User Daily Activity Pattern Learning: A Multi-memory Modeling Approach

Shan Gao and Ah-Hwee Tan

***Abstract*— In this paper, we propose a multi-memory model, ADLART model, to discover the daily activity pattern of a sensor monitored user from his/her activities of daily living (ADL). The proposed model mimics the human multiple memory system which comprises a working memory, an episodic memory, and a semantic memory. Through encoding user's daily activities patterns in episodic memory and extracting the regularities of activity routines in semantic memory, the ADLART system is able to learn, recognize, compare, and retrieve daily ADL patterns of the user. Experiments are presented to show the performance of the ADLART model using different parameter settings and its performance is discussed in details.**

I. Introduction

AGEING of population is now a common issue faced by major countries all over the world. According to World Health Organization [1], the proportion of elderly people will be doubled from 11% to 22% in 2050, and there will be about 2 billion people aged 60 and older by 2050. This situation will add to the burden of government and society in terms of healthcare and social welfare, however, it also provides great opportunities in renovating the existing economy model and bringing innovations.

Researches, e.g. [2], show that elderly people may have various problems ranging from sensory loss, reduced speed and increased variance in moving time, reduced speech capacity, to reduced information process capacity. The cumulative effect of these problems often leads to more elderly people move from home to care institutions which usually adds huge financial pressure to both individual and society.

Smart home is a concept that home could be equipped with information and communication technologies to relieve the problems that elderly people have, and assist them to stay in their own home comfortably over a long period. User activity pattern learning is one important open problem. The effective modeling of user's activity pattern could help in various applications such as life assistant, physical and mental health monitoring, financial planning, and product recommendation.

Activities of daily living (ADLs), as used in healthcare field, refers to daily self care activities performed by an individual in his place of residence, outdoor, or both. ADLs are usually used as a measurement of the functional status of a person, particularly for elderly, children, and disabled people. There are two subcategories of ADLs. Basic ADLs (BADLs) [3] refer to daily activities that people do to maintain their wellbeing, such as feeding themselves, bathing, and dressing. Instrumental ADLs (IADLs) [4] [5] are not necessary for fundamental functions, but they help an individual to live independently in a community. Examples of IADLs includes shopping, social activity, and financial management. Generally speaking, ADLs are important to indicate mental and body issues, especially for elderly people and disabled people. Some issues faced by elderly people will be reflected in their ADLs. For example, if a person is suffering from a joint disease, he will gradually develop slower motion and result in longer ADLs lasting time. Monitoring time and frequency of ADLs together with sufficient knowledge base will help caregivers to predict health trend of the elderly people and give advice in advance.

There have been many research work on ADL recognition. In 2005, Duong et al [6] had a research on ADL recognition and abnormality detection using switching hidden semi-Markov Model. In their research, they use camera to capture users position inside a smart home. They use switching hidden semi-Markov model to model the transition of positions in the smart home to classify different ADLs. In 2010, Fleury, Vacher, and Noury [7] used support vector machine to classify seven ADLs (hygiene, toilet use, eating, resting, sleeping, communication, and dressing) from various smart home sensor ranging from infra-red presence sensors, door contacts, temperature, micro-phone, to wearable kinematic sensor.

Compared with ADL recognition, there are few works tackle the problem of ADL pattern learning. Some researchers, e.g. [8] [9], use topic model approach to discover activity patterns, wherein topics are pulled from a document using a bag-of-words approach. In activity pattern learning, words are corresponding to low level activities, while topics are corresponding to ADLs (they call it daily routine in the sense that it is repeated everyday). The word distribution expected for a set of topics is:

$$p(w|d) = \sum_{z=1}^{T} p(w|z)p(z|d) \quad (1)$$

Where d denotes documents, w denotes word, and z denotes topic

Their works study mainly the relationships between low level activities to ADLs. However, to best of our knowledge, there is no existing work focusing on a larger scale of activity pattern learning, for example, daily activity pattern learning.

Authors are with School of Computer Engineering, Nanyang Technological University, Singapore (email: gaos0009@e.ntu.edu.sg, asahtan@ntu.edu.sg).

This research is supported in part by Interactive and Digital Media Programme Office (IDMPO), National Research Foundation (NRF) hosted at Media Development Authority (MDA) under Grant No.: MDA/IDM/2012/8/8-2 VOL 01.

In this paper we aim to address the user daily activity pattern learning problem. This level of activity pattern knowledge provides a higher level understanding of user activity pattern that could be used to analyze the relationship between user's daily activity routine and his health condition. To solve the daily activity pattern learning problem, we propose a memory model, called ADLART, to mimics the human multiple memory system which comprises a working memory, an episodic memory, and a semantic memory.

Experiments are conducted to test different parameter settings and show the performance of ADLART. The results show that ADLART is capable to learn, recognize, and retrieve daily activities patterns.

The rest of this paper is organized as follows: Section II describes the problem of user activity pattern learning, its requirement and design challenges. Section III gives the foundation ART model Fusion-ART and its episodic memory variation, EM-ART. IV introduces the proposed episodic memory inspired fusion ART model ADLART architecture and algorithms. Experiment is conducted and the results are presented in Section V to show its functionality and performance. Finally, Section VI concludes the paper.

II. User Activity Pattern Learning Issues

A. Data Structure Formation

There are two basic elements in daily ADL routines, namely the individual ADLs and their sequences. To achieve efficient encoding of ADLs and their sequences, the data structure should firstly serve the function of distinguishing them from each other. Secondly, the data structure should be able to identify key ADL differences between two ADL sequences. For example, the user had three meals in one day, but had no food in another day. Thirdly, the data structure should be able to give certain tolerance to difference in different ADL sequences in order to classify them into clusters. For example, it should not be a major difference between a day having lunch at 12:10pm and another day at 12:15pm. The basic challenges in data structure design is to distinguish ADLs and their sequence, having key ADL difference identified while tolerating certain level of variations in ADL orders and happening time.

B. Daily Routine Retrieval

In daily routine retrieval, we have identified some key tasks to be achieved. First task is ADL sequence formation that the system should be able to construct ADL sequence data structures from a series of ADL input with time-stamp. Second task is ADL sequence recognition that a stored sequence of ADL could be able to be identified in response to a incoming ADL sequence data. The recognition mechanism is preferred to be able to tolerate partial data with auto recall of missing parts. Lastly, the model should be able to retrieve all individual ADLs from recognized sequences.

C. Semantic Pattern Learning

People have variations in ADL timing every day, but they usually have clear routines in different type of days. ADLs in working day should be more or less the same to each other, but very different from those in vacation days. The preferred model should be able to gradually learn semantic patterns of user's typical days, like holiday, working days, Fridays, and sick days. In a case that an input ADL sequence matches a different type of day, it could be a signal for a abnormal situation to be noticed. For example, if the people is sick in a working day, he may wake up and dress up slower than his normal working days.

III. Adaptive Resonance Theory Models

A. Fusion ART

Adaptive Resonance Theory (ART) [10] models are capable to learn recognition categories of multi-dimensional mappings of input patterns in an online and incremental manner. Various models of ART and their supervised learning versions are used in pattern analysis and recognition tasks. Within the family of ART models, there is a group of networks known as Fusion ART [11] or multi-channel adaptive resonance associative map (multi-channel ARAM) [12], which formulates cognitive codes associating multi-modal patterns across multiple input channels. Multi-channel ARAM structures can also be used for reinforcement learning, for example, there is a multi-channel ARAM called FALCON described in [13] [14].

The architecture of a typical Fusion ART model is shown in Figure 1. The dynamics of Fusion ART are summarized below.

Input Fields: Let F_1^k denote the input field that holds the input patterns of channel *k*.

Input Vectors: Let $\mathbf{I}^k = (I_1^k, I_2^k, ..., I_n^k)$, where $I_i^k \in [0,1]$, denote the input vector of channel *k*, for $k = 1,...,n$.

Category Field: Let F_i where $i > 1$ indicate a category field. In the standard multi-channel ART, there is only one category field which is F_2.

Activity Vectors: Let $\mathbf{x}^k$ denote the activity vector for input field F_1^k. And, $\mathbf{y} = (y_1, y_2, ..., y_m)$ be the activity vector of F_2. Initially, $\mathbf{x}^k = \mathbf{I}^k$ for k = 1,2, ...,n.

Weight Vectors: Let $\mathbf{w}_j^k$ denote the weight vector associated with the *j*th node in F_2 for learning the input pattern in F_1^k. Initially, the F_2 nodes are uncommitted and the weight vectors contain all 1's.

Parameters: Each field's dynamics is determined by choice parameters $\alpha^k \geq 0$, learning rate parameters $\beta^k \in$

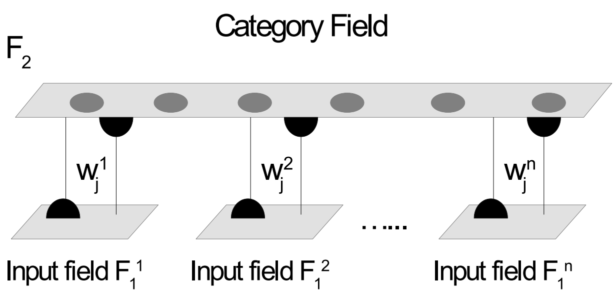

Fig. 1. Fusion ART architecture.(adapted from [12]).

$[0,1]$, contribution parameters $\gamma^k \in [0,1]$, and vigilance parameters $\rho^k \in [0,1]$.

Code Activation: Given the activity vectors $x^1, x^2, ..., x^k$, for each F_2 node *j*, the choice function T_j is calculated as:

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + |\mathbf{w}_j^k|} \tag{2}$$

where the fuzzy AND operation $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i), \tag{3}$$

and the norm $|.|$ is defined by

$$|\mathbf{p}| \equiv \sum_i p_i \tag{4}$$

for vectors **p** and **q**.

Code Competition: The F_2 node with the highest choice function value is selected by a code competition process. The winner is indexed at *J* where

$$T_J = max\{T_j : \text{for all } F_2 \text{ node j}\} \tag{5}$$

When a category choice is made on node *J*, y_J = 1; and y_j = 0 for all j $\neq$ *J*. This follows the winner-takes-all strategy.

Template matching: After the code competition process, the template matching takes place to check if resonance occurs. For each channel *k*, the match function m_J^k of the chosen node *J* is checked to see whether it meets a vigilance criterion.

$$m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k. \tag{6}$$

If any of the vigilance constraints is violated, mismatch reset occurs by setting the choice function T_J to 0 for the duration of the input presentation. The search process will keep selecting another F_2 nodes until a criteria is achieved. If no node in F_2 meets the vigilance, a new node in F_2 is created to represent a new category.

Template Learning: Once a node *J* is selected for firing, for each channel *k*, the weight vector $\mathbf{w}_J^k$ is updated by the learning rule.

$$\mathbf{w}_J^{k(new)} = (1-\beta^k)\mathbf{w}_J^{k(old)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)}). \tag{7}$$

Fusion ART network consists of multiple input and output fields and a category field. This makes fusion ART structures flexible to solve a wide range of problems. As the weight parameter **w** could be updated with every input pattern, the fusion ART architecture is suitable for online learning purpose. Another important feature of fusion ART is when no existing node is matched, the network could create a new node to represent the new pattern. This feature makes Fusion ART structure self-organizing.

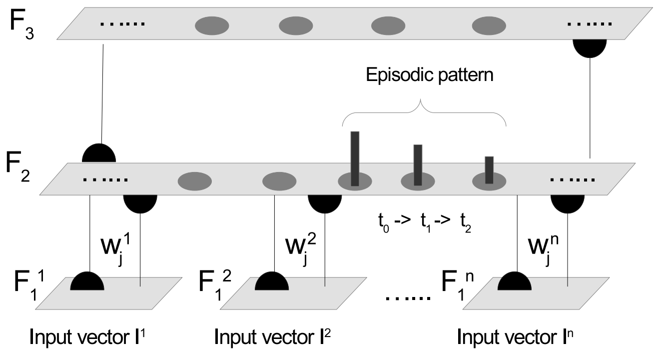

Fig. 2. EM-ART architecture.(adapted from [15]).

B. EM-ART

The Episodic Memory ART (EM-ART) [15] is an extension of fusion ART proposed to model episodic memory. The proposed architecture of EM-ART is shown in Figure 2.

Since EM-ART is an episodic memory inspired variation of fusion ART, it could learn episodic traces of sequences of sensory input and code the patterns into nodes in a higher category layer. Experiments [15] have shown that EM-ART is able to achieve a high level of memory performance with good robustness while managing memory usage over time.

There are three layers F_1, F_2, and F_3 in EM-ART model. As a result, there are two main mechanisms linking them: $F_1 \rightarrow F_2$ and $F_2 \rightarrow F_3$

The input data structure for the EM-ART are fixed-length sequences of sensory pattern inputs: Let $\mathbf{E} = (\mathbf{S}^1, \mathbf{S}^2, ..., \mathbf{S}^t)$ denote an episodic input that consists of a sequence of *t* instant states **S**. And each instant state **S** consists of the sensory input vectors from all the *k* channels that $\mathbf{S} = (I_1^k, I_2^k, ..., I_n^k)$. where $I_i^k \in [0,1]$ indicates the input *i* from channel *k*.

The operation for $F_1 \rightarrow F_2$ is similar to the standard fusion ART operation. It goes through the steps of code activation, code competition, template matching, and optionally template learning. For each input pattern in F_1 layer, either an existing code in F_2 layer is activated or a new code is created to represent the new input pattern.

Instead of using learned pattern codes as input, the EM-ART architecture uses the sequence of code activations in F_2 layer as the input for the second process $F_2 \rightarrow F_3$. To code the sequence of activations in a single episodic input vector **E**, the EM-ART first defines the time length *t* of each episode, so that each episodic input vector **E** consists of *t* sequenced input states S^t. Each time a code in the F_2 layer is activated by a F_1 layer input, the system decrease all existing code values in **E** and put the index of current activated code in front of $\boldsymbol{E}$ with value 1. The $F_2 \rightarrow F_3$ mechanism goes through the standard fusion ART processes same as that of $F_1 \rightarrow F_2$.

In the original EM-ART described in [15], the activation value is decreased exponentially as $y_j^{(new)} = y_j^{(old)}$ (1 - τ), where y_j is the activation value of the *j*th node in F_2 and

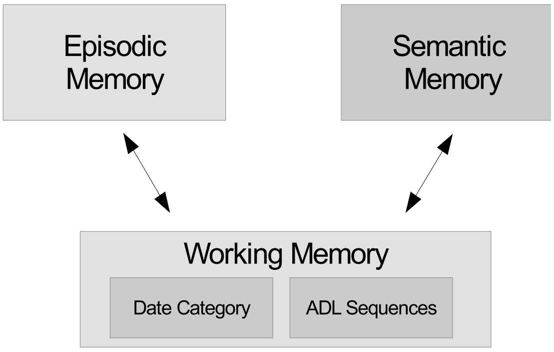

Fig. 3. The proposed ADLART model

Value	Meaning
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday
8	Public Holiday
9	Sick Day
10	Vacation
11	Special Days

TABLE I
DATE INFORMATION INPUT

$\tau \in (1, 0)$ is the decaying factor. In other words, it forgets very fast at the beginning and becomes slower later on. This pattern of activation decay is similar to human forgetting pattern. One good feature of this decay algorithm is that when the system performs a code read out operation, the sequence of states in F_2 could be easily retrieved by just looking at the value order of the F_2 codes in the activation vector, that the indexed code with the highest activation value always means the most recent event in the episodic. Due to the exponential decay feature, the decay is faster at the beginning. As a result, the more recent code activations have higher weighting in pattern comparison. For many applications, this feature could be valid as it is consistent to human-like forgetting pattern. However, it may not fit some other applications. Beside that, the activation sequence vector cannot represent multiple occurrence of same event in an episode. The reason is that as described in the algorithm, the latest occurrence of one state will overwrite its former occurrence record. There are several ways to work out of this problem. One possible way is to use another input channel to separate the possible repeated states in to many distinct states. Another possible way is to create another code in F_2 layer specially to represent the second occurrence of a state.

IV. ADLART

In this paper, we propose a multi-memory model, ADLART (Activities of Daily Living ART), which is inspired by human memory system, especially the episodic memory model, EM-ART, mentioned in the previous section. The architecture of the proposed ADLART is shown in Figure 3. The ADLART model contains a working memory component with two input fields, e.g. date category and ADL sequences. The working memory communicates with the episodic memory component to learn the episodic of ADL sequences, at the same time, it works with the semantic memory component for semantic extraction.

Inside the episodic memory sub-model (Figure 4), there are two input channels contributing to the process. The ADL sequence input field has activation vectors of ADL sequences of a day.

$$\mathbf{S^{ADL}} = (I_1^{ADL}, I_2^{ADL}, ..., I_n^{ADL}). \tag{8}$$

where the value of $I_i^{ADL} \in [0, 1]$ indicates the order of ADL sequence within the day and *n* is the total number of ADL categories. The ADL recognition and activation process goes beyond this architecture's scope and may be realized by using various ADL recognition algorithms, e.g. [6] [7].

At the same time, the date input field has vectors $\mathbf{S^{Date}}$ indicating the date information as tags.

$$\mathbf{S^{Date}} = (I_1^{Date}, I_2^{Date}, ..., I_{11}^{Date}). \tag{9}$$

In the current research and experiments, We have defined eleven types of days as listed in Table I.

The output is an ADL day category representation $\boldsymbol{A}$ which stores instances of categories of $\mathbf{S^{ADL}}$ with tags of the day information. Many applications could be made based on $\boldsymbol{A}$, and it could be used as the input for higher level analysis.

The mechanic of the ADLART algorithm is summarized in Algorithm 1.

Algorithm 1 ADL Sequence Coding and Routine Learning

for each ADL J in ADL sequence **A do**
  let node activation $I_J^{ADL} \leftarrow \frac{ADL\, occurrence\, time}{Whole\, day\, time}$
end for
for each ADL sequence vector $\mathbf{y}$ containing $\mathbf{S^{ADL}}$ and its corresponding Date Information vector $\mathbf{S^{Date}}$ **do**
  select a resonance node J' in F_2 based on sequence vector $\mathbf{y}$
  learn its associated weight vector as $\mathbf{w'}_J^{(new)}$ = $\mathbf{y}$ if **A** is a novel ADL Sequence
end for

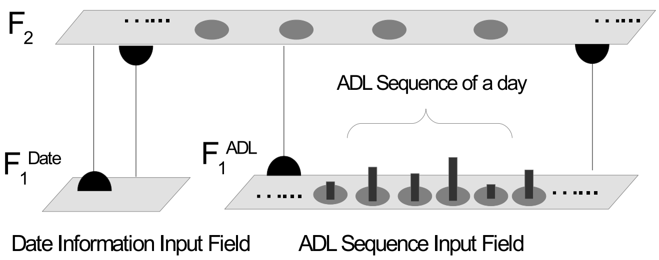

Fig. 4. The episodic sub-model

To encode ADL action time information into the input data structure, we propose to use time of day as the activation strength of $I_i^{ADL} \in [0, 1]$. The time of day is calculated as $\frac{ADL\,occurrence\,time}{Whole\,day\,time}$. It represents time in minutes. A whole day has 1440 minutes. For example, if a person dresses up at 8:00am, the activation strength value for dressing up will be 8*60/1440 = 0.333; if he has lunch at 12:25pm, the activation value for lunch will be (12*60+25)/1440 = 0.517. By using this representation, the time of day is encoded into the ADL sequences. Small variations in ADL occurrence time will be reflected as small difference in activation value, which will be tolerated in the template matching algorithm mentioned in the previous section. On the other hand, by using the activation strength assignment, it is very easy to retrieve the time of day and the sequence order for each individual ADL. At the end of a day, the most recent ADLs will have a bigger activation value, and the earliest ADLs in the morning will have smaller activation value. This is consistent with episodic memory forgetting trends. However, one issue in this setting is that the ADLs happening around the midnight will have very different values. The ADLs before midnight will have activation values near 1 while ones just after midnight will have activation values near 0. We will propose methods to handling this issue in future work.

A good feature of ADLART model is that it could learn semantics of the day models. As we propose two input channels for each daily routine category, the date information input could be treated as a tag for the routine models stored inside routine category F_2. If the date input vector is not fuzzy, there are few ways to implement semantic daily routine formation inside the routine category layer. The first and simplest way is to store strictly one category for each date type inside F_2 layer. To achieve this, all we need to do is to prevent new category creation in F_2 when no node meets the vigilance. This is not a good way as it will limit the categories in F_2, and hinder the ability of category learning. The second way is to tolerate different routine categories created for each day type, and use an average value (or weighted average for fuzzy dates) of all these categories to represent the semantic date type. In addition, each time when a category is fired, its ADL sequence and occurrence time could be updated with the new entry. We use a third way that creates a dedicated semantic component to store the semantic information.

$$\mathbf{S}^{Sem} = (A_1, A_2, ..., A_{11}). \tag{10}$$

where A_n is the semantic routine (ADL sequence) stored in F_2 for date type *n*.

Each time when a routine category is activated, the semantic ADL sequence value for the same date type is updated by

$$A_n^{new} = (1 - \beta_n^{Sem})A_n^{old} + \beta_n^{Sem}(\mathbf{S}^{ADL} \wedge A_n^{old}) \tag{11}$$

where the $\beta_n^{Sem} \in [0, 1]$ is the semantics update parameter for date type *n*.

The algorithm for semantics learning is summarized in Algorithm 2. The algorithm first select the matching category in F_2, and then update semantic memory according to the equation 11.

Algorithm 2 Daily Routine activation and Semantics Learning

for each input pattern of ADL sequence as vector $(\mathbf{S}^{ADL}, \mathbf{S}^{Date})$ in F_1 **do**
  Activate every category j in F_2 by choice function $T_j = \sum_{k=1}^{n} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha^k + \mathbf{w}_j^k|}$
  select category J such that $T_J = max\{T_j :$ for all F_2 node j$\}$
  set node activation $y_J \leftarrow 1$
  while match function $m_J^k = \frac{|\mathbf{x}^k \wedge \mathbf{w}_J^k|}{|\mathbf{x}^k|} \geq \rho^k$. (not in resonance)
    or J was selected previously **do**
    deselect and reset J by $T_J \leftarrow 0, y_J \leftarrow 0$
    select another node J with $T_J = max\{T_j :$ for all F_2 node j$\}$
  end while
  if resonance occurs(Routine recognized) **then**
    Update episodic category(template learning) $\mathbf{w}_J^{k(new)} = (1 - \beta^k)\mathbf{w}_J^{k(old)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)})$.
    Update semantic memory by $A_n^{new} = (1 - \beta_n^{Sem})A_n^{old} + \beta_n^{Sem}(\mathbf{S}^{ADL} \wedge A_n^{old})$
  else(No matching/resonance)
    let $J \leftarrow J^0$, where J^0 is a newly recruited uncommitted nodes in F_2
    Learn $(\mathbf{S}^{ADL}, \mathbf{S}^{Date})$ as a novel event with $\mathbf{w}_J^{k(new)} = \mathbf{S}^k$
  end if
end for

The proposed ADLART model fulfills the design challenges described in the previous section that the individual ADL and ADL sequences are represented as single data entry and data vectors. They have distinct definitions from each other. The activation strength design makes sure the model is tolerant to certain level of data variations inside same cluster, while the choice parameter α^k enables ADLART to differentiate patterns. In F_1^{ADL}, the ADL sequences are formed and in F_2 field, routine recognition takes place which compares the incoming entry with stored categories. Also, it could recall the sequence of recognized ADLs. Last but not least, it maintains a semantic routine memory in F_2 field as well.

There are some other algorithms could learn sequential patterns, for example Hidden Markov Model (HMM). If we represent daily ADL sequences using HMM models, the ADLs will be the HMM states. There is a difficulty to encode ADL happening time information into HMM model. Without ADL time, the accuracy of HMM model in solving ADL daily routine is relatively low compared to ADLART. From common sense, we could see that some ADL sequences may be shared by all types of days, e.g. weak up, go toilet, wash

ID	ADL
1	Wake Up
2	Wash Up
3	Dress Up
4	Breakfast
5	Entertainment
6	Lunch
7	Social
8	Dinner
9	House Work
10	Sleep

TABLE II

ADLS USED IN EXPERIMENT

ADL	Time
Wake up	6:00am - 8:00am
Wash up	wake up + 15m±5m
Dress up	wake up + 30m±10m
Breakfast	8:00am-9:00am
Entertainment	9:30am-10:30am
Lunch	11:30am-13:30pm
Social	no
Dinner	17:00pm - 19:00pm
House Work	14:00pm - 20:00pm
Sleep	21:00pm-23:00pm

TABLE III

SIMULATION STATISTICS: MONDAY

face, dress up, break fast, etc. The only difference of them in different types of days could be the happening time of such ADLs.

V. EXPERIMENTS

A. Experiment Settings

To test the performance of the proposed ADLART architecture, we designed a simulation environment. We first interviewed five people for their daily routine on different type of days. Based on their ADL time ranges, we write a programme to automatically generate data entries with some random variations. For example, a person has a standard morning routine, no social activities, and has an entertainment ADL in the evening for working days. At the same time, he has a delayed morning routine with both social and entertainment ADLs possibly ranging for the whole day in weekend and public holidays.

For the day category, as mentioned in the previous section, we have eleven types of days including Monday to Friday, weekends, sick days, public holidays, and vacations. For the ADL category, in the experiment we use a simplified version which consists of ten types of ADLs listed in the Table II.

B. Vigilance Parameter

In the first part of the experiment, we test the influence of different vigilance parameter values in creating new category codes for a given type of day. We use Monday as an example, of which the activities are listed in Table III.

As we are looking at the ADL patterns, the day category is used as tags by setting the vigilance parameter ρ^{Date} to 0.

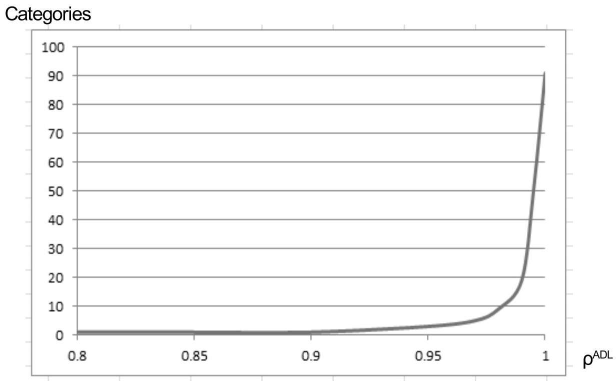

Fig. 5. Vigilance parameter vs. categories created in 100 samples

We use the simulation programme to generate 100 samples of Monday routine according to the variance mentioned in Table III. The relationship between the vigilance parameter ρ^{ADL} and the number of categories created is plotted in figure 5. We could see that when ρ^{ADL} is smaller than 0.95 there are only one or two categories created. This is not a good setting as one category could not represent the variation and distribution of the day type. When ρ^{ADL} increases towards 1, the number of categories created accelerates towards the number of samples. When ρ^{ADL} is set to 0.98 to 0.99, there are about 10 categories created. We prefer this number of categories as they could reasonably represent the variations of the day type, and at the same time avoid overfitting. When ρ^{ADL} increases over 0.99, there are more than 40 categories created, in other words, a higher risk of overfitting.

C. Combined day type experiment

In this experiment, we used five types of days to show the performance of the ADLART model: Monday, Tuesday, sick day, vacation, and Sunday. We specially designed their relationships as follows: The Monday and Tuesday routines are similar to each other that they are basically weekdays. They have starting time differences in ADLs but they have heavy overlap to each other. The Sunday, vacation, and sick Day are quite similar to each other that they have a delayed morning routine, vacation and sick day do not have breakfast, social, and house work. Besides, in sick days, the person sometime do not dress up (stay on bed). In the training phase, we use simulation generator generates 100 samples for each type of days. The vigilance parameter ρ^{ADL} is set at 0.99. The training sequence is Monday, Tuesday, Sunday, vacation, and sick days. After training, we use the same simulation generator generates another 100 samples for each type of days. We use the correctly recognized days over the 100 samples as the accuracy (recall). We run 10 times the experiment. The average result is summarized in Table IV. The result is as expected. Monday and Tuesday generators use similar sample distribution to generate data, and the model learn Monday first. As a result, Monday has a higher accuracy whereas the model sometimes mistake Tuesday data for Monday. The Sunday, Sick day, and vacation are well

ADL	Recall Accuracy
Monday	98.4%
Tuesday	66.4%
Sunday	100%
Sick Day	98%
Vacation	100%

TABLE IV

ACCURACY IN COMBINED EXPERIMENTS

ADL	Monday	sick day
Wake Up	7:14	11:52
Wash Up	7:29	12:51
Dress Up	7:46	
Breakfast	8:38	
Entertainment	10:4	
Lunch	12:45	12:52
House Work	17:49	
Dinner	18:07	18:20
Sleep	22:18	23:13

TABLE V

SEMANTIC MEMORY LEARNING

distinguished from other days.

D. Semantic Routine Learning

As mentioned in the previous section, we have learned a semantic memory of the daily routine in ADLART. In this experiment, we generated 200 samples of Monday routine and 200 samples of sick day routine. After training the system, the content of the semantic memory is listed in the Table V. The results satisfy the design requirement that over the time the system has learnt semantic pattern of a typical day of the user, e.g. compare Table III and Table V. At the same time, the most recent user activities contribute more on the semantics pattern which is very helpful for user health trend monitoring.

VI. CONCLUSIONS

In this paper we address a new problem of user daily activity pattern learning, and propose a multi-memory model called ADLART to attempt to solve this problem. Having an episodic memory component and a semantic memory component, ADLART is able to learn, recognize, and retrieve episodic and semantic patterns of user activities for different date types. Simulation experiments are conducted to show the functionality and performance of the proposed ADLART model. In the future, we will look at the issues in the current work, and at the same time, looking for other modeling alternatives to improve performance.

REFERENCES

[1] WHO, "10 facts on ageing and the life course," Apr. 2012.

[2] F. J. S. Salces, M. Baskett, D. Llewellyn-Jones, and D. England, "Ambient interfaces for elderly people at home," in *Ambient Intelligence in Everyday Life*, pp. 256–284, Springer, 2006.

[3] S. Roley, J. DeLany, C. Barrows, S. Brownrigg, D. Honaker, D. Sava, V. Talley, K. Voelkerding, D. Amini, E. Smith, *et al.*, "Occupational therapy practice framework: domain & practice," *The American journal of occupational therapy.: official publication of the American Occupational Therapy Association*, vol. 62, no. 6, p. 625, 2008.

[4] M. Lawton and E. Brody, "Assessment of older people: self-maintaining and instrumental activities of daily living.," *The Gerontologist*, vol. 9, no. 3, pp. 179–186, 1969.

[5] R. Bucks, D. Ashworth, G. Wilcock, and K. Siegfried, "Assessment of activities of daily living in dementia: development of the bristol activities of daily living scale," *Age and ageing*, vol. 25, no. 2, pp. 113–120, 1996.

[6] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 1, pp. 838–845, IEEE, 2005.

[7] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[8] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 10–19, ACM, 2008.

[9] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, 2010.

[10] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.

[11] A.-H. Tan, G. A. Carpenter, and S. Grossberg, "Intelligence through interaction: Towards a unified theory for learning," in *Advances in Neural Networks–ISNN 2007*, pp. 1094–1103, Springer, 2007.

[12] A.-H. Tan and H.-S. V. Soon, "Concept hierarchy memory model: A neural architecture for conceptual knowledge representation, learning, and commonsense reasoning," *International Journal of Neural Systems*, vol. 7, no. 3, p. 305, 1996.

[13] A.-H. Tan, "FALCON: A fusion architecture for learning, cognition, and navigation," in *IEEE International Joint Conference on Neural Networks, Proceedings. 2004.*, vol. 4, pp. 3297–3302, IEEE, 2004.

[14] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 230–244, 2008.

[15] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk, "Neural modeling of episodic memory: Encoding, retrieval, and forgetting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, 2012.