3P1-IOS-2a-1

Developing a Bayesian Framework for Human Behavior Tracking

Younes El Hamdi[*1 *2] Takumi Okamoto[*2] Nak Young Chong[*2] Il Hong Suh[*3]

[*1] University of Genoa, Italy [*2] JAIST, Ishikawa, Japan [*3] Hanyang University, Korea

Abstract: The problem of tracking human activities of daily living is considered an important subject that crosses a broad spectrum of disciplines. In this paper, we study the problem of creating an inference mechanism to recognize and respond to human behavior. We provide probabilistic methods to build a new Bayesian framework to deal with human tracking problem. Specifically, we present a technique to extract the structural features from data sensors and provide a set of algorithms to encompass the learning solutions in order to cope with unreliable and noisy measurements. The framework permits automatic activity tracking of daily routines of the inhabitants in a closed environment such as smart homes. Unlike almost all of related works, we propose an efficient algorithm for sensing systems that presents an alternative to sensors that are sometimes perceived as invasive, where notably we do not use vision-based learning. Preliminary results show that the proposed system can be deployed in different environments and significantly outperforms existing methods in a very reliable manner.

***Key Words*:** smart home, activity tracking, training, machine learning, computational complexity, supervised learning

1. Introduction

The advent of new sensors and communication technologies characterized by tiny and low cost devices opens new horizons to develop robust home automation systems. This trend continues to proliferate in conjunction with the new generation of electric household appliance that can be integrated in order to respond to the demands of such applications as tracking human activities in a smart home. The activity tracking focuses mainly on what are the mechanisms by which the system will generate a coherent interpretation to diverse sensor measurements caused by human activity, and tries to identify the activity engaged by the inhabitants to provide the right service at the right time in the right place. However, a number of problems arise due to the uncertainty that characterizes the human behaviors described below:

1. Normal and routine happenstance change across different people, resulting from different habits as well as gender or age differences.
2. Variability of the activity within the same person: people tend to alter their habits in various situations or contexts depending, *e.g.*, on whether he/she is in a hurry or just changes plans when interrupted for some reasons.
3. Time: how people behaves differently on weekdays and weekends under various situations. The frequency and sequential order of a certain activity may change.
4. Environment: the inhabitants adapt their behaviors accordingly to the arrangement of furniture.
5. Weather: people take shower more often in summer.

A key challenge is to understand and analyze the human activities. In recent years, several research projects have been initiated [1,2,3,4]. The most common approaches to modeling human activities consider the most frequently occurring activities used to measure the functional health of an individual [5]. Normally the approaches vary accordingly to the type of sensors used to monitor and on the type of model employed to represent the activities. While an activity model may vary depending on what kind of activities we are interested to track, the performance and the type of sensor information remain a critical aspect. In particular, the complexity of computations has been greatly underestimated. Almost all of existing works on activity recognition have focused on modeling and learning by entailing the sequential and temporal relationships between the observed events. Notable methods among them include Bayesian model [6], ontology methods using a semantic reasoner [7], or decision trees to learn the logical descriptions of the activities [8]. Other methods instead generalize the hidden Markov model by using the hierarchical hidden semi-Markov model [9], or HMM [10] with attempting to model the state duration in HMM. We retain the last approach unnecessary besides being computationally expensive. We believe also that the time duration is not so important at the end of the activity recognition and consider that the time duration could be estimated in different manners, without the necessity of the high precision required for speech recognition [11]. In this paper, we aim to build a complex model by developing thereby simpler components, and dealing efficiently with the complexity of computing. We present a new technique to extract the data features. There are few studies that have specified the system using solely discrete sensors, which reduces the possibility of the system to be deployed in our real life. We also consider a generic model to represent activities that could happen in our daily life. In particular, we propose an automated dynamic Bayesian model, which is motivated by the reduction of uncertainty that characterizes the measurements, and by our desire to address the above problems through a consistent Bayesian perspective. In a special case, we present here a dynamic selector of factorial HMM (DSFHMM), which is based on naïve Bayesian classifiers that help to select the desired model on the context location of the inhabitants.

The remainder of the paper is organized as follows: in Section 2, we discuss the problem evaluation and provide solution to some issues we presented in this section. Important assumptions

Contact: Y. El Hamdi, Japan Advanced Institute of Sci. & Tech., 1-1 Asahidai, Nomi, Ishikawa, younes.elhamdi@jaist.ac.jp

are considered to tackle these problems and to follow the rest of the paper. Section 3 gives a short introduction to the Bayesian network and introduces the DSFHMM model and its formulation for inference and learning. In Section 4, an experimental result for the task of activity recognition is presented and the conclusions are drawn in Section 5.

2. Problem statements and assumptions

The classical supervised learning problems are based on constructing a classifier that predicts the labels of the objects given sufficient training examples from the past observed objects which are considered the domain space H of all possible classifiers. In the vision of activity tracking, it is generally accepted to assume the space H including all activities classes that are performed in a consistent manner during the daily life of the inhabitants. The activities here are seen as a temporal sequence of human actions, in which every action can activate a particular configuration of sensor events located at specific home space. Any configuration of those events is maintained until a new activation will take place. However, the problem here is to provide a classifier function that maps from a sequence of configurations of sensor events to the class of activities. To formalize this task we need to make some assumptions we consider as follows.

2.1 Sensor events

Clustering data sensor by time and location is an important task in the activity tracking. We adopt an approach similar to that of a detective: the inhabitants when they are moving, activate different sensors located on the surrounding area, which generates a sequence of sensor events that will constitute our clues on predicting the nature of the activities that are taking place. Each configuration state of sensor events lasts normally for a period of time and terminates by switching to another according to the progress of the activities inside the home. For convenience, we consider only discrete sensors to represent the events. Therefore, if we have n binary sensors with two discrete values, we get 2^n events. Sensors that emit continuous value for example could be used through discretization to divide all interesting values into a set of intervals of interest. The actions are performed by the users while they are performing their daily tasks like moving around, cleaning, eating, to name a few. In the learning phase, the sensor events are considered as the action features variable Y_t whose state space is the set $Y_t = \{e_1, . . . ,e_N\}$ of unit (column) vectors $e_i = (0, . . . ,1,0, . . . ,0)$ of $\boldsymbol{R}^N$, which are *simplex.* We denote by e_i a sensor event. Using a configuration of n discrete sensor events we get $y_t = (e_1,e_2,…,e_n)$.

2.2 An illustrative example: washing hands

We consider an illustrative example to demonstrate how to find a solution to the problems. We assume a house equipped with all the necessary sensors, we imagine the inhabitant, henceforth called user, enters in the bathroom and activates the following sensor events: while the timer accounts for the time elapsed from the starting of the activity, a location status event results "Bathroom", a water usage status event is "ON" indicating a "specific area" of the bathroom. If we suppose to know when the activity starts, how can we predict its end? The most obvious answer to this question is to say when the user turns off the tap. But what about if the user opens and closes the tap while he is washing his hands? It is hard to admit that he washes them many times consecutively. Therefore, many scenarios can come true and others however are unlikely. We need to assume a criterion to define both an activity and how it can end.

2.3 Activity class

The activity gives rise to a sequence of action features which is regarded to as a realization of different length T of an unknown random process. To exploit the sequential patterns in the action features, we denote by an activity class i all the sequences of action having the same length T_i and the elements. All preselected activities are stored and clustered on N_a activity class, each class i contain n_i activities. Henceforth we denote by $X_{t,i}$ a random variable belonging to the activity class i.

2.4 Mean time duration

Back again to the above example, normally the user washes and dries his hands and leaves then the bathroom. But what happens if for some unknown reason he decides to stay in the bathroom doing probably nothing? This signifies that something is going wrong and it could be useful if we can report it, especially when we monitor elderly persons. In that case, we consider the mean time duration T_{m} spent by the users to perform their activities as information that could be useful and sometime a criterion for the decision making (see problem 1, 3). We consider also the end of each activity as the beginning of another. Furthermore, we can label also the tasks that are not pre-selected for the activity tracking by assigning them a default name. That is, if a user is doing nonsignificant task, it says that he is doing an "unknown" activity until it is proven otherwise.

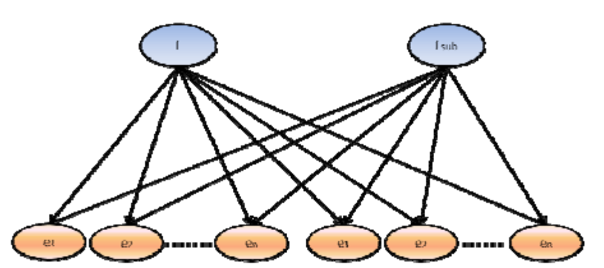

Figure 1 Two Naïf Bayesian Classifiers coupled for localization

2.5 The context of location

To reduce the computational complexity, we associated a set of an equal number of sensors to each location context and indicate it by the variable U. This variable will play an important role to enable the desired activity models applied to recognize which activity class is taking place. U is also partitioned into two other variables: the first one is L which represents the global location, and the second one is L_{sub} to represent the local sub-location. If L assumes for example a value such as kitchen, L_{sub} will assume a value such as refrigerator area. However, to encode a particular information such a composite activity (problem 2), where two individuals are seen to perform simultaneously the same task, we add a new activity class to the pre-selected activities and increase the number of sensors to avoid any sensor collisions. That is, if more than two activities are engaged simultaneously or more than one individual perform the same task, we can follow the same procedure. To implement the localization algorithms, we use the naïve Bayesian Classifiers shown in Fig. 1, which are efficient and also one of the most widely used methods.

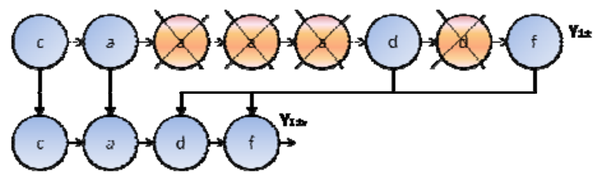

Figure 2: compressing online the observed action sequence

2.6 Data preprocessing

The sequence of actions $Y_{1:t}$ performed up to time t consists normally of a number of consecutive occurrences of actions. We retain that is unnecessary for the purpose of learning and do not carry any information that might be relevant. However, to speed up the decoding and learning stages, we keep just the first element of all consecutive actions y_t on the sequence $Y_{1:t}$ having the same value (see Fig. 1). To this end, we define the following function:

$$Y_{1:t_V} = F(Y_{1:t}) = \begin{cases} Y_{1:t-1} & if \quad y_t = y_{t-1} \\ Y_{1:t} & if \quad y_t \neq y_{t-1} \end{cases} \Bigg| \quad \forall t > 1 \qquad (1)$$

The timing of the activity class process follows the last element of the new sequence $Y_{1:tv}$. That is, a pair ($X_{tv,i}$, Y_{tv}) running in $Y_{1:tv}$ are without consecutive occurrences in the action Y_{tv} . This will improve greatly the time complexity and will render the flow depending on the new timing.

3. Model definition and implementation

Bayesian networks, also called Belief networks, are graphical structures for modeling the probabilistic relationships among a large number of nodes (or variables) representing a set of propositions regarding the structure of a given domain. The arcs signify direct dependencies between the linked propositions, and the strengths of these dependencies are quantified by conditional probabilities called the parameters. Each proposition, will be assigned a measure of belief. A dynamic Bayesian network is an optimal candidate to simulate a smart home, and emphasizes dynamically the interaction between the system and the environment.

3.1 Information flow

In light of all assumptions considered in the previous section, we formalize the problem by denoting an unknown activity i at time t with a variable $X_{t,i}$ whose state space have a size n_i , in which each every state is associated to the sequence order of the action consecutive on the time. It means that an activity i at timestep t generates a sequence of action features $y_{1:t}$ which is regarded as a realization of length t, generated *iid* from an unobserved homogenous Markov chain process $\{X_{t,i}\}$. The probability distribution on the state space of $X_{t,i}$, will generate values on the state space of the action features variable Y_t. At this point, we consider the two time-slices of the dynamic Bayesian model shown in Fig. 2. We model the probability distributions over the random variable $Z_t=(U_t, X_t, Y_t)$, where U_t represents the context location (see Section 2.4). The model is defined by the pair (A_1,$A_{\rightarrow}$), where A_1 is the prior probability distribution $P(Z_1)$ we assume uniform, $A_{\rightarrow}$ is the transition probability defined as follows:

$$p(Z_t / Z_{t-1}) = \prod_{i=1}^{N_a} P(Z_t^i / Pa(Z_t^i)) \qquad (2)$$

Z_t^i is the i'th node at time t, and includes the i'th component of the selected activity class with N_a the total number of pre-selected activities class. In light of Eq. (1) and (2), it seems reasonable to assume the model to be a first-order Markov, and represent all pre-selected activities through the persistent slices which are connected with arcs from the left to the right, which reflects naturally the causal flow of the time. The evidences from the context location U_t will play the fundamental role of selector by activating the persistent arcs flow between selected nodes which represent also the current activities class. Furthermore, special sensors are deployed to observe the actions in their respective context location. The data flows instant by instant reflecting the factual knowledge about the selected activity class.

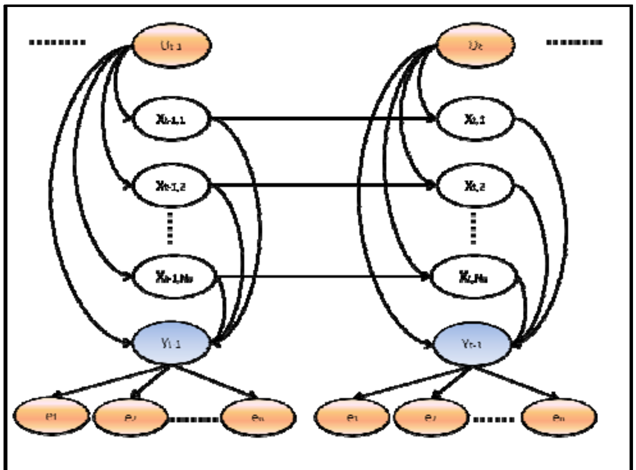
Figure 3 Two Time-Slice DBN - Activity class tracking

We can write the transition probability from Eq. (2) as:

$$A_{i,j,k} = P(X_{t,i} = j / X_{t-1,i} = k, U_t = L_i) \qquad (3)$$

which is the probability to make a transition in time t from the activity state k to state j. The probability to observe an action y given that the activity class i is on the state j can be written as:

$$B_{i,y,j} = P(X_{t,i} = y / X_{t-1,i} = j, U_t = L_i) \qquad (4)$$

The network in such a manner turns into a computational architecture due to the persistent links represented the selected activities due to the local context, and these links are used not merely for storing the factual knowledge about a given activity, but also for directing and activating the data flow in virtual timing. The manipulation of this dynamic knowledge will be executed in parallel to dynamic computations, which have the task to verify if one or more activities among the selected activities are completed. On this task, the most probable activity of the pre-selected activities is estimated by means of Viterbi's decoding algorithm VDA.

In the dynamic program of VDA, forming an ergodic HMM, the selected activity class *i* with n_i activities requires $O(n_i^2)$ at the time *t*. The whole sequence up to *t* require $O(t\ n_i^2)$. We denote by $v_{t,i}[j]$ the probability of the most probable sequence of states of the activity class *i* which has generated the sequence of actions $Y_{1:t}$ and is ended with the state *j*, we can write it as follows :

$$v_{t,i}[j] = B_{t,i,y,j}(y_t) \cdot \max_k (A_{t,i,j,k} \cdot v_{t-1,i}[k]) \tag{5}$$

which refers to maximize the following expression:

$$P(X_{1:T} / Y_{1:T}) \tag{6}$$

with $v_0[j] = 1/n_i$. The last expression will be clarified more later. The time duration can be evaluated when we reach the end of the activity. If we call *t* the current time and T_i the last time observed to complete the same activity *i*. We store the new mean time duration as follows:

$$T_{m,i} = \frac{T_{m,i} + t'}{2} \tag{7}$$

At this point we can reset the system.

3.2 Data acquisition

During the data acquisition stage, the activities are now considered the input which corresponds to a sequence of actions, and the sequence of action features will be assumed the output. The data was collected from villa Basilea, a protected residence for elderly, with an agreement with municipality of Genoa, Italy. This collection of data examples called dataset $D_{t,i}=\{(X_{1:t,i}, Y_{1:t,I}, U_{t,i}, t')|\ i=1:n_s\}^{1:Na}$ was separated accordingly to the generating activity chosen from the pre-selected categories. The users were supposed to be given a PDA endowed with a small custom program that displays tree images and the users were asked to select in a straightforward manner the tree icons including the following information; the start and ending of an activity class they were engaged to do and its respective name. Whereupon the relative frequency of the observed sequence of action features $Y_{1:t}$, namely the likelihood, was evaluated by following the probability rule $P(Y_{1:t}/X_{1:t,i})$, with which the sequence was produced, and simultaneously the mean time duration $T_{m,i}$ was updated. We observe that each activity class was considered on its variability.

3.3 Parameters learning

The second stage is to train the model parameters $\theta_i=(A_{i,j,k}, B_{i,y,j}, U_t)$ for activity class *i* that maximizes the likelihood $P(Y^*/\theta_i)$ of the observed training sequence of actions $Y^*=Y_{1:t'}$ from the training set $D_{t,i}$. Let $\underline{A}_{i,j,k}$ denote the number of times the state *j* follows the state *k* in the most likely sequence of states of the activity class *i*. Similarly, let $\underline{B}_{i,y,j}$ denote the number of times the action *y* is performed during the activity class state *j* in the most likely sequence. The examples $D_{t,i}$ collected as described above exhibits a sequential correlation we encode in the parameters θ_i by mean of Viterbi training. The updated parameters are given by:

$$A_{i,j,k} = \frac{\underline{A}_{i,j,k}}{\sum_{j'} \underline{A}_{i,j',k}} \tag{8}$$

$$B_{i,y,j} = \frac{\underline{B}_{i,y,j}}{\sum_{y'} \underline{B}_{i,y',j}} \tag{9}$$

We use Baum-Welch algorithm to Y^*. This is achieved at the first step by applying the subroutine forward and backward to the input data Y^* to compute all the following matrices:

$$back_{t,i,j} = \prod_{a=t+1}^{t'} K_i^{y_a} \cdot back_{t',i} \tag{10}$$

$$fwd_{t,i,j} = \prod_{a=1}^{t} K_i^{y_a} \cdot fwd_{0,i} \tag{11}$$

for every $1 \le t \le T_{m,i}$ and $1 \le j \le n_i$. The matrix is $n_i * n_i$ with elements:

$$K_{i,j,k}^{y} = B_{t,i,y,j} \cdot A_{t,i,j,k} \tag{12}$$

$fwd_{t,i,j}$ is the probability to observe a sequence of action features $Y_{1:t}$ and that at time *t* the activity $X_{t',i}$ is *j*, then

$$fwd_{t,i,j} = P(Y_{1:t}, X_{t,i}) \tag{13}$$

and $back_{t,i,j}$ is the probability to observe the sequence of actions $Y_{t+1:Tm,i}$ given that the state of the activity $X_{t,i}$ is *j*.

$$back_{t,i,j} = P(Y_{t+1:Tm,i} | X_{t,i}) \tag{14}$$

The algorithm employs $O(T_{m,i}\ n_i^2)$ to process all observed actions. We observe that the parameters θ_i encode insight the value of the context location U_t. That is, the second step is to re-estimate $\underline{A}_{i,j,k}$, and $\underline{B}_{i,y,j}$ through generalization over the time given by:

$$\underline{A}_{i,j,k} = \sum_t P(X_{t+1,i} = j, X_{t,i} = k / Y^*, \theta_i) \tag{15}$$

$$\underline{B}_{i,y,j} = \sum_t P(X_{t,i} = j / Y^*, \theta_i). \tag{16}$$

where $C_t = P(X_{t+1,i} = j,\ X_{t,i} = k\ /\ Y^*,\ \theta_i)$ is the probability that a transition from state *k* to state *j* occurred at the instant *t* given we observed the sequence of actions features Y^* and $D_t = P(X_{t,i}=j\ /\ Y^*,\ \theta_i)$ is the probability to be in the state *j* at time *t* given we observed the sequence of actions features Y^* . We compute these quantities as follows:

$$C_t = \frac{fwd_{t,i,k} \cdot A_{i,j,k} \cdot B_{i,y_{t+1},k} \cdot back_{t+1,i,j}}{\sum_t fwd_{t',i,j}} \tag{17}$$

$$D_t = \frac{fwd_{t,i,j} \cdot back_{t,i,j}}{\sum_{t'} fwd_{t,i,j}} \tag{18}$$

At this step, we use Baum-Welch to update the model following the equations (7) and (8).

At this point, the idea is to improve the correspondence between the action state $x_{t,i}$ and action features state y_t. Until here we addressed the problem of recognition of an activity as a time series with different length. To address the tracking problem, we derive a statistics from the preselected activities. To this end, we define the following parameters:

$$start(x_i) = \frac{NbS(x_i)}{n_i} \tag{19}$$

$$last(x_i) = \frac{NbL(x_i)}{n_i} \tag{20}$$

$$trans(n,m) = \frac{NbT(n,m)}{\sum_n Nb(n,m)} \tag{21}$$

$NBX(x_i)$ is the normalized number of activities on the class i starting with the action x_i, inversely $NBL(x_i)$ is the normalized number of activities on the class i lasting with the action x_i. NBT (n,m) is the normalized number of transition from the activity n to the activity m. The idea is to track the human activity by using the previous transition matrix and making some modification in order to take into account the sequence order with which the actions are taking place on one activity. The new state transition probability will be modified assuming the following values:

1. $A_{j,k}$ = Trans(n,m), if j is the last action on the activity n first to switch to another state, and k is the first action on the activity m.
2. $A_{j,k} = 0$, if j and k are executing on different activities.
3. $A_{j,k}=0$, if j and k are on the same activity but j comes out after k.
4. $A_{j,k}= A_{i,j,k}$, if j and k are on the same activity but j comes out before k.

The state transition probability $B_{y,j}$ keeps the same values.

4. Experimental results

4.1 Recognition testing

To validate the proposed model, the users were monitored while are performing their tasks, all located in the kitchen. Accordingly to the data training, we collected a number of runs of eight pre-selected activity classes performed in the kitchen. The purpose of the system is to recognize the activity from the measurement of the actions performed while an unknown activity is executing. The system recognition is based as seen above, on calculating the maximum path probability for actions over all possible paths resulting from experimental observation. This means that we need to compute the following equation:

$$P(X_{1:T}/Y_{1:T}) \infty P(Y_{1:T}/ X_{1:T})P(X_{1:T}) \tag{22}$$

which is similar to maximize the model of likelihoods for all possible models $P(Y^*/\square_i)$ built for all selected activities whose likelihood is the highest [11]:

$$i^* = \arg\max_{1 \le i \le N_i} [P(Y^* / \theta_i)] \tag{23}$$

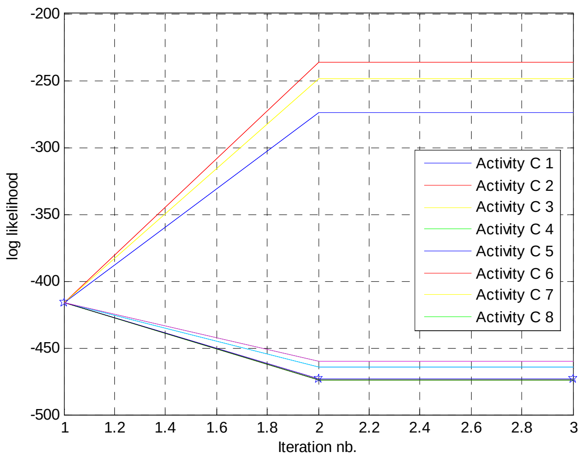

Figure 4 Log likelihood of learned activities class

Fig. 4 shows the logarithmic likelihood of eight activity class where it is clear that the activity class 2 will be selected satisfying the condition of Eq. (16).

4.2 Experimental verification

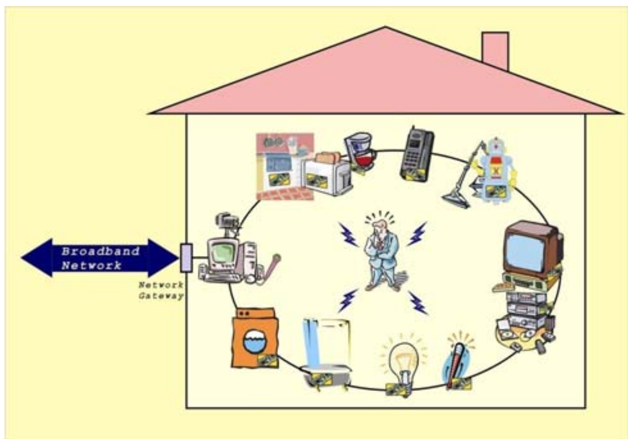

Figure 5 Networked smart environment

The idea behind the algorithms described above is to provide a reasoned guidance on what will belong to the probabilistic intelligent expert we attempt to implement. On the real world environment, the appropriate architecture as shown in Fig. 5 requires the support of ubiquitous networked computing. To this end, a distributed wireless sensor network seems to be an optimal candidate to well satisfy this necessity, supporting multiple sensory devices with sufficient networking capabilities. Meanwhile, an in-house simulator as shown in Fig. 6 was developed in order to display the sensor status events and to perform the further control on the devices using a GUI. We are also currently implementing the API for the hardware interfaces for a variety of sensor units seen in Fig. 7.

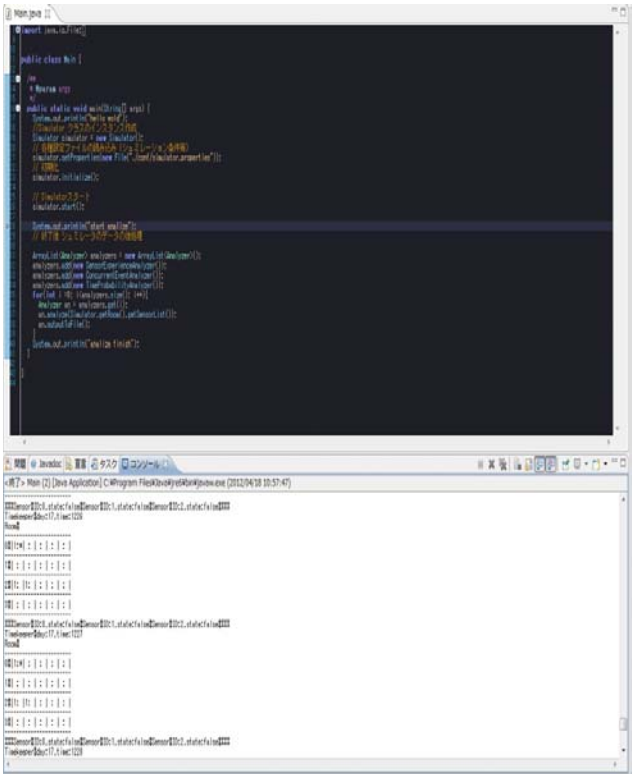

Figure 6 Simulator Screenshot

Figure 7 Arduino and PIR sensor

5. Concluding Remarks

In this paper, an efficient solution to the problem of tracking human activities of daily living is presented. This work is a first step toward implementing a probabilistic expert system capable of tracking human activities and providing useful services within a smart home environment. The acquisition and analysis of an enormous amount of data requires reliable yet cheaper devices, as well as efficient algorithms, which constitute a critical aspect to be used widespread. The current feasibility study showed encouraging preliminary results. The applicability and effectiveness of the proposed algorithms will be further verified through extensive simulations and real world experiments.

References

- D. Cook, M. Youngblood, E. Heierman, K. Gopalratnam, S. Rao, A. Litvin, F. Khawaja, "Mavhome: an agent-based smart home," *IEEE Int. Conf. on Pervasive Computing and Communications*, 2003.
- S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, "The gator tech smart house: A programmable pervasive space," *Computer*, vol. 38, no. 3, pp. 50–60, 2005.
- G.Abowd and E.Mynatt, *Smart Environments: Technology, Protocols, and Applications*, Wiley, pp. 153–174, 2004.
- H. K. Dieter, D. Fox, O. Etzioni, G. Borriello, L. Arnstein,"An overview of the assisted cognition project," *AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder*, 2002.
- P. Rashidi, D.J. Cook, L.B. Holder, M. Schmitter-Edgecombe, "Discovering activities to recognize and track in Smart Environment," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 23, Issue 4, 2011.
- A. Madabhushi, J.K. Aggarwal; "A Bayesian approach to activity recognition," IEEE Workshop on VS, 1999.
- L. Chen, C. Nugent, H. Wang, "Knowledge driven approach to activity recognition in smart homes," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 21, 2009.
- U. Maurer, A. Smailagic, D. Siewiorek, M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," *Int. Workshop on Wearable and Implantable Body Sensor Networks*, pp.4–116, 2006
- T.V. Duong, H.H. Bui, D.Q. Phung, S. Venkatesh, "Activity recognition and abnormality detection with switching hidden semi Makov model," *CVPR 2005.*
- T.V. Duong, H.H. Bui, D.Q. Phung, S. Venkatesh, "Human behavior with generic exponential duration modeling in the hidden semi-markov model," *ICPR 2006.*
- L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, *Proc. IEEE*, 77:257-286, 1989.