

Using Bayesian Networks for Daily Activity Prediction

Ehsan Nazerfard

School of Electrical Eng. and Computer Science
Washington State University
Pullman, WA USA
nazerfard@eecs.wsu.edu

Diane J. Cook

School of Electrical Eng. and Computer Science
Washington State University
Pullman, WA USA
cook@eecs.wsu.edu

Abstract

In spite of the significant work that has been done to discover and recognize activities in the smart home research, less attention has been paid to predict the future activities that the resident is likely to perform. An activity prediction module can play a major role in design of a smart home. For instance, by taking advantage of an activity prediction module, a smart home can learn context-aware rules to prompt individuals to initiate important activities. In this paper, we propose an activity prediction approach using Bayesian networks. We propose a novel two-step inference process to predict the next activity features and then to predict the next activity label. We also propose an approach to predict the start time of the next activity which is based on modeling the relative start time of the predicted activity using the continuous normal distribution and outlier detection. We evaluate our proposed models using real data collected from two smart home apartments.

Introduction

Over the last decade, there has been a growing interest in the development of smart environments which are able to reason about their residents. A smart environment is defined as one that is able to acquire and apply knowledge about an environment, such as a home, and adapt to its inhabitants to improve their experience in that environment (Cook and Das 2005). The main goal of such technologies is to achieve greater comfort, productivity, and energy efficiency. Researchers have come to realize the importance of applying smart home technologies for health monitoring (Pollack 2005; Helal et al. 2005; Rashidi and Cook 2009; Crandall 2011) and companies are recognizing the potential of such technologies in the market (BrainAid.com 2013).

In many smart home projects, the final goal is to automate residents' interactions with the environment, particularly interactions that are repetitive or difficult to perform for older adults or people with disabilities. An example of an assistive living system is a remote health monitoring system that monitors and tracks the activities of daily living (ADLs) for older adults with memory impairment. ADLs consist of self-care activities such as eating, sleeping, dressing, cooking

and taking medication. The ability to perform ADLs independently and completely on a regular basis provides measurement of the functional status of residents if they want to live independently in their own homes. Moreover, such systems can provide timely prompts to the residents in case they have forgotten to perform critical activities.

When older adults with cognitive impairment fail to initiate or complete everyday ADLs, caregivers are often responsible to monitor ADLs and provide the reminder or prompt. The prompt is defined as any form of verbal or non-verbal intervention delivered to an individual on the basis of time or context to help the successful completion of an activity. The mentioned interventions are time consuming and cumbersome tasks that too often impact the caregiver's own health in a negative way. Smart home technologies that help people with memory impairment perform their ADLs by detecting when assistance is needed and automatically delivering prompts have the potential to reduce caregiver burden and allow aging adults to retain their functional independence longer.

While reminder systems have been studied heavily, few take into account an individual's behavioral patterns to provide context-aware prompts (Holder and Cook 2013), despite the fact that studies indicate activity-aware prompts offer significant advantages over traditional time-based prompts (Kaushik, Intille, and Larson 2008). The required component which can provide an individual's behavioral pattern to deliver the context-aware prompts is the activity prediction module.

By taking advantage of an activity prediction module, a reminder system can customize its behavior to the residents with no input on their part. Particularly, our technology utilizes data collected from an individual's home to learn context-aware rules for prompting the individual to initiate important daily activities. We assume that sensor data is collected in a home while an individual performs her daily activities. We also assume that the training data is available from when the resident correctly performed the activities or were prompted by a caregiver to initiate the daily activities. Last, we assume that we are given a list of critical activities for which the resident needs to be prompted.

The following scenario highlights the role of an activity prediction component together with the activity recognition module to provide an automated context-aware prompt,

provided that “Taking Medication” is assumed to be a critical activity:

“In the morning, the activity recognition module recognizes that the breakfast activity has occurred. Then the activity prediction component, which has already been trained with the correctly performed activities, makes the following prediction with a high enough confidence: “The taking medication activity will occur within 30 minutes”. Then the activity prompting system would have a relative time offset when the taking medication activity usually happens. When the typical timespan has passed and medication has not been taken, a prompt is delivered.”

In this paper, we propose a prediction approach which uses Bayesian networks in a novel two-step inference process. While we focus on the activity prediction problem in this paper, the proposed two-step inference process is not limited to this area and can be extended easily to the other domains as well. To predict the relative start time of the predicted activity, we propose a method which is based on modeling the relative start time of the activity using a continuous normal distribution and outlier detection. The rest of the paper is organized as follows. We first overview the probabilistic graphical models. Next, we present the details of the proposed activity prediction approach, both for the activity label prediction and the relative start time prediction. We then provide the evaluation results of the proposed method. Finally, we close the paper by providing some concluding remarks and future directions.

Directed Acyclic Graphical Models

In this section, we briefly overview the directed acyclic graphical models, also known as Bayesian networks, and discuss the conditional independence relationships encoded by them.

Bayesian Networks

Probabilistic graphical models (PGMs) are graphs where nodes represent random variables and arcs represent the statistical dependencies between the corresponding random variables. Hence, the PGMs provide a compact way of representing the joint probability distribution. The directed graphical models are also known as Bayesian networks (BNs) or belief networks. We refer to the directed graphical models as Bayesian networks (BNs) throughout the paper. A sample of BNs, representing 5 random variables, is illustrated in Fig. 1.

One of the properties of BNs is their capability of encoding the notion of *causality*. For instance, the arc from A to B in Fig. 1 indicates that node A causes node B . One of the conclusions of the notion of causality is that the directed graph in a BN must be acyclic, as a node cannot cause itself. In addition to the structure, a BN should specify the conditional probability distribution (CPD) for each node. If variables in the model are discrete, the CPD can be represented as a conditional probability table (CPT). A CPT for a node indicates the probability that the node can take each

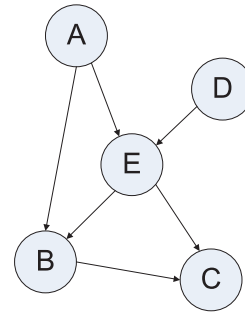


Figure 1: A sample of Bayesian networks.

of its different values for each combination of values of its parents.

The inference in BNs boils down to marginalizing the joint probability distributions (JPD). Given the JPD, we can answer all possible inference queries by marginalizing out the irrelevant variables. Considering a BN consisting of N random variables $Z = (Z_1, Z_2, \dots, Z_N)$, the general form of the joint probability distribution of the Bayesian network can be represented as in Equation 1, where $pa(Z_i)$ represents the parents of node Z_i :

$$P(Z_1, Z_2, \dots, Z_N) = \prod_{i=1}^N P(Z_i | pa(Z_i)) \quad (1)$$

The Equation 1 is derived due to the BNs property that each node is independent of the rest of the network, given its parents. Furthermore, dynamic Bayesian networks (DBNs) extend the BNs by adding a temporal dimension to BNs to model the time series data.

Conditional Independence Relationships

The conditional independence relationships let the joint probability to be presented in a more compact form. The more compact form (factorized form) has fewer parameters, which makes the learning easier. The conditional independence relationship answers to the question that if random variable A is conditionally independent of B , given random variable C . It is common to denote the conditional independence relationship with symbol $\perp\!\!\!\perp$, thus $A \perp\!\!\!\perp B | C$ should be interpreted as random variables A and B are conditionally independent, given variable C .

There are basically three types of relations between random variables A , B , and C , which could be considered as building blocks of the more complex relations. The mentioned conditional independence relationships can be best explained by the *Bayes Ball* algorithm (Shachter 1998) as follows:

Two nodes A and B are conditionally independent, given the third node C , if a ball cannot get to node B starting from node A (or vice versa), when the allowable move is presented in the second column in Fig. 2. In all of the three cases represented in Fig. 2, the shaded node represents variable C , which is assumed to be observable in data. The second column in Fig. 2, representing converging arrows, implies that

the observable node allows the ball to pass through. As a conclusion, the converging arrows case implies that nodes A and B are conditionally dependent, given node C . As opposed to the converging arrows case, the first and third columns represent the $A \perp\!\!\!\perp B|C$ conditional independence relationship.

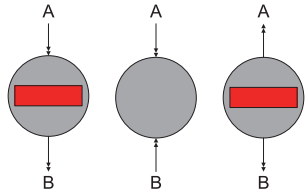


Figure 2: The Bayes ball algorithm: The stop sign indicates that the observable node blocks the ball to pass through.

Proposed Model

In this section, we present our proposed model for the daily activity prediction problem, when the BN structure is unknown and variables contributing in the model have full observability in the data. We assume the horizon of the prediction is one. We also assume variable X_t represents the current activity label and variables Y_t^i 's ($i = 1..n$) represent activity features that are caused by X_t . Taking into account the prompting scenario mentioned in the Introduction, we assume an Activity Recognition (AR) module reports the current activity label X_t , which makes the next activity conditionally independent of all previous activities, i.e. $X_{t+1} \perp\!\!\!\perp X_l|X_t$ where $l < t$. The mentioned conditional independence relationship is illustrated in Fig. 3. In addition to our proposed model, we also present an alternative network-based approaches in this section and compare its prediction accuracies with our proposed model in the Experimental Results section. Finally, we end this section by presenting our proposed approach to predict the start time of the predicted activity.

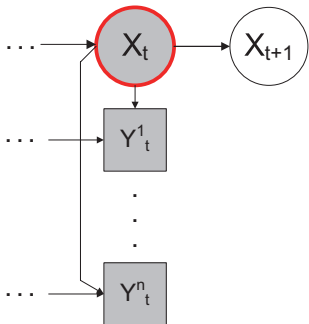


Figure 3: The current activity label X_t is assumed to be given by our AR module.

CRAFFT

In this section, we present our proposed method to solve the prediction problem illustrated in Fig. 3. Taking into consid-

eration the conditional independence relationships in Fig. 3, one should note that activity features Y_t^i 's are conditionally independent of the next activity X_{t+1} (see Fig. 2). Therefore, we propose the Bayesian network presented in Fig. 4 together with our proposed two-step inference process to answer the prediction problem illustrated in Fig. 3.

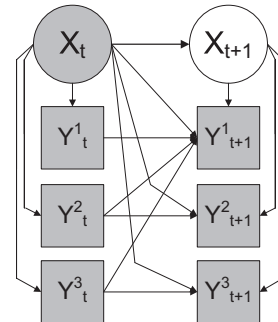


Figure 4: The structure of the proposed model (CRAFFT).

We call our proposed model CRAFFT (short for Using **Cu**rrent Activity and Features to predict the next **Fea**Tures). As it stands from Fig. 4, CRAFFT utilizes three features for each activity. The description of all variables contributing to CRAFFT follows:

- *State variable (X_t):* the label of the current activity. Similarly, the variable X_{t+1} refers to the label of the activity that is expected to occur immediately after X_t .
- *Activity location feature (Y_t^1):* the location in the smart home where the current activity occurs. The location is specified in terms of the location of a sensor. In a smart home, the corresponding sensor will generate a message if resident movement is detected in its field of view. The locations we consider in our smart home include Kitchen, Bathroom, Bedroom, Living Room, Dining Room, Medicine Cabinet, Lounge Chair, Kitchen Door, Bathroom Door, and Front Door.
- *Activity time of day feature (Y_t^2):* a discretized value of the time when the current activity occurs. Time values are binned into the following ranges: 0 – 3, 4 – 7, 8 – 11, 12 – 15, 16 – 19, and 20 – 23.
- *Activity day of week feature (Y_t^3):* an integer value ranging from 1 to 7 representing the day of the week in which the current activity happens, where 1 represents Monday.

Here, we present our proposed method to make the prediction inference in the CRAFFT model illustrated in Fig. 4. The proposed approach consists of two prediction steps. In the first step, we predict the features representing the next activity. In the second step, we predict the activity label based on the predicted features in the first step and the current activity label.

Step I: Next Activity Features Prediction

In the first step, CRAFFT predicts the features that are associated with the next activity, i.e. Y_{t+1}^1 , Y_{t+1}^2 , and Y_{t+1}^3 .

As illustrated in Fig. 5, we hypothesize that features representing the next activity are dependent upon their current values as well as the current activity label. More specifically, CRAFFT hypothesizes that the next activity location is influenced by the current activity label, location, time of day and day of week. Also, the next activity time of day and day of week are dependent upon their current values, as well as the current activity label. For instance, the goal of the activity time of day (Y^2) and day of week prediction is to find y_{t+1}^* which satisfies Equation 2.

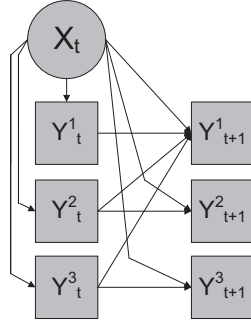


Figure 5: Step I- Next activity features prediction.

$$y_{t+1}^* \leftarrow \operatorname{argmax}_{y_{t+1}} P(Y_{t+1} = y_{t+1} | Y_t = y_t, X_t = x_t) \quad (2)$$

In Table 1, we provide 5 examples for the feature prediction step, where the first row represents the template used to represent examples. Note that the start sign (*) can be replaced by any value of the corresponding feature.

The first example suggests that the Bathing activity, when performed in the morning, is typically followed by an activity that happens in the Bathroom; however, the second example indicates that it is followed by an activity in the Bedroom, when performed in the evening. The first two examples imply why CRAFFT utilizes the current time of day feature to predict the next location feature as well as the current location. Moreover, the third example suggests that the Eating activity at noon is usually followed by an activity in the living room, when performed on Tuesdays through Sundays; however, the fourth example indicates that it is followed by an activity at the Front Door, when performed on Mondays. The last two examples justify the need for considering the current day of week to predict the next activity location as shown in Fig. 5. The last example also suggests that the Sleeping activity, when performed in the evening, is followed by an activity in the Kitchen next day in the morning. Therefore, the last example is to show when there is a transition in day of week and why we connect the current activity to the next day of week feature.

Step II: Next Activity Label Prediction

In the second step, CRAFFT predicts the next activity label based on the predicted features in the first step, as well as

the current activity label. Fig. 6 represents the next activity label prediction step.

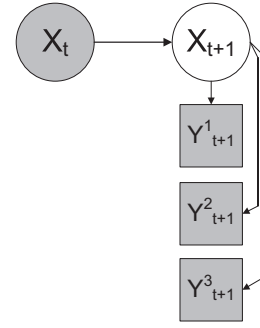


Figure 6: Step II- Next activity label prediction.

The prediction problem we take into account in this section is defined as to find the x_{t+1}^* , such that it satisfies the Equation 3.

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

Related Model

In this section, we review a related structure-based models for the prediction problem represented in Fig. 3. In the Experimental Results section, we compare its prediction accuracy with CRAFFT. Similar to the CRAFFT model, the model presented in this section utilizes the same feature set (i.e. location, time of day and day of week) as well as the current activity label.

Considering the features corresponding to the current activity, one natural model is to predict the next activity based on the current feature set. We refer to the presented model in Fig. 7 as CEFA (short for using CurrEnt Features and activity to predict the next Activity).

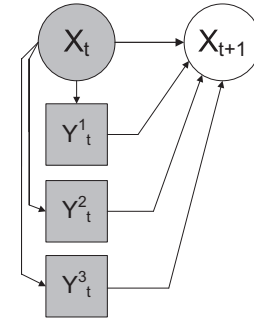


Figure 7: The structure of the CEFA model.

Relative Start Time Prediction for the Predicted Activity

In the previous sections, we discussed how CRAFFT and the other alternative approaches predict the next most likely activity label. In this section, we explore the question that

Table 1: Feature prediction step: examples

| | $(X_t, Y_t^1, Y_t^2, Y_t^3)$ | \rightarrow | $(Y_{t+1}^1, Y_{t+1}^2, Y_{t+1}^3)$ |
|---|------------------------------------|---------------|-------------------------------------|
| 1 | (Bathing, Bath, Morning, *) | \rightarrow | (Bathroom, Morning, *) |
| 2 | (Bathing, Bath, Evening, *) | \rightarrow | (Bedroom, Evening, *) |
| 3 | (Eating, Dining Room, Noon, 2 – 7) | \rightarrow | (Living Room, Noon, 2 – 7) |
| 4 | (Eating, Dining Room, Noon, 1) | \rightarrow | (Front Door, Noon, 1) |
| 5 | (Sleeping, Bedroom, Evening, 7) | \rightarrow | (Kitchen, Morning, 1) |

when the predicted activity starts. As we discussed in the prompting scenario presented in the Introduction, we are interested to know the relative start time of the predicted activity with respect to the start time of the current activity. In order to do that, we extract the time offset between each two consecutive activities in our dataset and cluster them using the Expectation Maximization algorithm to construct a normal mixture model for the time offsets. In Fig. 8, we show three most probable clusters which represent the time offsets between activity X_t and X_{t+1} . It is noteworthy that the time offset in our setting is not exactly the duration of the current activity (X_t), because some “other activities” that we do not consider in our study might occur between X_t and X_{t+1} .

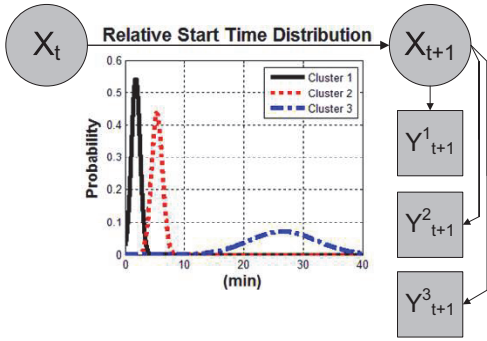


Figure 8: The relative start time distribution for X_{t+1} .

Lets denote the time offset between two activities a_i and a_j by $t_{i,j}$. Then we express the time offset as a normal probability density function with parameters $\Theta_k = (\mu, \sigma)$ as represented in Equation 4. Here μ and σ are the mean and standard deviation values, calculated for the time offset.

$$prob(t_{i,j}|\Theta_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t_{i,j}-\mu)^2}{2\sigma^2}} \quad (4)$$

According to the normal distribution characteristics, the distance of “two standard deviations” from the mean accounts for about 95% of the values, as illustrated in Fig. 9. Therefore, if we consider only observations falling within two standard deviations, observations that are deviating from the mean will be automatically removed. Such observations that are distant from the rest of the data are called “outliers”.

In the Experimental Results section, we present a sample for the time offset modeling between the “Breakfast” and the “Taking Medication” activities.

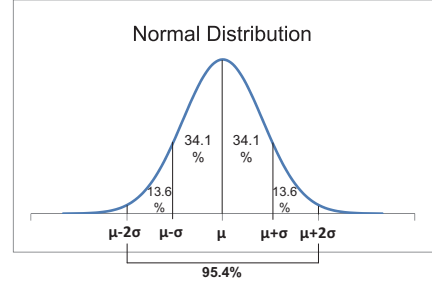


Figure 9: Outlier detection mechanism using the normal distribution properties.

Experimental Results

In this section, we present the experimental results for the CRAFFT model, together with the results of the other related algorithms. Before getting into the evaluation of the CRAFFT model, we describe our Experimental Setup.

Experimental Setup

We use two one-bedroom single resident smart home apartments for our experiments, referred to as Apt1 and Apt2. In order to track resident movements, we use sensors installed on ceilings and walls as well as doors and cabinets. The sensor layouts of Apt1 and Apt2 are shown in Fig. 10, where red circles and blue triangles represent infrared motion sensors and magnetic door/cabinet sensors, respectively. A sensor network captures all the sensor events and our middleware stores them in a SQL database. To provide real activity data for our experiments, we have collected data while the residents were living in smart homes performing normal daily routines. Table 2 provides the characteristics of our smart home testbeds:

Table 2: Characteristics of the smart homes used in our study

| | Apt1 | Apt2 |
|------------------------------|----------|----------|
| # of motion sensors | 20 | 18 |
| # of door/cabinet sensors | 12 | 12 |
| # of residents | 1 | 1 |
| # of sensor events collected | 371, 925 | 254, 918 |
| Timespan | 6 months | 4 months |

For our experiments, we take into consideration 11 ADLs that the residents perform in the smart homes. The activities we consider in our study are as follow: Bathing, Bed-Toilet Transition, Eating, Enter Home, Housekeeping, Leave

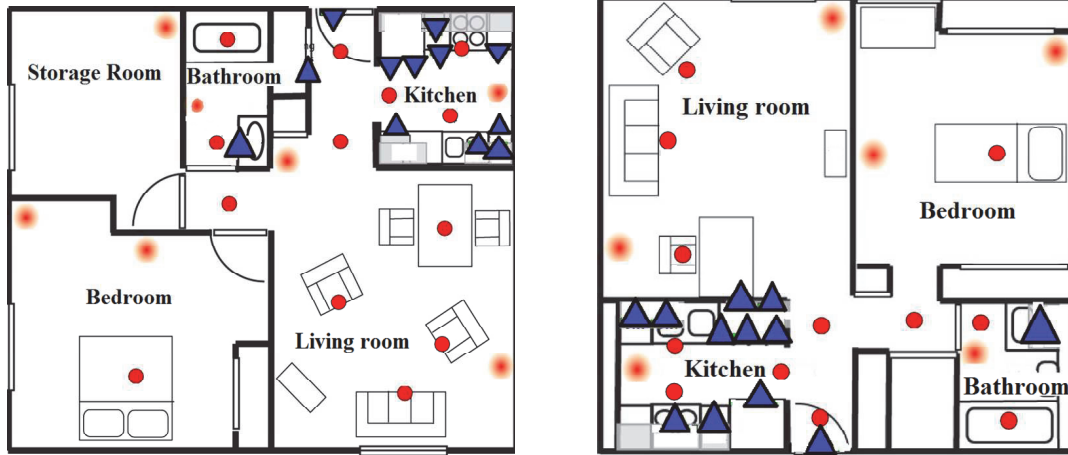


Figure 10: The sensor layouts of Apt1 (left) and Apt2 (right)

Home, Meal Preparation, Personal Hygiene, Sleeping in Bed, Sleeping not in Bed, and Taking Medication. Therefore, each sensor event in our dataset consists of a timestamp, a sensor ID, and an optional activity label. Events in the dataset have been manually annotated with corresponding activity labels by trained researchers who employed visualization tools and interviews with residents to generate accurate ground truth labels.

Evaluation of CRAFFT

In this section, we provide the results of running CRAFFT on our smart home data and compare them with the related prediction approaches. All of the prediction results are evaluated using 10-fold cross validation method.

As already mentioned, CRAFFT consists of two prediction steps, which are the next activity features and the next activity label prediction. The prediction accuracies related to both steps using data gathered from Apt1 and Apt2 are provided in Table 3. As shown, the results suggest that the activity label prediction accuracy of CRAFFT for Apt2 is 73.68% which is stronger than the Apt1’s accuracy of 64.59%. The next activity features prediction results of CRAFFT are also provided in Table 3. The results indicate that the location prediction accuracy for Apt1 is 46.60%, while it is 60.94% for Apt2. Furthermore, the time of day prediction accuracy for Apt1 is 88.10%, whereas it is 86.41% for Apt2. Finally, the day of week prediction accuracy for Apt1 is 96.74%, while it is 96.48% for Apt2. It is interesting to note that while CRAFFT total activity prediction for Apt2 outperforms its accuracy for Apt1, location is the only feature whose prediction accuracy is greater for Apt2 as compared to that of for Apt1. This result underscores the significance of the location feature in our prediction model.

To present the results for the related structures, we first provide the activity prediction results using the CEFA model on our datasets. The results suggest the activity label prediction accuracies of CEFA are 45.41% and 60.05% for Apt1 and Apt2, respectively. In contrast to the prediction results of CRAFFT represented in Table 4, the CEFA model shows

a 19.18% decline for Apt1 and a 13.63% decline for Apt2. It is worth mentioning that the CEFA model does not predict the next activity features, so there is not entry for the activity feature prediction in Table 4.

Table 4: CEFA Prediction accuracy for Apt1 and Apt2

| | CEFA Activity Label Prediction |
|-------------------|--------------------------------|
| Apartment1 | 45.41% |
| Apartment2 | 60.05% |

Moreover, we present the activity prediction accuracy of naïve Bayes (NB) in Table 5. Compared to the prediction results of CEFA presented in Table 4, NB shows a 3.31% decline for Apt1 and a 0.11% decline for Apt2. It is worth adding that NB ignores the meaning and temporal relations among variables represented in Fig. 3.

Table 5: NB Prediction accuracy for Apt1 and Apt2

| | NB Activity Label Prediction |
|-------------------|------------------------------|
| Apartment1 | 42.10% |
| Apartment2 | 59.94% |

In order to compare the results of CRAFFT with non-network based approaches, we present the prediction results for the Support Vector Machines (SVMs) and Multi-layer Perceptron (MLP) algorithms on our datasets. Table 6 provides the prediction accuracies of SVMs with a polynomial kernel on Apt1 and Apt2. The results suggest a 18.53% decline for Apt1 and a 26.15% decline for Apt2 as compared to the results of CRAFFT presented in Table 3. In Table 6, we also provide the prediction results of the MLP algorithm, where the results implies a 28.10% decline for Apt1 and a 38.88% decline for Apt2 as compared to the results presented in Table 3 for the CRAFFT model. The discussed comparisons imply the superior performance of the proposed network-based model over the non-network based algorithms for the task of temporal prediction.

Table 3: Activity label prediction accuracies of CRAFFT

| | Next Activity Feature Prediction | | | Next Activity Label Prediction |
|-------------|----------------------------------|-------------|-------------|--------------------------------|
| | Location | Time of Day | Day of Week | |
| Apt1 | 46.60% | 88.10% | 96.74% | 64.59% |
| Apt2 | 60.94% | 86.41% | 96.48% | 73.68% |

Table 6: Activity label prediction accuracies of SVMs and MLP

| | SVM | MLP |
|-------------------|--------|--------|
| Apartment1 | 46.06% | 59.71% |
| Apartment2 | 45.68% | 58.41% |

In Fig. 11, we summarize the activity label prediction comparison between CRAFFT and the other discussed structure-based and non-structure based algorithms.

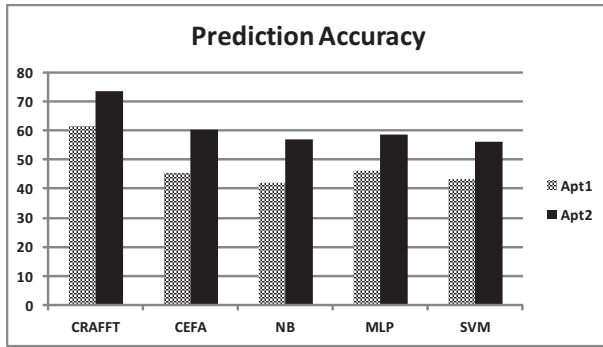


Figure 11: The activity label prediction accuracy comparison among discussed models.

Finally, we present a sample result for our proposed relative start time prediction method. Considering that the current and predicted activities are “Breakfast” and “Taking Medication”, Fig. 12 illustrates the predicted time offset between the mentioned activities. As already discussed, we model the time offset using a normal distribution. After discarding the outliers, the results illustrated in Fig. 12 suggest that when the “Taking Medication” activity occurs after the “Breakfast” activity, the typical time offset between them falls in the following range (see Fig. 9):

$$[\mu - 2\sigma, \mu + 2\sigma] = [31 - 2 * 8, 31 + 2 * 8] = [15, 47] \text{ min.}$$

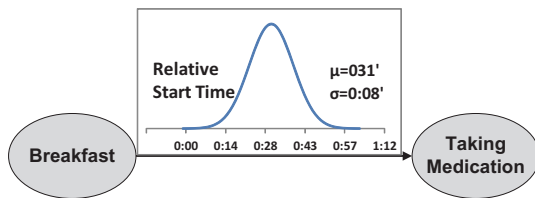


Figure 12: The predicted relative start time for Taking Medication, when it happens after the Breakfast activity.

Conclusions and Future Work

In this paper, we proposed an activity prediction approach, called CRAFFT, using Bayesian networks. CRAFFT uses a novel two-step inference process to predict the next activity features and then the next activity label. Furthermore, we proposed an approach to predict the relative start time of the next activity which was based on modeling the relative start time using the continuous normal distribution and outlier detection. We evaluated CRAFFT and other related prediction approaches using real data collected from two smart home apartments. The Experimental Results suggest the superior performance of the CRAFFT model over the related prediction approaches. In future, we aim to extend the proposed two-step inference process to the cases where the data is partially observable. Moreover, we plan to use the proposed approach for prompting-based interventions.

References

- BrainAid.com. 2013. PEAT: Android application for people with cognitive challenges.
- Cook, D. J., and Das, S. K. 2005. *Smart Environments: Technology, Protocols and Applications*. Wiley Series on Parallel and Distributed Computing.
- Crandall, A. S. 2011. *Behaviometrics for Multiple Residents in a Smart Environment*. Ph.D. Dissertation, Washington State University.
- Helal, S.; Mann, W.; El-Zabadani, H.; King, J.; Kaddoura, Y.; and Jansen, E. 2005. The gator tech smart house: A programmable pervasive space. *IEEE Computer Society* 38(3):50–60.
- Holder, L. B., and Cook, D. J. 2013. Automated activity-aware prompting for activity initiation. *Gerontechnology* 11(4):534–544.
- Kaushik, P.; Intille, S.; and Larson, K. 2008. User-adaptive reminders for home-based medical tasks. a case study. *Methods of Information in Medicine* 47(2):203–207.
- Pollack, M. 2005. Intelligent technology for an aging population: The use of ai to assist elders with cognitive impairment. *AI Magazine* 26(2):9–24.
- Rashidi, P., and Cook, D. J. 2009. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on Systems, Man, and Cybernetics, Part A (TSMC)* 39(5):949–959.
- Shachter, R. D. 1998. Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proceedings of the 14th Conf. in Uncertainty in Artificial Intelligence*, 480–487.