Artificial Intelligence Applied to Assistive Technologies and Smart Environments: Papers from the AAAI-14 Workshop

Activity Prediction Based on Time Series Forcasting

Mohamed Tarik Moutacalli, Kevin Bouchard,
Abdenour Bouzouane and Bruno Bouchard
Department of Computer Science
UQAC
Chicoutimi, Canada
mohamed-tarik.moutacalli@uqac.ca

Abstract

Activity recognition is a crucial step in automatic assistance for elderly and disabled people, such as Alzheimer's patients. The large number of activities of daily living (ADLs) that these persons are used to performing as well as their inability, sometimes, to start an activity make the recognition process difficult, if not impossible. To adress such problems, we propose a time-based activity prediction approch as a preliminary step to activity recognition. Not only it will facilitate the recognition, but it will also rank activities according to their occurrence probabilities at every time interval. In this paper, after detecting activities models, we implement and validate an activity prediction process using a time series framework.

Introduction

Automatic assistance (Layzell et al. 2009) is the most promising solution to reduce sky rocketing health care costs associated with traditional health care. In this type of assistance, the caregiver will be assisted in his task by an ambient agent capable of observing, inferring the activity being carried out and assisting, if needed, to ensure the smooth running of the activity started. The general idea behind this assistance is to transfer an elderly patient or one with a cognitive dysfunction, such as Alzheimer's disease into smart home (Stefanov et al. 2004) equipped with sensors and effectors. Different types of sensors may be installed to alert the ambient agent of any change in the home, including infrared sensors, pressure mats, electromagnetic contacts, temperature sensors, light sensors and RFID tags that can be embedded in every day objects. In contrast, effectors such as IP speakers, lights, LEDs and televisions are used by the ambient agent, if necessary, to guide the smart home occupant to take the next action of the activity he has started.

Between observing changes in the habitat, detection of activated sensors, sensors that have changed status or that have experienced a large change in measured value, and assisting the occupant, the ambient agent has to infer the current activity being performed. This activity recognition (Hoey et al. 2012) is the most challenging stage in the process of automatic assistance. It is done by finding, from all activities models, the one that best explains the ordered activated sensors list. An activity model reflects the way in which the occupant is used to performing this activity. Since these activities models must allow the ambient agent to know the next sensor to be activated and to detect if the occupant is struggling to continue the activity, an activity model consists of an ordered sequence of sensors that contains the estimated delay between two adjacent sensors. For example, this part of an activity model can be read:

CA9(5-15), MLK(1-4), CA9(1-2)

The fridge door, represented by the tag-object CA9, will be activated in five to fifteen seconds. After that, milk, represented by the tag-object MLK, will be activated in one to four seconds; and one to two seconds after the milk activation, the fridge door will be activated again.

Several problems may be encountered during the process of activity recognition. The high number of daily living activities that a person is able to perform as well as the high number of activities models, can make this process complex and slow. Having *m* activities models, if we observe *n* actions, the execution time of finding the activity or activities that contains the observed actions is in *O(nm)*. Therefore, considering only the most likely activities will reduce the execution time. The second issue is an equiprobability problem. It occurs when several activities models contain the observed actions. In this case, all activities have the same probability of being carried out, and if the occupant is no longer able to complete its activity, the ambient agent would not know which one to propose. It is notable that the inability to initiate an activity prevents its recognition since no sensor will be activated. To address all cited problems, the step of activities start time prediction must precede the stage of activity research. The following scenario highlights the role of activities start time prediction:

Suppose that we have sixty activities models and the activities start times predicted are: 8h:30 am for *Activity1*, 8h:35 am for *Activity2*, 8h:45 am for *Activity3* and the other activities are predicted after 9h:00 am. At 8h:32 am, for example, we can start looking, at first, for the activity among those predicted at a time close enough to the current time, i.e. to search among the first three activities instead of sixty. Furthermore, if activated sensors belong to the first three activities, or if no sensor is activated, then *Activity1* will most likely be performed since it is the closest to the current time.

As shown in the example, the prediction accuracy plays a key role in identifying the most likely activity– the one that will be proposed to the home occupant if no action is detected. Two minutes later, if *Activity2* predicted start time was 8:33, the most likely activity would be *Activity2* instead of *Activity1*. Getting the most accurate prediction explains our use of time series forecasting techniques.

In this paper, from a sensor history log, we detect activities models and their start times. Then, we explore time series (Box et al. 1994) techniques to predict activities' start times in new horizons that will be used to predict the most likely activity to be performed. The paper is organized as follows. We first discuss the existing approaches towards activity prediction. Then, our frequent pattern mining algorithm for creating activities models is summarised. Next, we detail the time series forecasting technique used in activities start times prediction and our activity prediction system before presenting experimental results of the approach employing real sensor database.

Related work

Over the last decade, there has been a growing interest in activity recognition in smart homes. Several approaches have been proposed based on different techniques, such as hidden Markov models (Duong et al. 2005), Bayesian networks (Inomata et al. 2009) or data mining techniques (Galushka et al. 2006). Spatiotemporal constraints have been used to improve these approaches and answer different problems. On the other hand, few studies have been conducted on activity prediction despite its ability to respond to various problems related to activity recognition.

Cook et al. (Cook & Nazerfard 2013) designed the CRAFFT algorithm based on a Bayesian network, shown in figure 1, for activity prediction.

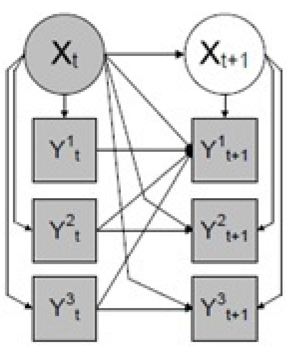

Figure 1: CRAFFT Bayesian network.

X_t is the current activity, X_{t+1} is the activity that is expected to occur immediately afterwards, Y_t^1 is the location in the smart home where X_t occurs, Y_t^2 is a discretized value of the time when X_t occurs and Y_t^3 is an integer value representing the day of the week.

CRAFFT consists of two prediction steps. In the first step, the features representing the next activity (Y_{t+1}^1, Y_{t+1}^2 and Y_{t+1}^3) are predicted by finding Y_{t+1}^* which satisfies:

$$y_{t+1}^* \leftarrow argmax_{y_{t+1}} P(Y_{t+1} = y_{t+1} | Y_t = y_t, X_t = x_t)$$

In the second step, the activity is predicted based on the predicted features in the first step by finding X_{t+1}^* which satisfies:

$$x_{t+1}^* \leftarrow argmax_{x_{t+1}} P(X_{t+1} = x_{t+1} | y_t^1, y_t^2, y_t^3, X_t = x_t)$$

CRAFFT was tested with eleven activities and the prediction accuracy was more than 70%,however, the first question that may arise (and no response were specified in their article) concerns the execution time, especially with a high number of ADLs. We can also ask about the choice of Y_t^2 intervals and note that if the occurrence time of X_t changes by one minute the Y_t^2 value may also change which can modify the activity predicted.

In the same work, they also described a method to estimate the predicted activity start time. They extract the time offset between each two consecutive activities in the dataset and cluster them using the Expectation Maximization algorithm in order to construct a normal mixture model for the time offsets. Beside the computational complexity in terms of execution time for finding clusters and choosing the mixture distributions, the use of the segmentation cannot ensure sufficiently short intervals for accurate prediction. The only predicted start time was for taking medication, and it was between 15 and 45 minutes.

Activity prediction based on the relationship with other activities has been studied more deeply in the work of Jakkula (Jakkula et al. 2007). After initially creating temporal intervals to represent events with their start and end times from the activated sensors history log, relations between frequent pair intervals are discovered. From the thirteen temporal Allen's relation types, they create only nine by comparing temporal range intervals; Y after X when start(X)<start(Y) and end(X)<start(Y), etc. The third step consists of keeping only the best rules that have a high frequency. After this step, activities can be predicted and anomalies can be detected. If an event Y happens, the probability that X occurs is calculated as follows:

P(X|Y)= |After(Y,X)| + |During(Y,X)| + |OverlappedBy(Y,X)| + |MetBy(Y,X)| + |Starts(Y,X)| + |StartedBy(Y,X)| + |Finishes(Y,X)| + |FinichedByBy(Y,X)| + |Equals(Y,X)| / |Y|

If P(X|Y) is near 1 then X is likely to happen. Otherwise, If P(X|Y) is near 0 and X occurred after Y then an error has occurred.

The approach of Jakkula et al. is very worthwhile, but it is more efficient for anomalies detection than activities detection. Indeed, for detecting an anomaly, one probability is calculated between the last two activities performed; whereas for activity prediction, probabilities between the last activity performed and all the others are calculated; thus the activity with the highest probability is chosen. The fact that this approach is not based on any constraints causes a second problem: the detection of an activity, Y for example, will always leads to the same prediction X because P(X|Y) will always be maximal.

Proposed Activity prediction approach

As mentioned in the introduction, activities prediction will allow us reduce the number of likely activities during the research time, solving the equiprobability problem and having the most likely activity, even if no action is detected. Our approach is conducted in three main stages. First we apply our new activities mining algorithm (Moutacalli et al. 2014) to sensors history log in order to detect activities models and extract important information that will be used in the next stages. After that, we use time series forecasting techniques to predict activities' start times. The last stage consists of using predicted activities' start times in order to find the most likely activity to be performed.

Activities models creation

Every person performs his/her activities in his/her own way. Hence, we have to create the activities models in a personalized and unsupervised way from the sensors history log. The large number of sensors that continuously send their measurements, makes a huge and almost unusable database. To reduce its size, without losing relevant information, the database structure must be modified. Instead of storing all the information sent by the sensors, as shown in Table 1, it will only retain activated sensors with their time of activation, Table 2.

Date	Time	Sensor 1	...	Sensor N
01/04/2014	08:50:20.000	ON	...	1245
01/04/2014	08:50:20.500	OFF	...	1245

Table 1: Sensors values on each 500 millisecond

Date	Activated sensor
01/04/2014	Sensor 1(Time1), Sensor 7(Time2), ...
02/04/2014	Sensor 4(Time1), ...

Table 2: Activated sensors sequences

From Table 2, activities models creation is done by finding frequent closed patterns. Several algorithms exist for frequent patterns detection. Most of them are based on the Apriori algorithm (Agrawal & Srikant 1994), which combines short frequent patterns in order to find the longest ones, unlike our proposed algorithm (Moutacalli et al. 2014), which divides between infrequent adjacent sensors. It starts by creating a table of frequent adjacent sensors couple with their time differences. For instance, if the sensors history log is composed of the two days sequences D1 and D2, and the frequency threshold is equal to 2, the result will be as shown in Table 3.

D1 = $<$ (A, 01), (B, 04), (C, 06), (D, 28), (A, 36), (A, 41), (B, 56), (E, 58), (A, 59) $>$

D2 = $<$ (E, 08), (A, 11), (B, 30), (E, 32), (A, 42), (B, 46), (C, 49)$>$

Time difference between two adjacent sensors is very important not only because it allows the ambient agent to know when the second sensor ought to be activated, but it also allows us to differentiate between activities. For example, boiling water with a time difference equal to two may belong to making tea, but with a time difference equal to fifteen, it may belong to making pasta. That's why we use C-Means (Bezdek & Ehrlich 1984), a fuzzy clustering algorithm, on Table 3 to divide sensors couples depending on time differences. The optimal number of clusters is determined by minimizing the average deviation C_N of each value from the median MD_j.The resultant table is shown in Table 4.

Sensor 1	Sensor 2	Frequency	Time differences
A	B	4	3, 15, 19, 4
B	C	2	2, 3
B	E	2	2, 2
E	A	3	1, 2, 10

Table 3: Time differences of frequent adjacent events

$$C_N = \sum_{i=1}^{N} [\frac{\sum_{j=1}^{ValueNum} |x_{ij} - MD_j|}{ValueNum}]$$

Sensor 1	Sensor 2	Frequency	homogenous interval
A	B	2	[3 - 4]
A	B	2	[15 - 19]
B	C	2	[1 - 3]
B	E	2	[2 - 2]
E	A	2	[1 - 2]

Table 4: Frequent homogenous interval of adjacent events

After the creation of table 4, each couple that is not listed in Table 4 or its time difference does not belong to one of its intervals is divided. Figure 2 shows the first cuts made in our example. Once the cuts are made, sequences composed of one couple are removed. In the remaining sequences, each couple is replaced by its index in Table 4. The three steps, frequent couples finding, infrequent couples cutting and couples substitution are then repeated until no more sequences remain. A, B(3-4), C(1-3) is one of activities models created by this algorithm.

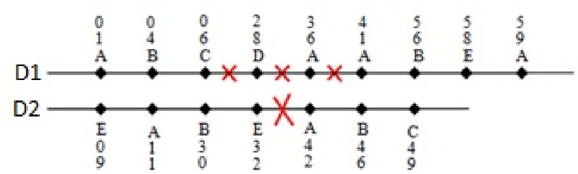

Figure 2: First cuts.

It should be noted that a last pass over the sensors history log is used to collect important information to be used in the next steps. For each activity model, we extract the daily start time, the daily start time range and the daily occurrence confidence as defined by its daily frequency. If an activity is usually performed *n* times a day, *n* different activities will be considered with different start time ranges. The value of *n* is calculated by dividing the activities models frequency by the minimal frequency, while start time ranges are found by segmentation using the k-means algorithm (Xiong et al. 2009) with $k = n$ on activity start times.

Activities start times prediction

To implement an accurate prediction system, we decided to use time series which are used in various fields, especially in econometrics, and which give promising results. A time series is a sequence of observations taken sequentially in time (Box et al. 1994) denoted $(X_t)_{t\in\theta}$ where the set θ is the space time. Time series data occur naturally in many application areas; monthly data for unemployment in economics, daily exchange rate in nance, daily rainfall in meteorology, etc.

Time series techniques are based on the notion of autocorrelation to predict future values. This means that successive series values must depend on each other; otherwise, the series is random and its future values are not predictable. This can answer the question that we might ask about the possibility of predicting the time when a person will carry out an activity. It is true that a person has the ability to perform any activity at any time, but we all have a lifestyle and habits such that the majority of our activities are carried out at almost the same time daily or weekly, making allowance for weekend variation. In addition, the prediction will be more accurate for the activities that we are more anxious to predict such as taking medication, where start time is rigorously monitored.

To predict series future values, many established models exist (Box et al. 1994). In this paper we opted for the Autoregressive integrated moving average (ARIMA) model which forecast X_t future values based on a linear regression of the series previous values and a linear combination of past forecast errors (i.e. noise).

$$X_t = \sum_{i=1}^{p}\Phi_i X_{t-i} + \sum_{i=1}^{q}\theta_i X_{t-i} + \epsilon_t + c$$

Where p is the autoregressive order, q is the moving average order, $\Phi_1, ..., \Phi_p$ and $\theta_1, ..., \theta_1$ are the model parameters, c is a constant and ϵ_t is white noise.

In ARIMA model, if a series is not stationary; its statistical properties, mean, covariance and autocorrelation, are not constant over time, a technique called differencing with order d is applied to make it stationary. The new series obtained by a differentiation with $d = 1$: $D_t^1 = X_t - X_{t-1}$ and with $d = 2 : D_t^2 = D_t^1 - D_{t-1}^1$.

Predicting X_t future values with ARIMA(p, d, q) is done by finding the three parameters p, d and q that yield the best results. The first parameter, d, is equal to 0 if X_t is stationary; otherwise, d is incremented until D_t^d is stationary. To decide if a time series, X_t, is stationary, KPSS tests (Kwiatkowski et al. 1992) is used. In the KPSS model, time series is represented as a sum of three components:

$$X_t = \xi t + r_t + \epsilon_t \text{ and } r_t = r_{t-1} + u_t$$

Where t is a deterministic trend, r_t is a random walk process, ϵ_t is a stationary error terms and u_t is an error term with a constant variation σ_u^2.

Then, the following three rules are used to decide if X_t is stationary:

- If $\xi = 0$ then X_t is stationary around r_0;
- If $\xi \neq 0$ then X_t is stationary around a linear trend;
- If $\sigma_u^2 > 0$ then X_t is non-stationary.

The second step of ARIMA model is to find the parameters p and q that better fit the model. For this purpose, the chosen p and q parameters are the ones that minimize the Akaike's information corrected criterion (AICC) (Khim & Shitan 2002). AICC statistics is given by:
$AICC = -2lnLikelihood(\widehat{\Phi}, \widehat{\theta}, \widehat{\sigma}^2) + [2n(p+q+1)]/[n-(p+q)-2]$
where $\widehat{\Phi}$ is a class of autoregressive parameters, $\widehat{\theta}$ is a class of moving average parameters, $\widehat{\sigma}^2$ is the variance of white noise, n is the observations number, p is the autoregressive component order, q is the moving average component order and $Likelihood(\widehat{\Phi}, \widehat{\theta}, \widehat{\sigma}^2)$ is the likelihood of the data under the Gaussian ARMA model with parameters $(\widehat{\Phi}, \widehat{\theta}, \widehat{\sigma}^2)$.

Predicting activities start times is performed using two different stages. At night, while the smart home occupant is asleep, start times of all activities are predicted for the next day. Therefore, each activity is represented by a time series $(X_t)_{t\in\theta}$ where θ is the observation days set and X_i is the activity start time at the day i. Because a large variation between an activity's predicted start time and its actual start time can influence the start time of each activity that occurs close enough to the detected one, a second prediction is performed. Now, a time series $(Y_t)_{t\in\theta}$ is created for each activity that takes place around the last activity detected X_t, where θ is always the observation days set and Y_i is start times difference between the two activities at the day i, $X_i' - X_i$. Thus, the second prediction stage is performed only if there is a large variation between an activity start time detected, XTd_h, and the one predicted for the same activity at the horizon h. It allows for updating predicted start times for the closest activities to X_t : $PX'ST_h = XTd_h + Y_h$
Where Y_h is the prediction at horizon h.

Activity prediction

In this step, we look for the most likely activity to be performed using the results of the two previous steps. To facilitate activity recognition system integration to the prediction system, we chose to use a Bayesian network to calculate the probabilities. Algorithm 1 details how this step has been programmed.

Algorithm 1 starts by creating the closest activities set based on predicted start time and current time and assigns an initial probability to each activity in the set, based on daily confidence. Then, probabilities are updated to make the closest ones to current time more probable. When a large variation between predicted start time and actual start time of the activity is observed, predicted activities start times are recalculated. Probabilities are updated again so that activities far from their occurrence range will be considered less probable. As mentioned, activity recognition system can be programmed using the same Bayesian network by updating probabilities according to the belonging of detected sensors to an activity.

Algorithm 1 Calculate probabilities

1: **for each** model a found in step1 **do**
2:     PSTa = predicted start time
3:     **if** (|PSTa Current time Ct| $< \epsilon$) **then**
4:         Add a to SA
5:         Probability Pa = (daily confidence)$_{normalized}$
6:     **end if**
7: **end for**
8: **for each** a in SA **do**
9:     Pa = Pa * $(\frac{1}{|PSTa-Ct|})_{normalized}$
10: **end for**
11: **if** (|PSTa $-$ ASTa| $> \epsilon$) **then**
12:     **for each** a in SA **do**
13:         PSTa = New predicted start time
14:     **end for**
15: **end if**
16: **for each** a in SA **do**
17:     distance from daily range(Ra_{min}, Ra_{max}) DR = 1
18:     **if** (PSTa $< Ra_{min}$) **then**
19:         DR = $Ra_{min}-$ PSTa
20:     **end if**
21:     **if** (PSTa $> Ra_{max}$) **then**
22:         DR = PSTa $-Ra_{max}$
23:     **end if**
24:     Pa = Pa * $(\frac{1}{DR})_{normalized}$
25: **end for**
26: Sleep(S seconds)
27: **Go to** 8

Validation

In order to validate our new approach, we tested it by simulating a person's waking up routine for 28 days in the "Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités" LIARA laboratory. The LIARA laboratory possesses a new cutting edge smart home infrastructure that is about 100 square meters and about one hundred different sensors and effectors. Among the sensors, there are infrared sensors, pressure mats, electromagnetic contacts, various temperature sensors, light sensors, eight RFID antennas and RFID tags implemented on everyday objects. Figure 3 shows a cluster of images from different parts and angles of our smart home. The obtained sensors history log is composed of approximately 1100 events. In the activities models detection six activities were detected: *Wake up*, *Use toilet*, *Wash hands*, *Take shower*, *Prepare coffee* and *Leave house* (for more detail please refer to (Moutacalli et al. 2014)).

The second step first stage consisted of predicting all activities' start times using the best ARIMA forecasting model. Time series contained only three weeks of start times, while the fourth week data were used for validating the forecasting results. 83% of detected activities were well predicted. Only *Leave house* seems to be random. The accuracy of our model can be seen in Figure 4, which shows the four weeks plot of *Use toilet* time series and Figure 5, which shows the plot of three weeks and the next predicted week.

The chosen model for this time series was ARIMA(1,0,0) which means that the series is stationary, d =0, and the minimal value of AICC is 358.67 and was found by p=1 and q=0. As shown in the two lasts figures, predicted values are close enough to actual ones except for the 24th day; predicted value PSTa is 3101.941 while the actual value is 4559. The same generalization can be made regarding *Wake up* prediction, which precedes *Use toilet*. Thus, step 2 stage 2 creates a new time series containing start time differences between

Figure 3: LIARA laboratory.

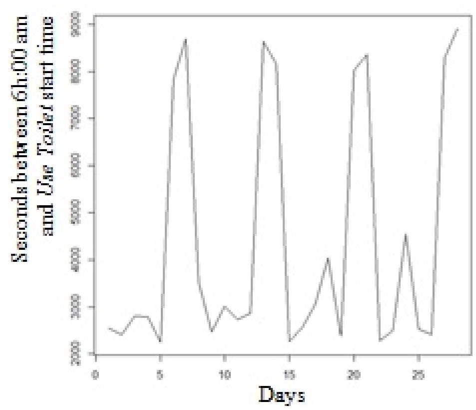

Figure 4: *Use Toilet* four weeks plot.

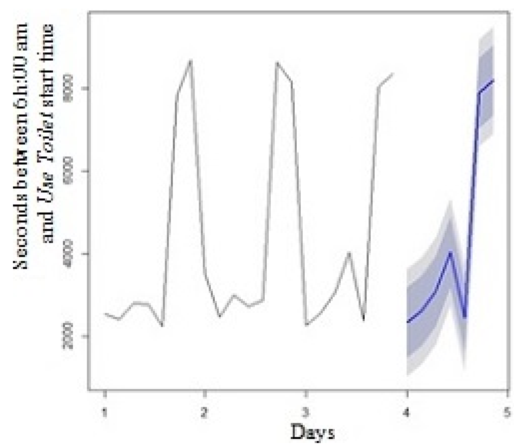

Figure 5: *Use Toilet* predicted week.

the two activities. The predicted week of this activity is presented in Figure 6.

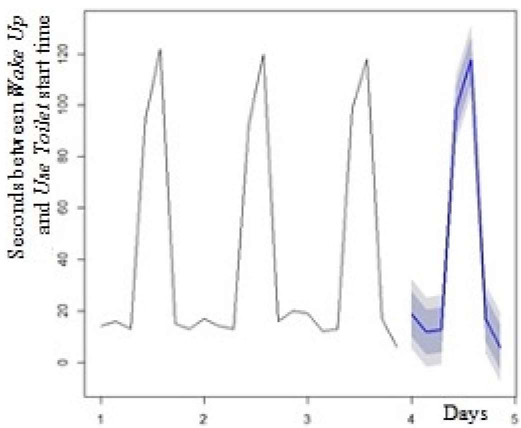

Figure 6: Predicted times between *Wake Up* and *Use toilet*.

Knowing *Wake up* actual start time for the 24th day, 4546, and with the new activity predicted value for the same day, 12.714, the *Use toilet* predicted time is updated:

PSTa = 4546 + 12.714 = 4558.71 $\approx$ 4559

A general evaluation of our approach is made by calculating the percentage of performed activities that have the maximum likelihood calculated by Algorithm 1 during the fourth week. Figure 7 shows the obtained results. From Figure 7, we can see that *Wake up* and *Wash hands* were perfectly performed as predicted. The *Wake up* result was expected because it is always the first performed activity. The same observation can be made to *Wash hands* which always follows *Use toilet*. The result of *Take shower* was not optimal because this activity was not carried out regularly. The worst result was for *Leave house* which we could not forecast. What can be deduced from Figure 7 is that the best results occurred for activities performed less randomly. Since most activities are not random, the proposed method recommends itself.

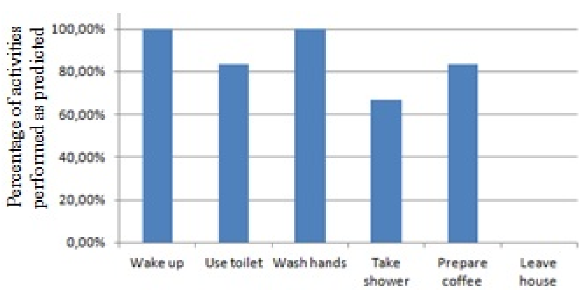

Figure 7: Percentage of well predicted activities.

Conclusion

In this paper, we proposed a new approach that predicts the next activity to be performed. This prediction can be very useful for reducing the number of activities to be considered during activity recognition, for eliminating the equiprobability problem or for recognizing an activity even if no action is detected. It takes advantage of time series forecasting to predict an activity start time from its previous values or from the start time differences with the last detected activity. The results were satisfactory for most activities, especially the ones that are performed less randomly. The tests were conducted on simulated data. More tests will be conducted with real data recorded in our laboratory LIARA. In our next research, we will ameliorate this approach and use it as a preliminary step for an activity recognition system.

References

Layzell, B.; Manning, B.; Benton, S. 2009. The elderly demographic time bombSharing the load with the active ageing: Can eHealth technologies help defuse it? Stud. Health Technol. Inform. 146, 166170.

Stefanov, D.H., Bien, Z. and Bang, W.C. 2004. The smart house for older persons and persons with physical disabilities: Structure, technology arrangements, and perspectives. IEEE Trans. Neural Syst. Rehabil. Eng. 12, 228250.

Hoey, J, Nugent, C.D. and Cook, D.J. ; Zhiwen Yu. 2012. Sensor-Based Activity Recognition. IEEE Systems, Man, and Cybernetics Society, 790 808.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. 1994. Time Series Analysis, Forecasting and Control. Englewood Clifs, NJ: Prentice Hall, 3rd ed.

Duong, T., Bui, H. H., Phung, D. Q., and Venkatesh, S. 2005. Activity recognition and abnormality detection with the switching hidden semi-markov model. IEEE Computer Vision and Pattern Recognition, 838 845.

Inomata, T., Naya, F., Kuwahara, N., Hattori, F., and Kogure, K. 2009. Activity Recognition from Interactions with Objects using Dynamic Bayesian Network. Inomata.

Galushka, M., Patterson, D., and Rooney, N. 2006. Temporal Data Mining for Smart Homes. Springer, Berlin, pp 85-108.

Cook ,D.J. and Nazerfard, E. 20013. Using Bayesian Networks for Daily Activity Prediction. AAAI Workshops on Artificial Intelligence.

Jakkula, V. R. and Cook, D. J. 2007. Mining Sensor Data in Smart Environment for Temporal Activity Prediction. Poster session at the ACM SIGKDD, San Jose, CA.

Moutacalli, M.T., Bouzouane, A. and Bouchard, B. 2014. New frequent pattern mining algorithm for activities models creation. ACOMORE.

Bezdek, J, C., and Ehrlich, R, W. 1984. The Fuzzy C-Means Clustering algorithm., Computers & Geosciences, Volume 10, Issues 2 3, Pages 191203.

Xiong, H., Wu, J., and Chen, J. 2009. K-means Clustering versus Validation Measures: A DataDistribution Perspective. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEETransactions.

Kwiatkowski, D., Phillips, P.C.B., Schmidt, P. and Shin, Y. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root. Journal of Econometrics 54, 159-178.

Khim, L. and Shitan, M. 2002. The Performance of AICC as an Order Selection Criterion in ARMA Time Series Models. Pertanika J. Sci. & Technol, 25-33.

Agrawal, R. and Srikant, R.S. 1994. Fast Algorithms for Mining Association Rules. IBM Research Report RJ9839, IBM Almaden Reseach center, San Jose, Calif.