s_1, s_2, \dots, s_m\}$. S is a discrete nonquantitative index, which only records the sensor trigger state. We need to transform the sensor dataset into discrete quantitative indicators for easy data processing. The house in which the data was collected was made into a two-dimensional graph with its boundary as the coordinate axis. The sensor s_i has the corresponding position coordinate (x_i, y_i). We take the actual data collection scenario as an example. Figure 2 shows the deployment of the residential environment, and Fig. 3 shows the sensor deployment of the residential environment.

As shown in Fig. 3, we not only record the sensor position, but also divide the indoor activity area, set as $C = \{C_1, C_2, \dots, C_k\}$. The categories of human activities in a given area are limited. Therefore, we can construct an identifiable and universal activity categories set $A = \{A_1, A_2, \dots, A_n\}$ based on the sensors deployed indoors. Collected activity data show that most human activity is concentrated within a single area, such that once a person moves from one activity area to another, the type of activity often changes. Therefore, we use sensor distance to capture user movement. Suppose that there are any two sensors, s_i

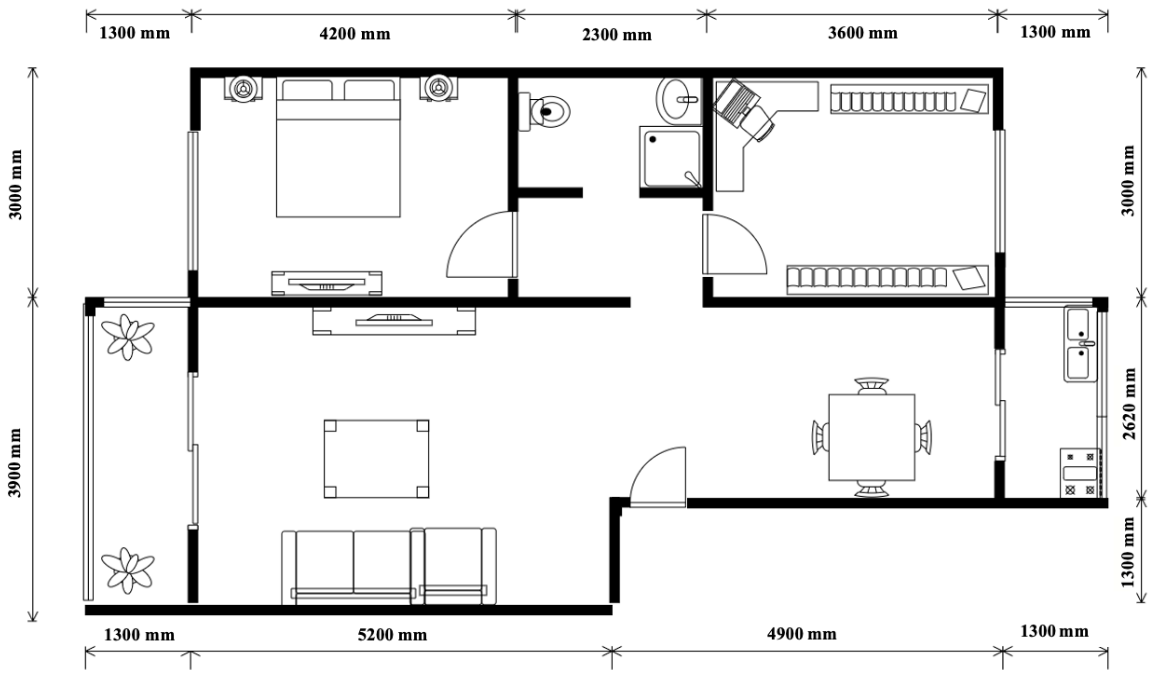

Fig. 2 Deployment of the residential environment.

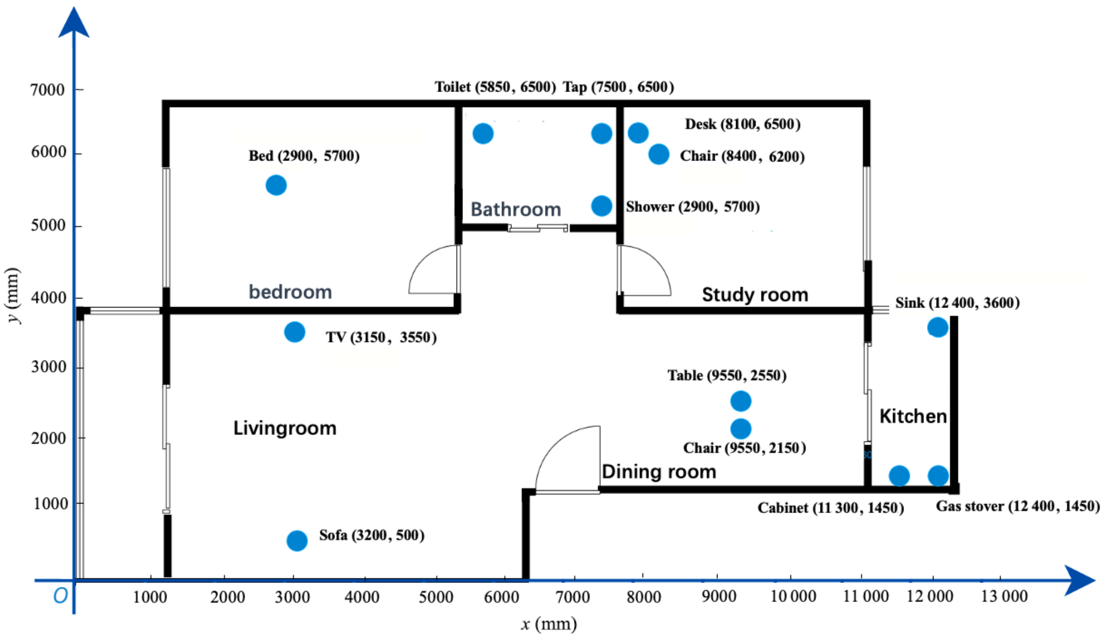

Fig. 3 Sensor deployment of the residential environment.

and s_j in the room; we use their straight-line distance $d(s_i, s_j)$ to describe the mobile range of the user,

$$d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

Sensor positions are close to each other for sensors in the same area, while sensors in two different areas have positions that are far apart. To describe the change of user activity category more specifically, we select the two nearest sensor distances in different areas of the room as the critical value d_{min},

$$d_{min}(A_i, A_j) = min_{s_i \in C_i, s_j \in C_j} d(s_i, s_j).$$

If the distance difference between the two sensors exceeds d_{min}, it is believed that the activity areas have been changed, and the activity category has changed. Based on d_{min}, we can conduct a rough segmentation of sensor data, and get $\{S_1, S_2, \ldots, S_k\}$, where $S_i = \{s_j\}(1 \leqslant j \leqslant m)$. Sensors are used for recording people's momentary movements. Some sensors are repeatedly triggered through engagement in certain activities. In this way, some redundant data in the sensor dataset is not conducive to the subsequent semantic processing. Therefore, we only retain its initial occurrence data for any sensor s_j of S_i. If s_j appears again, we need to remove it directly from S_i. Thus, S_i only preserves the types of sensors triggered in certain active areas, arranged in chronological order. And we obtain a refined sensor dataset $S' = \{S'_1, S'_2, \ldots, S'_{k'}\}$ $(k' \leqslant k)$, where $S'_i = \{s_j\}(1 \leqslant j \leqslant m)$.

Because human activities are limited and repetitive, some of the same elements exist in the dataset S'. If S'_i and S'_j are the same and S'_i is realized earlier than S'_j. In this way, we can select different S_i in S' and get the dataset $S'' = \{S''_1, S''_2, \ldots, S''_{k''}\}$ $(k'' \leqslant k', S''_i = \{s_j\}(1 \leqslant j \leqslant m))$. It is assumed that the activities in S'' are different and independent. To find the same element in S', we change the coordinate value of sensor s_i to the straight-line distance d_i from the sensor to the origin $(0, 0)$,

$$d_i = \sqrt{x_i^2 + y_i^2}.$$

Each type of activity has a different number of sensors. We choose c, which has the largest number of sensors in S''_i, as the benchmark, and the other vectors need to fill 0 to c bits automatically. In this way, we can find different $\{S''_i\}$ by calculating the minimized squared error e of S'',

$$e = \sum_{i=1}^{k'} \sum_{j=i+1}^{k'} \|S_i - S_j\|_2^2.$$

To some extent, e characterizes the similarity of the activities in $\{S'_1, S'_2, \ldots, S'_{k'}\}$. The smaller the e value, the higher the degree similarity. $e = 0$ indicates that the two activity categories are the same. According to e, we can find the independent active data $\{S''_1, S''_2, \ldots, S''_{k''}\}$ in $\{S'_1, S'_2, \ldots, S'_{k'}\}$.

It is important to note that drawing two-dimensional axes on the boundaries of the house does not destroy the original placement of sensors. We define the value of sensors as the distance between the origin $(0, 0)$ and the sensor $s_i(x, y)$. If any point in the house is $(0, 0)$, some sensors will be equidistant from $(0, 0)$. We take the boundary of the house as the coordinate axis; there will be no sensors in relation to the coordinate axis symmetry, reducing the total number of sensors that are equidistant from $(0, 0)$.

According to Algorithm 1, data segmentation occurs through the following three steps:

(1) Use a two-dimensional coordinate system to transform the sensor dataset into a discrete pointset.

(2) Determine the data segmentation value according to the active area and the straight-line distance of the sensor.

Algorithm 1 Sensor distance segmentation algorithm

Input: Dataset $S = \{t_1, t_2, \dots, t_l\}$ and set of sensors $s = \{s_1, s_2, \dots, s_m\}$

Output: $S' = \{S'_1, S'_2, \dots, S'_{k'}\}$ and $S'' = \{S''_1, S''_2, \dots, S''_{k''}\}$

for $i = 1$ to m **do**
  Convert sensor trigger state to the corresponding position coordinate (x_i, y_i);
  Calculate the sensor distances of the two in different areas: $d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ $(s_i \in C_i, s_j \in C_j)$;
  Select two nearest sensor distance in different areas as the critical value:
  $d_{min}(A_i, A_j) = min_{s_i \in C_i, s_j \in C_j} d(s_i, s_j)$;
end for
for $i = 1$ to l, **do**
  $d(t_i, t_{i+1}) = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$;
end for
if $d(t_i, t_{i+1}) \geqslant d_{min}$ **then**
  $i = i + 1$;
  $j = j + 1$;
  S_j ={The next position of the previous cut is retained to t_{i+1}};
else
  S_j={The next position of the previous cut is retained to t_i};
end if
return $\{S_1, S_2, \dots, S_k\}$;
for $i = 1$ to k **do**
  Retain its initial occurrence data for any sensor s_j of S_i;
end for
return $S' = \{S'_1, S'_2, \dots, S'_{k'}\}$ $(k \leqslant k')$;
for $i = 1$ to k' and $j = 1$ to k' **do**
  Change the sensor $s_i = (x_i, y_i)$ to the straight-line distance d_i from (x_i, y_i) to the origin (0, 0);
  $d_i = \sqrt{x_i^2 + y_i^2}$;
  Select the number of sensors in S_i with the largest number c of sensors in S' as the benchmark, and the other vectors need to fill 0 to c bits automatically;
  $e = \sum_{i=1}^{k'} \sum_{j=i+1}^{k'} \|S_i - S_j\|_2^2$;
end for
return $S'' = \{S''_1, S''_2, \dots, S''_{k''}\}$.

(3) Obtain mutually independent datasets by using the minimum square deviation.

3.2 Semantization and rule modification

A series of human movements are recorded by sensors deployed indoors hence forming part of the data information. Dataset S'' is not associated with activity category construction. We need to establish the mapping relationship between sensor information and activity category with the help of expert knowledge. Therefore, we need the semantic dataset. Semantization and rule modification consist of knowledge list creation, crossregional activity analysis, and potential probabilistic semantic analysis.

3.2.1 Knowledge list creation

The set of activity categories $A = \{A_1, A_2, \dots, A_n\}$ is an identifiable and highly versatile set for the sensor based on indoor deployment. We use expert knowledge to create a simple knowledge list K about A. Simplicity in this context means that there are at most three active atoms in A and that A only contains the key actions associated with the activity. For example,

$$R = \{sink, pot, oatmeal, bowl, gasstove\}$$

would be the $Cooking$ rule for a real user scenario, and

$$R(Cooking) = \{pot, gasstove\}$$

would be the knowledge of $Cooking$ in knowledge list K. The knowledge list K consists of activity category set A and ruleset F corresponding to the activity. We define K as

$$A = \{A_1, A_2, \dots, A_n\};$$
$$R = \{R_1, R_2, \dots, R_m\};$$
$$F_i = \{Active\ atomset\ R_j; Activity\ category\ A_i\},$$

where category A_i is the index of the ruleset R.

3.2.2 Crossregional activity analysis

Previously, we assumed that the user's activity area is related to its activity category and only analyzed activities in a single region. There was no analysis of crossregional activities. We need to use a knowledge list K for data integration and semantic analysis of crossregion activities. Therefore, we need to mark the activity category A_i with special annotation based on sensors, and splice $S'' = \{S''_1, S''_2, \dots, S''_{k''}\}$ to get $S''' = \{S'''_1, S'''_2, \dots, S'''_{k'''}\}$.

The splicing of S'' occurs over the following three steps:

(1) Mark area label S''_i in S'' according to the position of sensors;

(2) Find crossregional activity A_t in K, and determine the active area C_i and C_j of A_t;

(3) Find the elements of S''_i and S''_j in S'', and splice them.

3.2.3 Potential probabilistic semantic analysis

Next, we transform independent sensor data S'' and S''' into semantic information with the help of prior expert knowledge. Therefore, we need to deploy the sensor set $s = \{s_1, s_2, \ldots, s_m\}$, perform semantic annotation to generate $w = \{w_1, w_2, \ldots, w_m\}$, where w_i is the semantization of sensor s_i and represents active action atoms. In this way, $S'' = \{S''_1, S''_2, \ldots, S''_{k''}\}$ and $S'''=\{S'''_1, S'''_2, \ldots, S'''_{k'''}\}$ will be converted to the text set $D = \{D_1, D_2, \ldots, D_{k''}\}$ and $D' = \{D'_1, D'_2, \ldots, D'_{k'''}\}$, respectively, and D_i and D'_i is a set of w_i.

To ensure the integrity of text sets D and D', there is redundant data in S'' and S''' selected by us. Therefore, we need to delete or modify D and D' to construct a ruleset conforming to the current indoor. Text sets D and D' are composed of rulesets of different activity classes; each activity contains several active atoms. Activity atoms w_i can be observed from the text set, while activity categories are latent and cannot be found in the text. Therefore, we use probabilistic latent semantic analysis for D and D' to delete or modify.

The text D_i consists of the active atoms in the ruleset $\{R_i\}$, so we only need to determine the active atomset of a certain activity class A_i. That is, we need to calculate $p(\{w_i\}|A_i)$. Our ultimate goal is to calculate how many activities there are in the text and which active atoms are contained in the activities. Therefore, we use the *EM* algorithm to calculate $p(A_k|D_i)$ and $p(w_i|A_i)$ in the text sets D and D', and iterate over Steps E and M until they converge (see Algorithm 2).

Step *E*: Solve $Q(\theta, \theta^{(i)})$ and its maximum likelihood estimation.

$$p(A_i|w_i, D_j) = \frac{p(w_i|A_k)p(A_k|D_j)}{\sum_{k=1}^{K} p(w_i|A_k)p(A_k|D_j)},$$

$$Q(\theta, \theta^{(i)}) = E_A[\log p(w, A|\theta)|w, \theta^{(i)}] = E_A \log p(w, A|\theta)p(A|w, \theta^{(i)}).$$

Step *M*: Solve the maximization of $Q(\theta, \theta^{(i)})$ and get $\theta^{(i)+1}$.

$$p(w_i|A_i) = \frac{\sum_{j=1}^{N} n(w_i, D_j)p(z_k|w_i, D_j)}{\sum_{m=1}^{M}\sum_{j=1}^{N} n(w_m, D_j)p(z_k|w_m, D_j)},$$

$$p(A_k|D_i) = \frac{\sum_{j=1}^{N} n(w_i, D_j)p(z_k|w_i, D_j)}{n(D_j)},$$

$$\theta^{(i+1)} = \arg\max_{\theta} Q(\theta, \theta^{(i)}).$$

Algorithm 2 Semantization and rule modification algorithm

Input: Initial values of $p(w_i|A_i)$ and $p(A_k|D_i)$
Output: $p(w_i|A_i)$ and $p(A_k|D_i)$

for $i, i' = 1$ to m, $j, j' = 1$ to k **do**
  $p(A_i|w_i, D_j) = \frac{p(w_i|A_k)p(A_k|D_j)}{\sum_{k=1}^{K} p(w_i|A_k)p(A_k|D_j)}$;
  $Q(\theta, \theta^{(i)}) = E_A[\log p(w, A|\theta)|w, \theta^{(i)}] = E_A \log p(w, A|\theta)p(A|w, \theta^{(i)})$;
end for
for $i = 1$ to m **do**
  $p(w_i|A_i) = \frac{\sum_{j=1}^{N} n(w_i, D_j)p(z_k|w_i, D_j)}{\sum_{m=1}^{M}\sum_{j=1}^{N} n(w_m, D_j)p(z_k|w_m, D_j)}$;
  $p(A_k|D_i) = \frac{\sum_{j=1}^{N} n(w_i, D_j)p(z_k|w_i, D_j)}{n(D_j)}$;
  $\theta^{(i+1)} = \arg\max_\theta Q(\theta, \theta^{(i)})$;
end for

3.3 Matching activity categories with rulesets

The sensor information we collected was never mapped to activity categories in the first and second steps. Since activity category A_i comprises potential atoms, $\{A_i\}$ can be used to modify the rules but cannot build contact between $\{A_i\}$ and $\{w_i\}$. Therefore, we need to build a ruleset $R' = \{R'_1, R'_2, \ldots, R'_n\}$ with high generality and establish the mapping of $\{A_i\}$ and $\{w_i\}$ with prior expert knowledge. With the help of a ruleset F, we can build mapping through atomic,

$$f : R(D_i) = \{\{w_i\}; A_i\},$$

which links activity for atoms and categories. The mapping construction needs to match the action atoms in the text set $\{D\}$ and F one by one. If the action atoms in D_i are successfully paired with F_j, D_i contains all the action atoms in F_j.

To build a rulebase, we need to do the following:

(1) If an activity category A_i corresponds to only one text D_j, the activity category is added directly to rulebase R;

(2) If the text D_i does not find its matching R and fails to find the corresponding activity category of D_i, we need to use expert knowledge to determine whether it corresponds to the existing activity categories. If it corresponds to an existing activity category, we remove D_i. If this activity category is not defined in F, we need to manually add D_i to build new rules R and F.

(3) If there are multiple rules for the same activity category, we select text D_i with the largest number of action atoms and modify it. We need to eliminate the excess active atoms and keep R_i in the rulebase R. The rest of the text set is deleted.

(4) If text D_i has multiple activity categories, we need to determine whether D_i needs to be further cut. If

multiple activities sequentially generate D_i and there is no crossover of activities, D_i needs to be cut further. If D_i occurs across multiple activities, we need to split D_i so that one activity category corresponds to one rule. Besides, we need to decide whether to update the rules contained in D_i (see Algorithm 3).

Let's take *Eating* and *Cooking* as an example,

$$R'(Eating) = \{dishes, table\},$$
$$R'(Cooking) = \{pot, gasstove\}.$$

Let the corresponding text D_i we collected according to the specific scene be in the following:

$$D_1 = \{cabinet, dishes, tableware, table\},$$
$$D_2 = \{sink, pot, oatmeal, bowl, gas\ stove\},$$
$$D_3 = \{pot, tap, oatmeal, gas\ stove\},$$
$$D_4 = \{dishes, sink, tap, dish\ soap, cabinet\},$$
$$D_5 = \{sink, pot, oatmeal, bowl, gas\ stove, dishes, sink, tap, dishsoap, cabinet\},$$
$$D_6 = \{sink, dishes, pot, oatmeal, bowl, tap, dishsoap, gasstove, cabinet\}.$$

Algorithm 3 Activity category matching algorithm

Input: $R' = \{R'_1, R'_2, \ldots, R'_n\}$, $A = \{A_1, A_2, \ldots, A_n\}$, and $D = \{D_1, D_2, \ldots, D_{k''}\}$
Output: $R' = \{R'_1, R'_2, \ldots, R'_{n'}\} (n \leqslant n')$ and $R = \{R_1, R_2, \ldots, R_t\} (t \leqslant k'')$

for $i = 1$ to k'' and $j = 1$ to n **do**
  Match R' and D one by one;
  $f(D_i) = \{\{w_i\}; A_j\}$;
end for
if D_i corresponds to only one activity category A_j **then**
  return $R_i = \{\{D_i\}; A_j\}$;
end if
if D_i does not find its matching R' **then**
  Determine whether D_i corresponds to existing activity category;
  if D_i corresponds to an existing activity category A_j **then**
    Remove D_i;
  else
    Add D_i to build new rules R and R';
  end if
end if
if exiting multiple rules for the same activity category **then**
  Select D_i with the largest number of action atoms;
  Modify D_i to D'_i;
  Delete the rest of the text set that has the same activity category;
  Return $R_i = \{\{D'_i\}; A_j\}$;
end if
if D_i has multiple activity categories **then**
  Whether D_i needs to be further cut;
  if D_i does not occur across multiple activities, **then**
    Cut D_i;
  else
    Split D_i that one activity category corresponds to one rule;
    Decide whether to update the rules contained in D_i;
  end if
end if

We match text set D with rulebase R one by one to see whether text D_i contains all action atoms of a certain rule R'_i. D_1 contains all the atoms in $R'(Eating)$, so $R_1 = \{D_1; Eating\}$. D_2 and D_3 contain atoms of $R'(Cooking)$. Since D_2 has more atoms than D_3, we exclude D_3 and keep D_2, so $R_2 = \{D_2; Cooking\}$. Both D_5 and D_6 contain all atoms of $R'(Eating)$ and $R'(Cooking)$. We just split D_5 and D_6 in terms of R_1 and R_2. D_5 has no active crossover, while D_6 has an active crossover. We need to split D_5 and D_6 into independent text sets according to R_1 and R_2. The split rules are compared with R_1 and R_2 to see whether R_1 and R_2 are further updated. D_4 does not find a matching R', it is analyzed to determine whether it is a new rule. It is found that D_4 is a new rule and we need to manually add the activity category for D_4 and add $R_3 = \{$*dishes*, *sink*, *tap*, *dish soap*, *cabinet*; *Washing dishes*$\}$ to R and R'. Our final rulebase R is as follows:

$$R_1 = \{cabinet, dishes, tableware, table; Eating\},$$
$$R_2 = \{sink, pot, oatmeal, bowl, gasstove; Cooking\},$$
$$R_3 = \{dishes, sink, tap, dishsoap, cabinet; Washingdishes\}.$$

4 Experimental Analysis

In this section, we applied the rulebase building scheme to single-person indoor activity identification, evaluated the practicality and stability of the constructed rulebase, and verified whether the semantic rule library improved the learning efficiency of MLN.

4.1 Dataset description

We selected two datasets as data sources for experimental analysis, both of which recorded the activity information of a single person in the room. In order to evaluate the rationality of the semantic rulebase building scheme, we selected the dataset of the WSU CASAS smart home project as dataset 1[23], which contains more than 8000 items. The data record five user activity types: making phone calls, eating, washing hands, washing dishes, and cooking. Dataset 2 is the sensor data we collected by ourselves and contains 6000 data items. Figure 2 shows the indoor living environment from which we collected the data, and Fig. 3 records the indoor sensor deployment position. We collected six user activity types during the data collection period, including washing, eating, watching TV, cooking, sleeping, and reading.

4.2 Experimental setup

The indoor residential environment and sensor deployment diagram of datasets 1 and 2 were drawn on a certain scale, and the position of the sensor deployment diagram was used to construct a two-dimensional coordinate system. Then, we converted the sensor position into coordinates and divided the active area. By this way, we get the sensor coordinate dataset for datasets 1 and 2. Referring to the sensor point set, we calculated the distance between the sensors and the user's active area and analyzed the whole indoor sensor deployment position. To protect the integrity of sensor data as much as possible, we have to evaluate all distances calculated to select the segmentation comprehensively; the distances are d_{min_1} and d_{min_2} of datasets 1 and 2, respectively. Using d_{min_1} and d_{min_2} to cut the sensor coordinate dataset, we get S_1 and S_2 of datasets 1 and 2, respectively. It is necessary to ensure that S_1 and S_2 contain all sensor datasets.

For better semantic annotation, we only need the sensor of the element set S_i in $S_{dataset1}$ and $S_{dataset2}$ to appear only once. Since our ultimate goal is to build a semantic rulebase for the current indoor environment, sensors represent actions at a certain moment. Actions are the constituent atoms of a rule and only need to appear once to express the semantic integrity of the rule. In this way, we get datasets $S'_{dataset1}$ and $S'_{dataset2}$. The indoor activities are limited, and users generally do some repetitive activities, such as eating, washing, and cooking. So there will be the same S'_i in S'. We filter out the same S'_i in S' by the least squared difference, so that the elements in S' are independent of each other. Finally, we get the sensor dataset S''.

Next, we semantically segment datasets $S''_{dataset1}$ and $S''_{dataset2}$, annotating deployed sensors as active actions. In this way, we convert all elements in S'' into sets of actions and get D. When slicing raw sensor data, we do not consider the crossregional activity. To realize the construction of the semantic rulebase for crossregional activities, we introduce the knowledge list K and use K to merge the elements with crossregional activities in S'' to obtain D'. We do not map activity rules and categories in the whole process, so we regard activity categories as latent elements. Therefore, we use latent probabilistic semantics to construct and revise rules for D and D'.

To establish the mapping relationship between rulesets and activity categories, we introduce prior knowledge K. The list of prior knowledge K covers all possible activities in datasets 1 and 2. The content of K contains the iconic actions and activity categories of the activities, not all the actions. With the iconic actions active in K, we match the rule elements in D and D' with the rules in K. In this way, we establish a one-to-one matching of rules and activity classes in D and D' and build a rulebase R. If we find that new rules appear, we automatically add the new rules to the knowledge list K and the rulebase R.

4.3 Experimental evaluation and results

4.3.1 Analysis of effectiveness

To improve the learning efficiency of MLN, we propose a rulebase building scheme based on probabilistic latent semantic analysis. In this part, we need to evaluate the effectiveness of the semantic rulebase scheme. Therefore, we select the MLN conventional learning algorithm for comparison, as shown in Table 1. During the experiment, we design three steps using datasets 1 and 2, and the data and knowledge are incremental from the first step to the third step. By analyzing the experimental results, we have the following results:

First, we find that the rulebase building scheme can accept larger datasets because the increase of data has little effect on the learning time of the rulebase building scheme (Table 1).

MLN structure learning is to update each atom of any rule iteratively, which is single-threaded learning. And it needs to update and store the learned knowledge in the learning process, which consumes a lot of computing and storage resources. The rulebase building scheme cuts the data stream to obtain the MLN structure, only to analyze the data stream itself (based on the sensor's inherent properties, such as location and time). MLN weight learning does not need to consume a lot of resources and time since it is just optimizing and calculating the weights of rules.

Second, we analyze the accuracy of the rulebase building scheme from three perspectives of precision, recall, and F1-score. We found the accuracy of this scheme to be slightly higher than that of MLN learning (see Fig. 4).

Ideally, MLN structure learning does not end until

Table 1 Comparison of semantic rulebase building scheme and MLN learning.

Dataset	Rule	Semantic rulebase building scheme			MLN learning	
		Time for semantic rulebase construction	Time for MLN weight learning	Accuracy (%)	Time for MLN structure learning	Accuracy (%)
Dataset 1	$R(Callphone)$ $R(Washhand)$ $R(Cookoatmeal)$	0.12 s	32 min, 14.18 s	89.27	3 d, 51 min, 36.10 s	64.13
	$R(Callphone)$ $R(Washhand)$ $R(Cookoatmeal)$ $R(Eating)$	0.12 s	37 min, 34.67 s	88.47	10 d, 2 h, 40 min, 48.10 s	65.76
	$R(Callphone)$ $R(Washhand)$ $R(Cookoatmeal)$ $R(Eating)$ $R(Washdishes)$	0.56 s	8 h, 17 min, 4.80 s	88.23	17 d, 5 h, 10 min, 12.10 s	62.09
Dataset 2	$R(Washing)$ $R(WatchTV)$ $R(Reading)$ $R(Cooking)$	1.25 s	3 h, 0.10 s	91.31	5 d, 1 h, 46 min, 48.10 s	73.13
	$R(Washing)$ $R(WatchTV)$ $R(Reading)$ $R(Cooking)$ $R(Eating)$	2.01 s	4 h, 28 min, 10.20 s	91.27	13 d, 8 h, 7 min, 12.10 s	74.13
	$R(Washing)$ $R(WatchTV)$ $R(Reading)$ $R(Cooking)$ $R(Eating)$ $R(Sleeping)$	6.22 s	1 d, 6 min, 36.10 s	89.63	19 d, 11 h, 7 min, 48.10 s	74.83

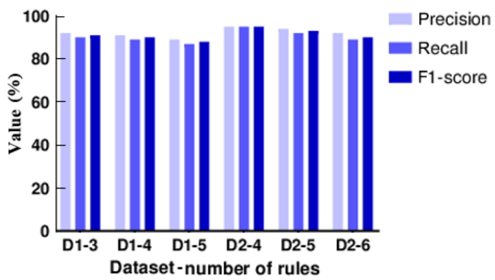

Fig. 4 Accuracy analysis of rulebase building scheme, D1 and D2 denote datasets 1 and 2, respectively.

the rule parameters converge. However, its parameter convergence requires a lot of time and computational resources. Therefore, we need to set the parameters of structure learning convergence to reduce the learning time, affecting the accuracy rate of MLN structure learning. The accuracy of a rulebase building scheme is affected by the ruleset. The ruleset is obtained by data segmentation and modified by latent probabilistic semantic analysis. There are two parameters, d_{min} and e, in data segmentation. To ensure data integrity, we set d_{min} to a fixed value, so there is no data loss. It is worth noting that in case the data integrity requirement in the cutting process is low, the value of d can be increased based on a fixed value. e is the standard for further data cutting and correction, which can be regarded as the convergence value of the ruleset. To improve the accuracy of ruleset construction, we evaluate it to optimize the scheme. Therefore, the scheme can select the optimal rule set convergence value in the execution process, and the result is shown in Fig. 5.

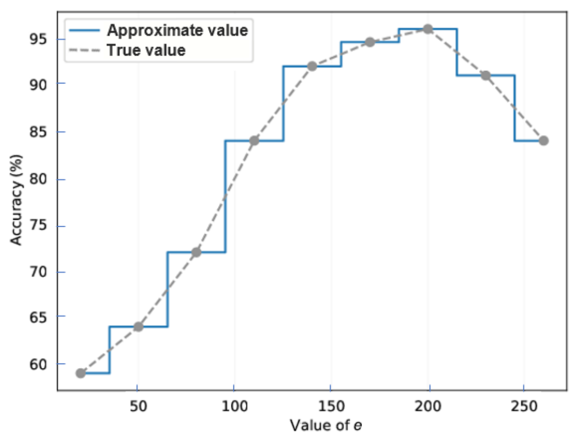

Fig. 5 Accuracy assessment of parameter *e*.

Third, under the condition of ensuring accuracy, the learning time of the rulebase building scheme is much less than the MLN learning time (see Table 1). The construction time of the semantic rulebase is composed of three parts: data cutting, semantic modification, and activity category mapping. The rulebase building scheme increases workload in the learning process when compared with systematic learning methods, such as MLN structure learning. However, the building scheme can be used for multiple activity scenarios by modifying the parameters and activity categories required by the user. Furthermore, Table 1 shows that semantic rulebase building takes much less time than MLN learning. So we find that the semantic rulebase building scheme is more effective and more practical than MLN learning.

4.3.2 Analysis of stability

To better evaluate the stability of the rulebase building scheme based on potential probabilistic semantic analysis, we compare *EM* in the probabilistic latent semantic algorithm with other typical clustering comparison algorithms. We select four rules of dataset 2 as experimental data. The correct performance of dataset 2 is the best, and it could eliminate data interference to the greatest extent and evaluate the stability of the building scheme from the algorithm itself. We select the current representative K-means, affinity propagation, and mean shift for evaluation and comparison. The baseline comparison in the experiment is shown in Fig. 6.

K-means: we randomly select K objects as the initial clustering center. Then the distance between each object and the seed cluster center is calculated, and each object is assigned to the cluster center closest to it. Cluster centers and the objects assigned to them represent a cluster. Once all objects have been assigned, the cluster center for each cluster is recalculated based on objects existing in the cluster. This process is repeated until a certain termination condition was met.

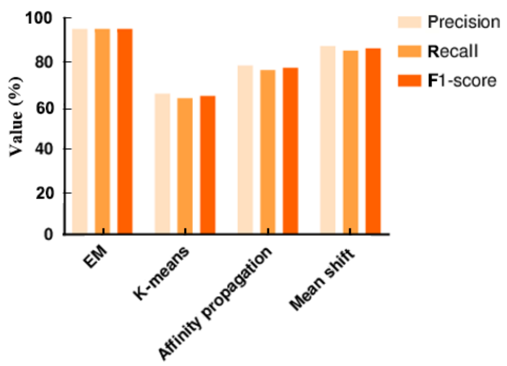

Fig. 6 Stability evaluation of rulebase building scheme.

Affinity propagation: we regard all samples as the network's nodes and then calculate the cluster center for each sample based on the messages passing through each edge in the network. In the clustering process, there are two kinds of messages transmitted between nodes: responsibility and availability. The algorithm continuously updates the attractiveness and attribution of each point through an iterative process until high-quality exemplars (equivalent to centroids) are generated, and the remaining data points are assigned to the corresponding clusters.

Mean shift: It is a density-based nonparametric clustering algorithm. Assuming that the different clusters' datasets conform to different probability density distributions, the fastest direction in which the density of any sample point increases is found, and the area with high sample density is considered to correspond to the maximum value of the distribution. These sample points eventually converge at local density maxima. Points that converge to the same local maxima are considered members of the same cluster class.

5 Conclusion

This paper proposes a semantic rulebase building scheme based on probabilistic latent semantic analysis, which effectively improves the efficiency of MLN learning. First, we use sensor location and activity characteristics to segment the data stream, so as to remove redundant data and achieve independence between data stream segments. Second, we semantically interpret the deployed sensors and further revise the semantic fragments through probabilistic latent semantic analysis to maximize the mining of rules in the semantic fragments. Finally, we use prior knowledge to construct a one-to-one mapping of rules and activity categories to construct a semantic rulebase. Based on datasets 1 and 2, we compared the rulebase building scheme with traditional MLN learning. We also tested the clustering algorithm and parameters in the scheme to verify the effectiveness of the rulebase building scheme. In the future, we will further refine the segmentation algorithm for sensor data so that the rules can be created more accurately.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61872038).

References

[1] D. Triboan, L. M. Chen, F. Chen, and Z. Wang, A semantics-based approach to sensor data segmentation in real-time

Activity Recognition, *Future Gener. Comput. Syst.*, vol. 93, pp. 224–236, 2019.

[2] S. Kok and P. Domingos, Learning the structure of Markov logic networks, in *Proc. 22nd Int. Conf. on Machine Learning*, Bonn, Germany, 2005, pp. 441–448.

[3] P. Domingos and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*. San Rafael, CA, USA: Morgan & Claypool, 2009.

[4] P. Chahuara, A. Fleury, F. Portet, and M. Vacher, Using Markov logic network for on-line activity recognition from non-visual home automation sensors, in *Proc. 3rd Int. Joint Conf. on Ambient Intelligence*, Pisa, Italy, 2012, pp. 177–192.

[5] I. Chakraborty, A. Elgammal, and R. S. Burd, Video based activity recognition in trauma resuscitation, in *Proc. 10th IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition*, Shanghai, China, 2013, pp. 1–8.

[6] K. S. Gayathri, S. Elias, and B. Ravindran, Hierarchical activity recognition for dementia care using Markov logic network, *Pers. Ubiquit. Comput.*, vol. 19, no. 2, pp. 271–285, 2015.

[7] K. S. Gayathri, K. S. Easwarakumar, and S. Elias, Probabilistic ontology based activity recognition in smart homes using Markov logic network, *Knowl.-Based Syst.*, vol. 121, pp. 173–184, 2017.

[8] Y. Honda, H. Yamaguchi, and T. Higashino, Daily activity recognition based on Markov logic network for elderly monitoring, in *Proc. 16th IEEE Annu. Consumer Communications & Networking Conf.*, Las Vegas, NV, USA, 2019, pp. 1–6.

[9] S. Cui, T. Zhu, X. Zhang, and H. Ning, MCLA: Research on cumulative learning of Markov logic network, *Knowl.-Based Syst.*, vol. 242, pp. 108352, 2022.

[10] T. Zhang and C. C. J. Kuo, Audio content analysis for online audiovisual data segmentation and classification, *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 441–457, 2001.

[11] K. Iqbal, M. Isa, S. A. Buzdar, K. A. Gifford, and M. Afzal, Treatment planning evaluation of sliding window and multiple static segments technique in intensity modulated radiotherapy, *Rep. Pract. Oncol. Radiother.*, vol. 18, no. 2, pp. 101–106, 2013.

[12] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, Dynamic sensor data segmentation for real-time knowledge-driven activity recognition, *Pervasive Mob. Comput.*, vol. 10, pp. 155–172, 2014.

[13] S. L. Chua, S. Marsland, and H. W. Guesgen, Spatio-temporal and context reasoning in smart homes, in *Proc. COSIT Workshop on Spatial and Temporal Reasoning for Ambient Intelligence System*, Freiburg, Germany, 2009, pp. 9–20.

[14] D. Riboni, L. Pareschi, L. Radaelli, and C. Bettini, Is ontology-based activity recognition really effective? in *Proc.* 2011 *IEEE Int. Conf. on Pervasive Computing and Communications Workshops*, Seattle, WA, USA, 2011, pp. 427–431.

[15] H. Cho, J. An, I. Hong, and Y. Lee, Automatic sensor data stream segmentation for real-time activity prediction in smart spaces, in *Proc.* 2015 *Workshop on IoT Challenges in Mobile and Industrial Systems*, Florence, Italy, 2015, pp. 13–18.

[16] E. M. Tapia, S. S. Intille, and K. Larson, Activity recognition in the home using simple and ubiquitous sensors, in *Proc. 2nd Int. Conf. on Pervasive Computing*, Vienna, Austria, 2004, pp. 158–175.

[17] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin, Evidential fusion of sensor data for activity recognition in smart homes, *Pervasive Mob. Comput.*, vol. 5, no. 3, pp. 236–252, 2009.

[18] J. Y. Yang, J. S. Wang, and Y. P. Chen, Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers, *Pattern Recognit. Lett.*, vol. 29, no. 16, pp. 2213–2220, 2008.

[19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in *Proc. 4th Int. Workshop on Ambient Assisted Living*, Vitoria-Gasteiz, Spain, 2012, pp. 216–223.

[20] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. Van Gool, stagNet: An attentive semantic RNN for group activity recognition, in *Proc. 15th European Conf. on Computer Vision*, Munich, Germany, 2018, pp. 104–120.

[21] Z. Wang, M. Jiang, Y. Hu, and H. Li, An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors, *IEEE Trans. Inform. Technol. Biomed.*, vol. 16, no. 4, pp. 691–699, 2012.

[22] J. Xie, C. Liu, Y. C. Liang, and J. Fang, Activity pattern aware spectrum sensing: A CNN-based deep learning approach, *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1025–1028, 2019.

[23] D. Cook, Learning setting-generalized activity models for smart spaces, *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, 2010.

Shan Cui received the MEng degree from Shandong Normal University, China in 2015, and the BEng degree from Weifang University, China in 2012. Currently, she is a PhD candidate at the School of Computer and Communication Engineering, University of Science and Technology Beijing. Her research interests include probabilistic semantic algorithms and indoor activity recognition.

Xiao Zhang received the BEng degree from Ludong University, China in 2015. He is currently a master student at the School of Computer and Communication Engineering, University of Science and Technology Beijing. His research interests include evolutionary computation and the Internet of things.

Tao Zhu received the PhD degree from the University of Science and Technology of China, China in 2015. Then, he worked as a postdoctoral researcher and a lecturer at the School of Computer and Communication Engineering, University of Science and Technology Beijing in 2016–2018. Currently, he is working at University of South China. His research interests include evolutionary computation and the Internet of things.

Liming Chen received the PhD degree in computer science from De Montfort University, Leicester, UK in 2003. Currently, he is a professor and research director at the School of Computing, Ulster University, UK. His research interests include data analytics, pervasive computing, artificial intelligence, as well as user-centered intelligent systems and their applications in digital healthcare. He is an IET fellow and a senior member of the IEEE. He has authored six books and over 230 peer-reviewed papers.

Lingfeng Mao received the PhD degree from Peking University, Beijing, China in 2001. He is currently a professor at the School of Computer and Communication Engineering, University of Science and Technology Beijing. He has published about 180 journal and conference papers. His research activities include reliability, modeling, and characterization of materials, devices, and systems.

Huansheng Ning received the PhD degree from Beihang University, Beijing, China in 2001. He is now a professor and vice dean at the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research focuses on the IoT and general cyberspace. He is the founder of the cyberspace and cybermatics international science and technology.