

A Location Based Sequence Prediction Algorithm for Determining Next Activity in Smart Home

Mohammad Marufuzzaman*, Mamun Bin Ibne Reaz, Labonnah Farzana Rahman and Araf Farayez

Department of Electrical, Electronics and Systems Engineering, Universiti Kebangsaan Malaysia, 43600 Bangi, Malaysia,

Received 5 November 2016; Accepted 12 May 2017

Abstract

Smart home or home automation has become widely popular especially in the case of easing the lives of people with special needs, for instance the elderly and handicapped people. In every home, a specific user has a unique pattern or sequence of using the functions of that house. Recognizing that unique pattern is the key to ensuring an intelligently and properly automated household where the house will remember the behavior of a user and predict the next service required by the user successfully. In this research, a recently developed algorithm named as sequence prediction via enhanced episode discovery (SPEED) is considered for modification by inclusion of location agents. A smart home prototype consisting of two rooms is designed as a testbed for verification. The results show that the accuracy of this algorithm is more than 40%, which is better than the previous SPEED. Moreover, the algorithm detects the location of next predicted event. Since human activity can be distinguished by their existing locations, predicting the next event as well as the location helps to determine the next action more accurately.

Keywords: artificial intelligence; hidden markov models; prediction by partial matching; smart homes; location agent

1. Introduction

In the modern era, health and social care improvements help to increase life expectancy worldwide. Moreover, according to the World Health Organization at least 650 million people live with disabilities around the world [1]. A growing number of these handicapped and older adults require a greater focus on policies and resources to meet their needs. Smart home technologies encourage and allow elderly and handicapped people to live in their own homes for a longer period [2]. Smart home technology started more than a decade to introduce the concept of networking devices and equipment in the house [3]. According to the Smart Home Association, the definition of smart home is the integration of technology and services through home networking for a better quality of living [2]. Early research on smart home concentrated on appliance intelligence, rather than focused on human behavior [4]. However, a robust smart home needs to collect data about the user's behavior in the environment and further analyze them [5]. A smart home should consciously observe the user activity sequences and try to predict the next most probable event [6]. Therefore, an algorithm, which can effectively make predictions of the users' future activities, should be developed [7]. After observing a sequence of events or activities, the smart home should be able to recognize the periodic patterns of those events and train itself to predict the next most probable event, and this event can then be automatically executed [8].

Recently, Alam et al. developed an algorithm known as sequence prediction via enhanced episode discovery (SPEED), which used the appliances' event patterns and

predicted the most probable future event [9]. SPEED algorithm showed better accuracy compared to some recent algorithms like C4.5, incremental probabilistic action modeling (IPAM), LeZi Update and Active LeZi (ALZ) [10-13]. SPEED worked with the episodes of smart home events that had been extracted based on the different states of home appliances. Prediction by partial matching (PPM) algorithm was applied to the extracted episodes that arranged in a finite order Markov model. In this way, the next user activity was predicted from the previous sequences. However, human activity patterns are usually affected by some other components such as location and time [14]. Marufuzzaman et al. modified the SPEED algorithm by including time component [15]. The algorithm not only included time component but also modified the previous SPEED algorithm by considering all possible episodes. However, the research still did not consider an important agent such as location. Human activity is completely changed based on their location of events. For example, a living space has two rooms each of which contains a cooler. However, the cooler in a bedroom is used far more than the cooler in the living room. Therefore, even if the time and the events ON/OFF states are considered, the future state cannot be predicted correctly, as we are not sure which room's cooler will be ON/OFF. The prediction accuracy will be much smaller if two or more similar appliances are in different locations. Therefore, it is essential to include location agents as well for predicting the future activities.

Several researchers work on location-based prediction. Maharaj et al. proposed an algorithm to predict an individual's real-time travel path and destination using location data from a Global Positioning System (GPS) enabled mobile phone [16]. The algorithm uses spatially aware, geometric representations of a user's historical trip data in its comparisons, thus eliminating the need for taxing

*E-mail address: marufust@gmail.com

calculations. The technique also promotes scalability using enterprise spatial database systems. A similar application proposed by Gambos, Killijian and Cortez using mobility Markov Chains is used to predict the next location of an individual based on the observations of his mobility behavior over some period and the recent locations that he has visited [17]. The research extended a mobility model called Mobility Markov Chain (MMC) in order to incorporate the previously visited locations and to develop a novel algorithm for next location prediction based on this mobility model that coined as n-MMC. Rotariu, Manta and Ciobotariu described the designing of an integrated system for monitoring patients suffering from cardiac arrhythmias [18]. The research is based on a wireless sensor network (WSN) and is able to measure and transmit the patient's heart rate (HR) by radio to a central telemonitoring station. However, all of these algorithms mainly located the human instead of their activities. The human activities need to be predicted in a smart home environment. Therefore, it is important to recognize the state of the events as well as the locations of occurrences. The prediction can be more accurate if we know the location of the individual events.

In this research, the very recent algorithm SPEED is modified by including the location agent into the algorithm. As SPEED algorithm produces much better accuracy compared to other algorithms, therefore, the prediction accuracy of this algorithm is compared with SPEED. An experimental house is also described where the dataset is generated and is used for validation.

2. Methodology

This research modified the SPEED algorithm by including the location agent in order to predict the next probable event along with their location. The sequence in previous SPEED algorithm has no location agent. Therefore, the sequence is generated accordingly first then PPM based algorithm is used to predict the next event.

In a home environment, the inhabitant does numerous tasks repeatedly every day. If each of these tasks is considered as an event, then these repeated events are generated a specific temporal pattern. Moreover, these sequences of events have different patterns at different locations. Consider there are two rooms in a home, one is a kitchen and the other is a bedroom. The inhabitant enters into the bedroom and turns ON the light. After that, he turns ON the light in the Kitchen and uses the cooker i.e. turns it ON. After finish cooking, he turns OFF the cooker and the light and comes back to the bedroom. He then turns ON the TV and the fan. Finally, he turns OFF all the appliances and leaves the bedroom. By analyzing the scenario, it can be noted that the inhabitant activities maintain a pattern on different locations. Therefore, the next activity can be predicted based on this pattern and intelligence can be applied to the home appliances. Assume an example contains the appliance states and their corresponding locations as, *A1B1A2b1C2c2D2a2a1d2A1A2*.

Here, each letter represents one appliance. The capital letters indicate that the particular appliance is ON while the small letters represent its OFF state. The number preceding each character represents its corresponding location. Afterward, the episodes are extracted from the sequence. An episode is created with two opposite states of events. For example, from the earlier sequence, the first element, A1

indicates the starting point of an episode, while the ending point of that episode is indicated by the ninth element, a1. Using the location information, the previous episode is not ending in the eighth element, a2 as the starting appliance location is different. In a practical smart home, if an appliance such as light is common in different rooms then opposite state, as well as the location of the light, is important to build an episode. Since the starting of the episode is turning ON the light in a room, it should end with turning OFF the light in the same room. The episode cannot be generated if the light is turned OFF in a different room. Figure 1 shows the pseudo code to convert the raw sequence into a new sequence so that the episode is generated based on the locations as well.

```

initialize j = 1
loop until length of sequence
j is even number
if raw_sequence(j) = 1
process_sequence(j-1) = raw_sequence(j-1)
else
process_sequence(j-1) = Symbol (ASCII values of
raw_sequence(j-1)+ 9)
j = j+2;
end loop
    
```

Fig. 1. Pseudo code to convert raw sequence

In order to differentiate the same event in a different location and to minimize the memory needed, the events in different locations are assigned with different letters. Depending on the location, the ASCII value of that particular event is changed and therefore the symbol is changed as well. For example, consider the sequence above, the element A2 becomes J while A1 becomes A. The conversions of the other elements are shown in Table 1. The ASCII value is increased by nine steps to accommodate three rooms containing a maximum of eight appliances each. In this research, the data are collected from two rooms and each room has maximum three appliances. Therefore, the algorithm is sufficient for generating proper sequences.

Table 1. Conversion of sequence based on location

Original elements	Converted element
A1/a1	A/a
A2/a2	J/j
B1/b1	B/b
B2/b2	K/k
C1/c1	C/c
C2/c2	L/l
D1/d1	D/d
D2/d2	M/m
E1/e1	E/e
E2/e2	N/n
F1/f1	F/f

After applying the above-mentioned algorithm, the original sequence is changed as, *ABZbYyXzaxAZ*. After this conversion, episode generation from the original sequence became much simpler. Now an episode starts with the first element A and ends with the ninth element a. The internal numbers are not effected on generating episodes. An episode, *ABZbYyXza*, is constructed and this episode contains some sub-episodes as well such as *Yy*, *Xzax*, *BZb*. In this process, all these episodes are generated. After detecting an episode, the algorithm generates all possible

contexts and their frequencies of occurrences as shown in Table 2. In this algorithm, it is necessary for detecting all episodes to make an accurate prediction of the next events. This is because the algorithm only generates possible context after an episode is detected, and the prediction of next event strongly depends on the entire possible contexts and their frequency of occurrences. After this step, the

algorithm generates a (k-1)th order Markov model, where k equals Max_Window_Length. After defining all possible contexts of the episode, a decision tree is generated to organize the episodes. The probability distribution for each of the events to happen is calculated as well. After that, the decision is made by choosing the event with the highest probabilities.

Table 2. All possible contexts with their frequencies

Seq	Current event	Window after current event	Max Window Length	Episode	Max Window Len. after episode extraction	Window after episode extraction	Possible context from the episode	All possible context with frequency of occurrence
1	A	A	1	Not Found	1	A		A(2), a(3), B(1), b(1),
2	B	AB	1	Not Found	1	AB		x(2), X(2), Y(1), y(1),
3	Z	ABZ	1	Not Found	1	ABZ		Z(2), z(2), BZ(1), Zb(1),
4	b	ABZb	1	BZb	3	BZb	B, Z, b, BZ, Zb, BZb	Yy(1), Xz(1), za(2), ax(2), aX(1), XA(1),xA(1), AZ(1), BZb(1), Xza(1), zax(2), aXA(1), axA(1), xAZ(1), Xzax(1), zaxA(1), axAZ(1), zaxAZ(1)
5	Y	BZbY	3	Not Found	3	BZbY		
6	y	BZbYy	3	Yy	3	bYy	Y, y, Yy	
7	X	bYyX	3	Not Found	3	bYyX		
8	z	bYyXz	3	Not Found	3	bYyXz		
9	a	bYyXza	3	Not Found	3	bYyXza		
10	x	bYyXzax	3	Xzax	4	Xzax	X, z, a, x, Xz, za, ax, Xza, zax, Xzax	
11	A	XzaxA	4	aXA	4	zaxA	a, X, A, aX, XA, aXA	

Figure 2 shows a tree diagram generated by considering all possible contexts with their frequency of occurrences as shown in Table 2. The tree nodes contain the filtered symbols as well as their frequencies.

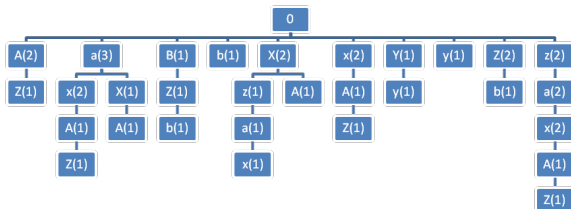


Fig. 2. Trie formed by the Algorithm after processing the Sequence

PPM method is used to calculate the probabilistic values of the entire possible context uniformed in the decision tree. The method actually measured the weighted probability distribution according to the episode length. The method calculates any event probability by decreasing the length of an episode [19]. Consider the previous ABZbYyXzaxAZ sequence and assume that the window is now at a state of zax. Suppose that we want to calculate the probability of the next event A, with all the possible contexts i.e. zax, za, z, and null. According to the tree in figure 2, A occurs once out of two times in occurrence of zax. Therefore, the probability of A to occur in the context of zax is 1/2. Now we escape to lower order with escape probability of 1/2, where escape probability is multiplied by lower order episode to reduce their influence on the result. For the context za, it is noticed that no A occurs after them, so the probability is 0 and the probability of a null outcome is also 0. Lastly, we consider

the context z, and A never comes after z. Hence, the probability of A is also 0. Again, we go to a lower level and we need to consider the escape probability, which is 1/2 since the total occurrence of z is 2. Going for the next context, which is null, the probability of A to occur is 2/17 as the total occurrence of all events is 17 and total occurrence for A is 2. Using this information, the probability of A as the next event is calculated as shown in equation 1.

$$\frac{1}{2} + \frac{1}{2} \left[\frac{0}{2} + \frac{1}{2} \left\{ \frac{0}{2} + \frac{1}{2} \left(\frac{2}{17} \right) \right\} \right] = 0.515 \quad (1)$$

Next, if we are going to calculate the probability of Z as the next sequence, the calculation will be like equation 2.

$$\frac{0}{2} + \frac{1}{2} \left[\frac{0}{2} + \frac{1}{2} \left\{ \frac{0}{2} + \frac{1}{2} \left(\frac{2}{17} \right) \right\} \right] = 0.015 \quad (2)$$

Using the same formula as above, we can calculate the probabilities of other events, which is A, a, B, b, X, x, Y, y, and z. The event with the highest probability is chosen as the next probable event. After predicting the event, corresponding location is stated as well. For example, if the next event is A then it is stated as “Event A in room 1”, whereas if the next event is Z then it is stated as “Event A in room 2”. Therefore, the method not only predicts the next event but also predicts the location. Predicting the particular event with the location will certainly make the accuracy much better than that of only events.

Smart home algorithms can either be tested using the dataset provided by fully functional smart homes available in the world or by creating an experimental setup and collecting the data from that home. To compare the accuracy results with a similar environment, this research implements a very simple yet adequate prototype of a functional smart home. A two-room house is rented for one year to set up a smart house and the data is stored for further uses. The rent is provided by the ministry of higher education, Malaysia (Escience grant code 01-01-02-SF0744). The house has two interconnected rooms, and there were few appliances in each room. One room contained one light, one fan and a TV while the other room contained one light, one fan, one cooker and one toaster. The home thus contains two common appliances in both rooms. The photograph of the experimental home setup is shown in figure 3.



Fig. 3. Basic Experimental setup of the testbed

One of our laboratory students acts as 'the resident' during the experiment. The participant is introduced to the smart home testbed and is guided through a preliminary task to familiarize with the layout of the premises. The participant is then asked to perform a sequence of activities. Every day the participant performs the activities in the morning and in the evening. Both weekdays' and weekends' activities are recorded. It is obvious that, a working person spends his/her time in the home mostly in the morning after waking up from the bed and in the evening after returning home until he/she falls asleep. Therefore, the data is collected in the morning session and in the evening session. Obviously, weekend activities are different from weekday activities, and the dataset of this prototype reflects that occurrence too. To simplify the network of appliances, in

this research, PowerUSB Smart Switches (SS) are used to communicate with the computer. In this research, the PowerUSB SS is used from PH Technical Labs. The PowerUSB is designed for using with a computer to control the power of peripheral electronics. The SS has four outlets. Among them, one outlet is not controlled and is always in ON state. The other three outlets can be controlled through software to power on external devices. The unit is connected to the computer through USB cable. The current sensing feature of the PowerUSB lets the user know how much power is consumed by the PowerUSB at a given instance and over a period of time [20]. The "ReadCurrentPowerUSB()" function is used to measure the load in the SS outlet. By using this feature, appliance states are stored. The Visual Studio software is used to call the API functions. The current sensing method senses the current of the SS through USB ports. Therefore, by analyzing the sensed current, it can be easily calculated that which appliances are now in ON/OFF state. In every 30 sec, the current is being sensed, and the statuses of all those appliances are stored in a file as testbed dataset.

All these seven appliances are connected to the laptop through the USB port. In every 30 seconds, the data is stored in a text file. Every day the user does some activities by utilizing some appliances and the data is stored. A sequence filter method is used to discard any redundant data and generate the sequence with only unique changes of events. This filtering saves lot of spaces as well as helps to execute the algorithm run faster. More than 500 sequential states are generated using this testbed and this research is used the dataset for verification.

3. Results and Discussion

In order to validate the method of this research, the dataset from the practical smart home is used for both SPEED and this algorithm. As mentioned earlier SPEED is more accurate compared to other research works thus this research compared the accuracy with SPEED. The accuracy graph is shown in figure 4 where the x-axis represents the number of patterns and the y-axis represents the accuracy.

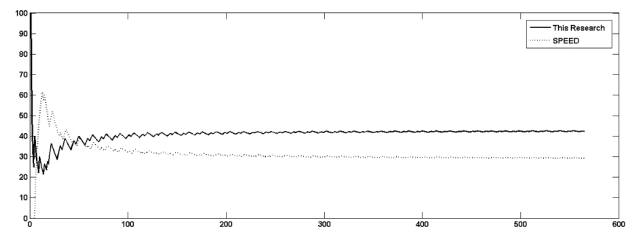


Fig. 4. Accuracy comparison results between SPEED and this research.

The accuracy measurement is done by comparing the value of predicted event with the next event of the sequence. If the predicted event is the same as the next event then accuracy counter is increased by one, otherwise it remains same. Lastly, the accuracy counter is divided by the total number of sequences and multiplied by 100% to measure the accuracy in percentage. In figure 4, it is seen that, initially, the accuracy of SPEED is about 60%, which is higher than this work. This is because only one room data is stored in the testbed file. However, after using data from two room it is observed that the accuracy of SPEED is lower i.e. 35%

while with the location agent the accuracy is higher which is 40% and is stable.

SPEED actually calculated the prediction accuracy based only on the appliance states. Therefore, if some appliances are common in a different location, SPEED fails to predict that next state correctly. For example, in this practical home environment, one light and one fan were common in two rooms. SPEED distinguishes the light and fan as different event but fails to distinguish differently located lights or fans. Moreover, after finishing the probabilistic calculation, SPEED only predicts the next probable event not the location of that event. It is understandable that only appliances' states are not sufficient to predict the next probable activities. The decision of the next probable event is more accurate if it considers their location as well. Hence, using location agent and converting the sequence accordingly are definitely going to produce more accurate results than that of the previous SPEED algorithm.

4. Conclusion

Predicting the next ADL certainly plays a key role in the smart home environment. Moreover, effective classification of episodes improves the prediction accuracy. In order to

identify effectively, the episodes should consider all the appliances' states as well as their locations. In this research, a very recent algorithm SPEED is modified and location agent is included to improve the accuracy. Moreover, a testbed is designed in order to collect the data and test the location agent-based algorithm. The results reveal that using location agent the accuracy is about 40% while without location agent the SPEED accuracy is below 35%. The performance of this algorithm shows a clear improvement in accuracy using the same testbed data. Moreover, the algorithm can successfully predict the location of the next event. Improving the prediction accuracy eventually makes smart homes more secure and comfortable, especially for the older and handicapped people.

Acknowledgement

The authors would like to express sincere gratitude to Ministry of Higher Education, for sponsoring this research work under Escience grant (01-01-02-SF0744).

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence



References

1. Tordrup, David, and Roberto Bertolini. "Research Agenda for Health Economic Evaluation." 2013.
2. M. A. Al-Qutayri and J. S. Jeedella, "Integrated Wireless Technologies for Smart Homes Applications", Smart Home Systems, Mahmoud A. Al-Qutayri (Ed.), InTech, 2010, doi: 10.5772/8412.
3. C. Douligeris, "Intelligent home systems". IEEE Communications Magazine, vol. 31, no. 10, 1993, pp.52-61, doi: 10.1109/35.237984.
4. A. M. Vainio, M. Valtonen and J. Vanhala, "Proactive Fuzzy Control and Adaptation Methods for Smart Homes", IEEE Intelligent Systems, vol.23, no.2, 2008, pp.42-49, doi: 10.1109/mis.2008.33.
5. P. Rashidi and D. J. Cook, "Keeping the Resident in the Loop: Adapting the Smart Home to the User," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 39, no. 5, 2009, pp. 949-959, doi: 10.1109/tsmca.2009.2025137.
6. M. Marufuzzaman, M.B.I. Reaz, M.A.M. Ali and L.F. Rahman, "Classification and Detection of Intelligent House Resident Activities using Multiagent", Proc. of the 4th Int. Conf. on Computing and Informatics, Sarwak, Malaysia, 2013, pp. 195-200.
7. M. Marufuzzaman and M. B. I. Reaz, "Hardware Simulation of Pattern Matching and Reinforcement Learning to Predict the User next Action of Smart Home Device Usage," World Applied Sciences Journal, vol.22, no.9, pp. 1302-1309, 2013.
8. M. Marufuzzaman, M. B. I. Reaz, and M.T. Islam, "FPGA based Distributed Task Organizing Agents in Smart Home", Proc. of the 2nd Int. Conf. on Research in Science, Engineering and Technology, Dubai, pp. 1-5 doi: 10.15242/iie.e0314504.
9. M.R. Alam, M.B.I. Reaz, and M.A.M. Ali, "SPEED: An Inhabitant Activity Prediction Algorithm for Smart Homes", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 42, no.4, 2012, pp.985-990, doi: 10.1109/tsmca.2011.2173568.
10. Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine Learning, 16(3), 235-240. doi:10.1007/bf00993309.
11. K. Gopalratnam and D.J. Cook, "Online sequential prediction via incremental parsing: The active LeZi algorithm", IEEE Intelligent Systems, vol. 22, no. 1, 2007, pp. 52-58, doi: 10.1109/mis.2007.15.
12. A. Bhattacharya and S. K. Das, "LeZi-update: An information-theoretic framework for personal mobility tracking in PCS networks", Wireless Networks, vol.8, no.2/3, 2002, pp. 121-135 doi: 10.1023/A:1013759724438.
13. L.S. Zhang, M.J. Yang, and D. J. Lei, "An Improved PAM Clustering Algorithm Based on Initial Clustering Centers" Applied Mechanics and Materials, vol.135-136, 2012, pp. 244-249, doi: 10.4028/www.scientific.net/amm.135-136.244.
14. A. Venkatesh, "Smart Home Concepts: Current Trends, Center for Research on Information Technology and Organizations I.T. in the Home", University of California, Irvine, Year 2003 Paper 377, Retrieved from: <http://eprints.cdlib.org/uc/item/6t16p6pf>.
15. M. Marufuzzaman, M.B.I. Reaz, M.A.M. Ali and L.F. Rahman, "A Time Series Based Sequence Prediction Algorithm to Detect Activities of Daily Living in Smart Home", Methods of information in medicine, vol.54, no.3, 2015, pp. 262-270, doi: 10.3414/me14-01-0061.
16. N.P. Maharaj, S.J. Barbeau, M.A. Labrador, P.L. Winters, R. Pérez and N.L. Georg-gi, "Real-Time Travel Path Prediction Using GPS-Enable Mobile Phones". Center for Urban Transportation Research, Department of Computer Science and Engineering, University of South Florida, 2008.
17. S. Gambs, M.O. Killijian and M.NdelP. Cortez, "Next Place Prediction Using Mobility Markov Chains", Proc. of the First Workshop on Measurement, Privacy, and Mobility, 2012, doi: 10.1145/2181196.2181199.
18. C. Rotariu, V. Manta, R. Ciobotariu, "Integrated System Based on Wireless Sensors Network for Cardiac Arrhythmia Monitoring," Advances in Electrical and Computer Engineering, vol.13, no.1, pp.95-100, 2013, doi:10.4316/AECE.2013.01016
19. J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," IEEE Trans. Commun., vol. 32, no. 4, pp. 396-402, Apr. 1984, doi: 10.1109/tcom.1984.1096090.
20. "PowerUSB-Basic", [Online <http://www.pwrusb.com/powerUSB-basic.html>