Daily Activity Learning from Motion Detector Data for Ambient Assisted Living

GuoQing Yin and Dietmar Bruckner
Institute of Computer Technology
Vienna University of Technology, Austria, Europe
{yin, bruckner}@ict.tuwien.ac.at

***Abstract*- In an intelligent environment one important task is to observe and analyze person's daily activities. Through analyzing the corresponding time series sensor data the person's daily activity model should be build. To build such a model some problems have to be overcome: the sensor data count increase sharp with time and the distribution of the data is dynamically according the person's daily activities.**

In an Ambient Assisted Living (AAL) project we handle this kind of time series sensor data from a motion detector. At first we reduce the data count through a predefined threshold value and build data "states" in time interval. Secondly, we analyze the states using a hidden Markov model, the forward algorithm, and the Viterbi Algorithm to build the person's daily activity model.

To test the correctness of the model some special and random day's activities routine will be given.

***Keywords*: Forward algorithm, hidden Markov model (HMM), intelligent environment, Viterbi algorithm.**

I. Introduction

There are many papers about Markov chain and hidden Markov model: [1] introduced Markov chain and stochastic stability in detail. The basic definition of Markov chain and the hidden Markov model explained in [2], furthermore the applications of HMM. In paper [3] the authors describe a technique to learning the number of states and the topology of hidden Markov model from examples. Here the Bayesian posterior probability was used for choosing the states to merge and for the stopping criterion. The author of paper [4] describes the EM Algorithm and parameter estimation for hidden Markov model. In the paper [5] the author attempts to review the hidden Markov model from the theoretical aspects and shows how the hidden Markov model has been applied in speech recognition. The hidden Markov model and the related structures of probabilistic independence networks searched in [6]. In paper [7] a variable-length Markov models used to efficient representation of behaviors. Hidden Markov model used as Bayesian networks to the tasks of transmembrane protein topology prediction and signal peptide prediction described in paper [8]. The papers [9] and [10] adopt the hidden Markov model to analyze the motion detector data, to learn the behavior of the user. In paper [9] the authors take advantage of semantic symbols, build probability model in building automation systems.

A. AAL Background and ATTEND Project

Ambient Assisted Living (AAL) is a new search field, it focus on enhance the life quality of the elderly and prolong the independent living in the elderly own home with the help from modern technology. But because of the elderly have their own problems, such as action obstacles, memory disorder … how the elderly people can use the modern technological system?

Within the scope of the project ATTEND (AdapTive scenario recogniTion for Emergency and Need Detection) a system will be developed that increases the time frame of independent living of elderly persons in their used living environment. The system comprises an intelligent, adaptive network of sensors, which are to be installed in the living environment of the user in order to thoroughly observe his activities and behavior. An important aspect is that the sensors shall work independent and in a preferably invisible fashion.

ATTEND learns about normal activities and behavior of the user. In case of unusual activities and behavior an alarm plan can be worked out (e. g. enquiring the user, calling a neighbor, calling an external organization). The system is intended to increase comfort, security and social inclusion of the customer and ideally also help with the early detection of upcoming medical problems. In case of an emergency the system can contact primary and secondary users (family, neighbor, care giver) via external interfaces.

In this paper we use hidden Markov model, forward algorithm, and Viterbi algorithm to analyze the sensor data, for example the data from motion detector installed at a corner of the living room. The generally daily activities models about the sensors will be build and according the model if some unusual activities and behaviors happened, the system will send aware signal to user or alarm signal to neighbor or caregiver.

For example a user has activities in the morning, at noon and in the evening at the living room. A motion detector installed at a corner of the living room and it records the daily activities of the user. Using the hidden Markov model, forward algorithm, and Viterbi algorithm the activities model

978-1-4244-7561-2/10/$26.00 ©2010 IEEE

of the user will be build. Because the elderly person has relatively stable lifestyle but on the other hand there are always any different in the daily living. For example at weekend the user gets up a little later, has a little longer cleaning work or has visitors in the living room. So the activities of the user should be generally keeping the similar style but not exactly the same. That means the build activities model should keep the similar style but has different routines. Furthermore if any abnormal activities happened for example: in the morning at the living room there are without activities from the user, and this kind of situation never happened before, so the system should according the model to send an aware or alarm signal to user, neighbor, or care giver.

B. Basic Parameters

In the living room a motion detector installed, if there are activities from the person, the motion detector sends the value “1” to the controller. If there are without activities, the motion detector sends the value “0” to the controller. If there are continuous activities the value will keep in “1” till the activities halt and the motion detector sends value “0” to the controller. If there are continuous stillness the value will keep in “0” till the activities emerge and the motion detector sends value “1” to the controller. This is the basic function from the motion detector.

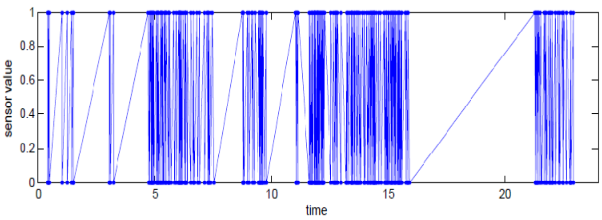

Fig. 1. The data from one motion sensor in one day.

Figure 1 shows the gathered sensor data value from one day, there are 364 values all. The x-axis is time and the y-axis is the sensor value. Generally in the morning (between 5 and 10), at noon (between 11 and 16) and at night (around 21 and 23) there are many activities, but between 0 to 5 o’clock and after 16 to 21 there are really quiet.

The questions about this kind of parameters are: first, there are too many parameters in each day; second, each value has the same importance but on the other hand each individual value doesn’t have great meaning. For example there is just seldom sensor value (activities) between 0 and 5 in figure 1. It perhaps comes from the user rolls the body in sleeping at night. If we think a step deeper: a few activities (for example rolling the body in bed or get up to the WC) in sleeping is normal but if there are too many activities – perhaps it means the elderly person has sleep disorders or other hidden physical health problems.

How can we decide “a few activities” and “many activities” with this kind of sensor data? An easy and useful idea is that: gather the activities in a time interval, if the sum value of activities bigger than a predefined threshold value, so this time interval will be treated as activities value “1”, others will be send value “0” to the time interval. Here the predefined threshold value and the time interval play an important role.

- The gathered sensor value (for example in 30 minutes)

$$T = \{t_{1\,(1)}, t_{2\,(0)}, t_{3\,(1)}, t_{4\,(0)}, t_{5\,(1)}, t_{6\,(0)} \ldots t_{n\,(v)}\} \qquad (1)$$

Here t_n is the time point that the motion detector send value to controller ($n >= 1$). V is the sensor value itself, it has value “0” or “1”.

- The activities between sensor value

$$\Delta T = (t_{n\,(0)} - t_{n-1\,(1)}) \qquad (2)$$

- The sum of the activities in the time interval

$$T' = \sum(\Delta T) \qquad (3)$$

- Decide if the time interval gets value “1” or “0”.

$$\text{If } T' >= T_{th} * T_{interval},\ Q_{interval,\,ix} = 1 \qquad (4)$$

$$\text{If } T' < T_{th} * T_{interval},\ Q_{interval,\,ix} = 0 \qquad (5)$$

Here T_{th} is the threshold value, $0 < T_{th} < 1$. $T_{interval}$ is the predefined time interval, for example 30 minutes. $Q_{interval,\,ix}$ is the result value that the interval should take. Here “ix” is the interval count (index).

We predefine the $T_{interval}$ as 30 minutes, so there should be 48 intervals (that means $1 <= n <= 48$) one day. If the $T_{interval}$ chosen a smaller value the accuracy will be increased but the computational will be increased too. On the other hand if we chosen the $T_{interval}$ a bigger value, for example 120 minutes, there should be just 12 intervals each day, the accuracy will be reduced significant. The same situation happened with the threshold value T_{th}. If the T_{th} be chosen too small, some “noise” sensor values (activities for example the user rolls body in sleeping) will be translated to interval value $Q_{interval,\,ix} = 1$, but for example in fact the user only has activities about 10 seconds in 30 minutes; if the T_{th} be chosen too big, some activities will be depressed, in both situations it creates a related great deviation to the reality.

In figure 2 we chose 30 minutes as $T_{interval}$ and 0.2 as T_{th}.

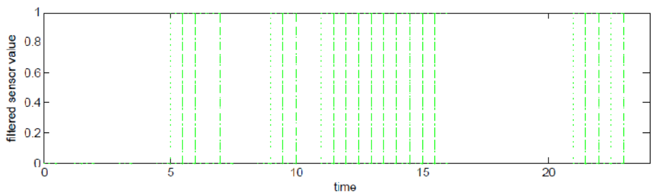

Fig. 2. Filtered sensor value

The x-axis of figure 2 is time and the y-axis is the filtered sensor value. The interval begin with the dotted line and end of the dash-dot line (or between dash-dot lines) indicated in the interval has a value “1” (it means in the interval has activities), others has value “0” (it means in the interval has without activities). From the figure 2 we can see that between 0 to 5 o’clock there are without activities. It is not corresponding with the figure 1 – this is just the function from (1) to (5) – the “noise” value be filtered. On the other hand if there are many activities, such as between 5 and 10 o’clock, these activities will be translated to value “1” in time interval.

If we gather more filtered sensor data together, for examples for 12 days, we can get a more generally intuition of the data distribution and the activities of the elderly person. The result showed in figure 3.

From figure 3 we know that between 0 to 5 o'clock the elderly person has a few activities, between 5 to 10 there are a little more activities, from 10 to 15 there are most activities and after 15 the activities reduced and around 20 there are again more activities.

Till now we reduced the data account, send the value (activities) of each time interval, but this is just a generally intuition of the activities of the elderly person. Using the hidden Markov model, forward algorithm, and Viterbi algorithm we can get an activities model of the elderly person.

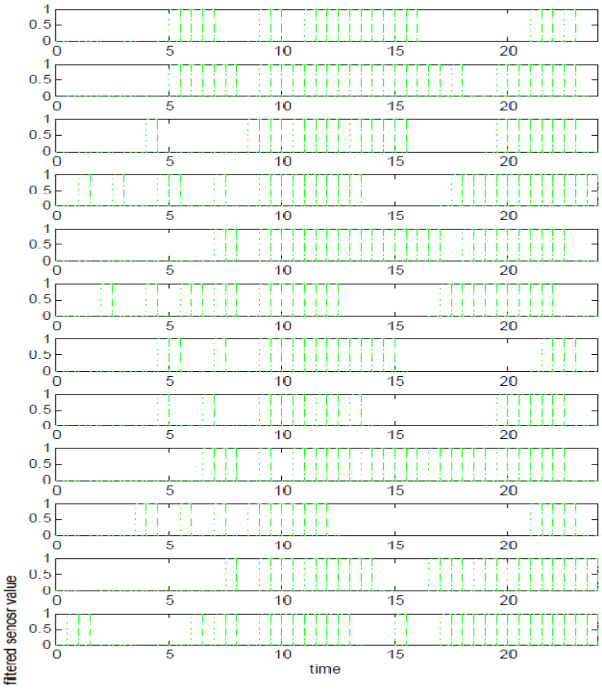

Fig. 3. Filtered sensor value in 12 days

II. Markov Chain And Hidden Morkov Model

From above we get the activities model for the elderly person. If we chose $T_{interval}$ as 30 minutes, so there should be 48 activities value one day, for example $Q_{interval,\,ix} = \{ Q_{interval,1} = 0, Q_{interval,2} = 0, Q_{interval,3} = 1, \ldots, Q_{interval,\,m} = 0, \ldots, Q_{interval,46} = 1, Q_{interval,47} = 0, Q_{interval,48} = 0\}$. Here $1 <= m <= 48$. If we treat each interval activity value (0 or 1) as "state", so there should be 48 states each day.

A. Markov Chain

There are some papers [1], [2], and [5] give the definition about the Markov chain. For us the discrete time first-order Markov chain is the most important. $P(Q_{t+1} = q_{t+1} \mid Q_t = q_t, Q_{t-1} = q_{t-1} \ldots Q_0 = q_0) = P(Q_{t+1} = q_{t+1} \mid Q_t = q_t)$. Here Q_t is a random variable from a countable state space at time t, q_t is the taken variable in a countable set at time t.

B. Hidden Markov Model

In Markov model each state corresponded to an observable (physical) event, it is too restrictive in many real situations. What we treat the problems are the observation is a probabilistic function of the state. This is called hidden Markov model: a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observation [5].

Connecting the theory and practice (i.e., hidden Markov model definition and the real application), we should at first build a hidden Markov model according the daily activities and then using the model to explain the observed sequence of daily activities. The first question we have to deal with is how to build a hidden Markov model. It needs some basic definitions and algorithm.

C. Basic Parameters of Hidden Markov Model

There are some parameters characterize hidden Markov model, for a better understanding, on the following we will consult the real application to explain these parameters.

- The number of states N
- The number of output distinct observation symbols each state M
- The state transition probability distribution matrix A $= \{p_{ij}\}$.

$$p_{ij} = p\,\{Q_{t+1} = j \mid Q_t = i\}. \quad (6)$$

$0 <= p_{ij} <= 1$ and $\sum_{j=1}^{N} p_{ij} = 1$, $1 <= i, j <= N$. Here Q_t is the current state at time t.

- The state emission probability distribution matrix $B = \{b_{ik}\}$.

$$b_{ik} = p\,(O_t = k \mid Q_t = i),\ 1 <= i <= N,\ 0 <= k <= M \quad (7)$$

Here O_t is the output symbol at time t.

- The initial state distribution $\pi = \{\pi_i\}$.

$$\pi_i = p\,\{Q_o = \mathrm{i}\}. \quad (8)$$

Now we will explain these parameters step by step.

- The number of states N

As we predefined above if $T_{interval}$ chosen as 30 minutes, so there should be 48 activities value one day. If we treat each activity as a state there should be 48 states, but because of merging different states the states count will be reduced. There are two different situations that the states can be merged together.

The first situation is that merging the identical states:

For example in some time intervals in one day the activity value of the user keep on "1", $Q = \{\ldots 1, 1, 1, 1, 1, \ldots\}$, because the state transition probability value between state t and state t+1 keep on 100% and the state emission probability value keep on the same, so these states can be merged in to one state. In the merged state there are 2 parameters: $P(Q_{ii})$ and $P(Q_{ij})$.

$$P(Q_{ii}) = N / (N + 1) \quad (9)$$

$$P(Q_{ij}) = 1 / (N + 1) \quad (10)$$

Here $P(Q_{ii})$ is the "self-transition" probability, $P(Q_{ij})$ is the transition probability, and N is the number that is merged states. $P(Q_{ii}) + P(Q_{ij}) = 1$. In the situation $b_{ik} = 1$.

The second situation is that merging the consecutive states and these states has different alternately states value:

For example in some time intervals on one day the activity value of the user is $Q = \{\ldots 1, 0, 1, 0, 1, 0 \ldots\}$, the value "1" and "0" appear alternately. It has the 100% state transition probability value between state t and state t+1 and the emission transition probability with increased states count closer to 0.5. All these states could merge into one state. The parameters $P\ (Q_{ii})$ and $P\ (Q_{ij})$ computed as above but the emission transition probability is different.

$$b_{i0} = N_0 / N \quad (11)$$

$$b_{i1} = N_1 / N \quad (12)$$

Here N_0 is the count that all the states have value "0" and N_1 is the count that all the states have value "1". It is clearly $N_0 + N_1 = N$.

- The number of output distinct observation symbols each state M

Here are only 2 distinct observation symbols "0" and "1". So $M = 2$.

- The state transition probability distribution matrix A

We have discussed the transition probability distribution in the merging situation above. Another situation is the split situation: from time interval t to the next time interval $t+1$ there is more than one state connected with the same state Q_t. For example in state Q_t there are 10 values, all these values are "1". In the next time interval t+1 there has a state that has 4 values and all the 4 values keeping "1" and another state has 6 values and all the 6 values are "0". So the state transition probabilities are 0.4 and 0.6 separately.

- The state emission probability distribution matrix B

This has been discussed above.

- The initial state distribution π

For example there are 2 initial states π_1 and π_2, π_1 include 3 values, all are "0" and π_2 include 7 values, all are "1", so $\pi = \{\pi_1, \pi_2\} = \{0.3, 0.7\}$.

III. The Forward Algorithm And The Viterbi Algorithm

Given a hidden Markov model that means the parameter (π , A, B) are known, how we can find the probability of an observed sequence $Q^{(i)} = \{q_1, q_2, \ldots, q_t\}$? Here each of the q is one of the observable set. The forward algorithm is used.

- Get the first transition probability a_1 for $t = 1$.

$$a_1\ (j) = \pi\ (j) * b_{jt} \quad (13)$$

Here j is the observation count of each observation set and $\sum \pi\ (j) = 1$.

- For t >= 2 get the transition probability $a_{t+1}\ (j)$

$$a_{t+1}\ (j) = \sum_{i=1}^{n} (a_t\ (i) * a_{ij}) * b_{jt} \quad (14)$$

- For $t <= T$ repeat (14).

Here T is the length of the sequence.

Most of the time we are not only need the probability of an observed sequence but also need to find out the best interpretation of the observation. In this situation the Viterbi algorithm will be used. The essence point of the algorithm is finding the maximum value of each step.

- For $t = 1$

$$a_1 = argmax_j\ (\pi\ (j) * b_{jt}) \quad (15)$$

- For $t >= 2$

$$a_t = argmax_j\ (a_{t-1} * b_{jt}) \quad (16)$$

IV. Result And Conclusion

A. Result

The basic data come from a motion detector installed in a living room. We gathered the data for 12 days and use the formula (1) to (5) to translate these data to 48 states (30 minutes as one state interval) for each day. Figure 3 is the activities value for each state in all 12 days. For avoiding the different days' sequences crossing we sort these days at first and then merge the same activities at the same time interval. Fig.4. shows the result.

Fig. 4. Filtered sensor value in 12 days

In figure 4 each small cycle means a state, the activities value is "0" or "1". From left to right if at the same time interval there has the same value, these states will be merged together till different state value appeared, then one state split into two states (two paths). From figure 3 we know that at the first time interval there are only value "0" in all 12 days, so in figure 4 the first time interval merged to only one state with value "0". At the time interval 2 there are value "0" and "1" in figure 3, so in figure 4 at the time interval 2 there are 2 states. The states value not showed in figure 4 but according figure 3 we can judge that the last sequence of figure 4 is the last day in figure3 (with value 0, 1, 1,…) and the one before the last in figure 4 is the fourth day in figure 3 (with value 0, 0, 1, 0, 0, 1,…).

Figure 4 is just for showing the data structure and there are always too many states should be merged later.

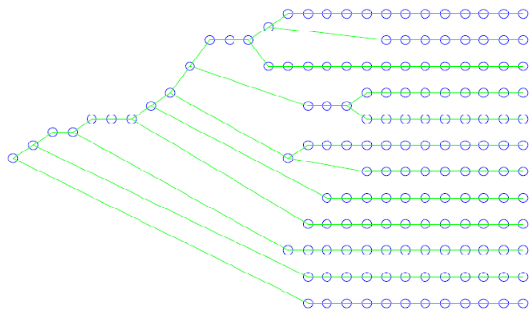

Fig. 5. States after merging in 12 days

According the formula (9) and (10) the identical states and the states with alternately value will be merged together. Figure 5 showed the merging result with forward algorithm. Compare figure 5 with figure 4 it is clearly that they have the same structure but the states count reduced.

Till now the hidden Markov model with parameter $\lambda = (A, B, \pi)$ are build. With the model we can calculate the probability of the observation sequences, find out the best "explains" of the observation, with Viterbi algorithm to find out the best parameters in each step.

Figure 6 shows the logarithm value for a chosen observation sequences in the 12 days. The logarithm value comes from the transition probability value according the A and B matrixes. X-axis is the states for each day (or the sequences number on each day); y-axis is the logarithm value of the transition probability. In figure 6 the sequences with cyan cycle is the best day that adapt to the chosen day. It has a logarithm value -17.15 after 48 steps. In fact they are the same days. On the other hand the sequence with magenta star is the worst day that adapts to the observation sequences. It has a logarithm value -103.1 after 48 steps. Because the parameters in A, B matrix all smaller (or equal to) than 1, so the logarithm value reduced step by step.

It is often that in some step there are without states value adapt to the observation sequence value, for example in step "s" the observation sequences has value "1", but in the same step the state has value "0", that means the state sequences cannot adapt to the observation sequences. In generally the state sequences should be defined "not adapt" to the observation sequences, and the comparing should be stop at once. The observation sequences should be comparing with other new state sequences. But in the real application what we meet is that: perhaps there just a few steps did not adapt to the observation sequences, and then the state sequences adapt to the observation sequences again. So in such situation we send a probability 1% to the transition matrix A in the step, to allow the comparing continues on. In figure 6 the logarithm value with steep drop connections are this kind of situations.

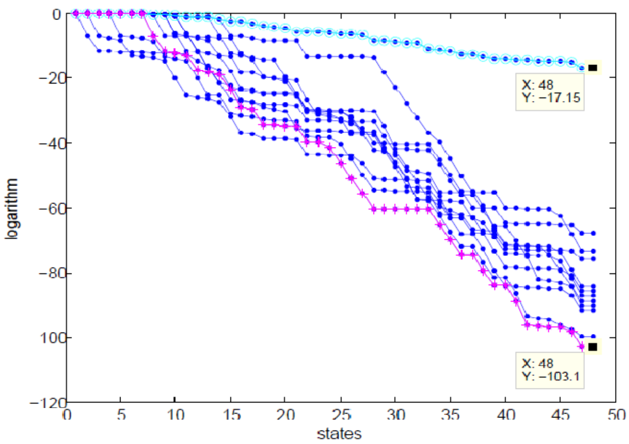

Fig. 6. Compare the logarithm with chosen observation sequences in 12 days

Figure 7 shows the logarithm value for a random observation sequences in the 12 days. Because it is a random observation sequences there are many step not adapt to the hidden Markov model, so the logarithm value reduced generally. The result is the best sequences adapt to the observation sequences has a logarithm value -76.42 and the worst logarithm value is -130.3.

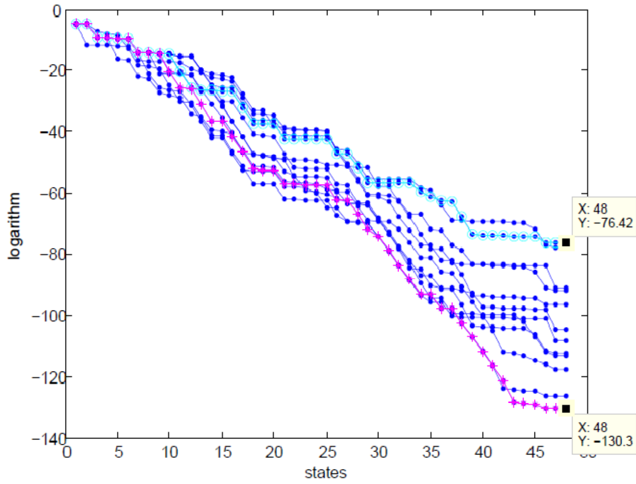

Fig. 7. Compare the logarithm with random observation sequences

From figure 7 we know that in the best adapt sequences (the sequences with cyan cycle) there are not each step has the maximum logarithm value. Using the Viterbi algorithm we can get the best sequences that adapt to the observation sequences. The result showed in figure 8. In figure 8 the sequences with green triangle is the best sequences adapt to the observation sequences. It comes from different states sequences.

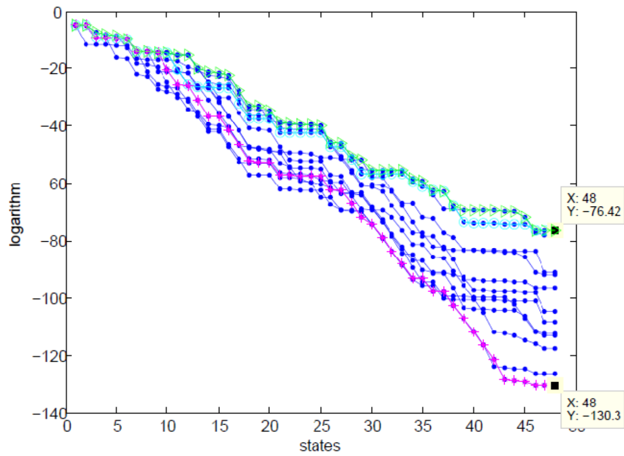

Fig. 8. Compare the logarithm with random observation sequences and find the best value in each state

B. Conclusion

From the above presented result we can say that hidden Markov model, the forward and Viterbi algorithm is a powerful tool for unsupervised learning and very useful in real-world applications in the AAL context.

For example if we build a hidden Markov model according the activities of the user for some days and through adapt to some chosen days we will get a transition probability value (then change it to logarithm value) range. If for a new observation sequences the logarithm value is in the range that means the user has the normal activities but if the logarithm value beyond the range, that means the user has abnormal activities. For example in the morning the activities value is keeping on "0", it cannot adapt to the hidden Markov model, so the logarithm value will be reduced rapidly. In such situation the system will send messages to the user or alarm signals to a neighbor or caregiver.

For the application of AAL the stable life style of the user is the basic of a useful learning result.

V. Outlook

Till now we just used the hidden Markov model analyzed one sensor data from some days, in the future different sensor data and different algorithm will be tried to analyze the behaviors of the user. Because of different life style directly influence the model, so by different elderly with different life style the build model should be different. This work will be done with more data come from different elderly in the future. Some more robust learning algorithm should be developed and tested. Furthermore the life style changing caused by hidden health problem will be searched, too.

References

[1] S.P. Meyn and R.l. Tweedie, "Markov Chains and Stochastic Stability", Springer-Verlag, September, 2005.

[2] J. Bilmes (2002). "What HMMs Can do", UWEE Technical Report, Number UWEETR-2002-0003, January 2002.

[3] A. Stolcke and S. Omohundro, "Hidden Markov Model Induction by Bayesian Model Merging", Berkeley CA 94704, advances in Neural Information Processing System, 5, San Mateo, CA, Morgan Kaufman, 1993.

[4] J.A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its application to Parameter Estimation for Gaussian Mixture and hidden Markov Models", international computer science institute, Berkeley CA, 94704, April 1998.

[5] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", proceedings of the IEEE, Vol. 77, NO. 2, FEBRUARY 1989.

[6] P. Smyth, D. Heckerman and M.I. Jordan, "Probabilistic Independence Networks for Hidden Markov Probability Models", May 1, 1996.

[7] A. Galata, N. Johnson, and D. Hogg, "Learning Variable-Length Markov Models of Behavior", Computer Vision and Image Understanding 81, 398-413 (2001).

[8] S.M. Reynolds, L. Käll, M.E. Riffle, J.A. Bilmes, W.S. Noble, "Transmembrane Topology and Signal peptide Prediction Using Dynamic Bayesian Networks", PLoS Computational Biology, November 2008, volume 4, Issue 11, e1000213.

[9] D. Bruckner, B. Sallans, and G. Russ, "Probability Construction of Symbols in Building Automation Systems", in: Proceedings of 2006 IEEE International Conference of Industrial Informatics INDIN' 06, S. 6, Singapore, 2006.

[10] D. Bruckner, B. Sallans, and R. Lang, "Behavior Learning via State Chains from Motion Detector Sensors", Bionetics, 2007, Hungary.