Interleaved Activity Recognition for Smart Home residents

Geetika Singla and Diane J. Cook[1]
Washington State University
{gsingla,cook}@eecs.wsu.edu

Abstract. Smart environments rely on artificial intelligence techniques to make sense of the sensor data and to use the information for recognition and tracking activities. However, many of the techniques that have been developed are designed for simplified situations. In this paper we discuss a more complex situation, namely recognizing activities when they are interweaved in complex and realistic scenarios. This technology is beneficial for monitoring the health of smart environment residents and for correlating activities with parameters such as energy usage. We describe our approach to interleaved activity recognition and evaluate various probabilistic techniques for activity recognition. We validate our algorithm on real sensor data collecting in our smart apartment testbed.

Keywords. activity recognition, naïve Bayes classifier, Markov model

1. Introduction

With the dramatic improvements in intelligent environments that the community has witnessed, researchers have begun investigating the use of smart environment data and artificial intelligence techniques to automate caregiver tasks. With the development of smart environment technologies, at-home automated assistance can allow people with mental and physical challenges to lead independent lives in their own homes. Researchers have generated ideas for designing smart environment software algorithms that track the location and activities of residents, that generate reminders, and that react to hazardous situations [1]. Because performing Activities of Daily Living (ADLs) are a normal part of independent living, a key technology for monitoring an individual's health is to identify and track a smart environment resident's regular performance of these activities.

Researchers have investigated the recognition of resident activities using a variety of mechanisms. While these approaches have been fairly effective for certain types of tasks, they face difficulties when applied in general situations because they make simplifying assumptions about the nature in which the activities are performed. In this paper we address the problem of recognizing activities in complex situations. In particular, we investigate approaches to recognizing activities that are interrupted and interwoven. We assess the effectiveness of each approach using real data collected in the CASAS smart apartment testbed.

[1] Corresponding Author.

2. CASAS

The activity recognition algorithms we introduce in this paper are part of the CASAS smart environment project. In order to evaluate our algorithms, we test them using data collected in the CASAS smart apartment testbed. The apartment, shown in Figure 1, includes three bedrooms, one bathroom, a kitchen, and a living/dining room. This environment is equipped with motion sensors, temperature sensors, humidity sensors, contacts switches in the doors, and item sensors on key items. We have designed special-purpose sensors to detect water usage and stove burner usage and use the Asterisk software to monitor outgoing phone usage. All of these sensors have the advantage of being non-obtrusive and easy to monitor remotely.

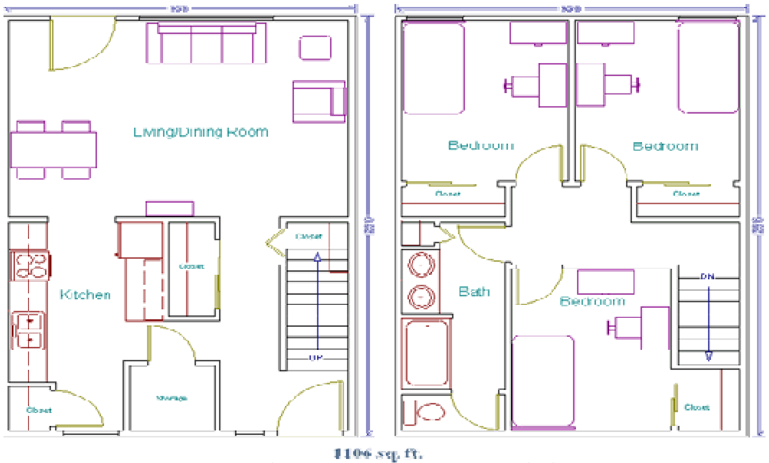


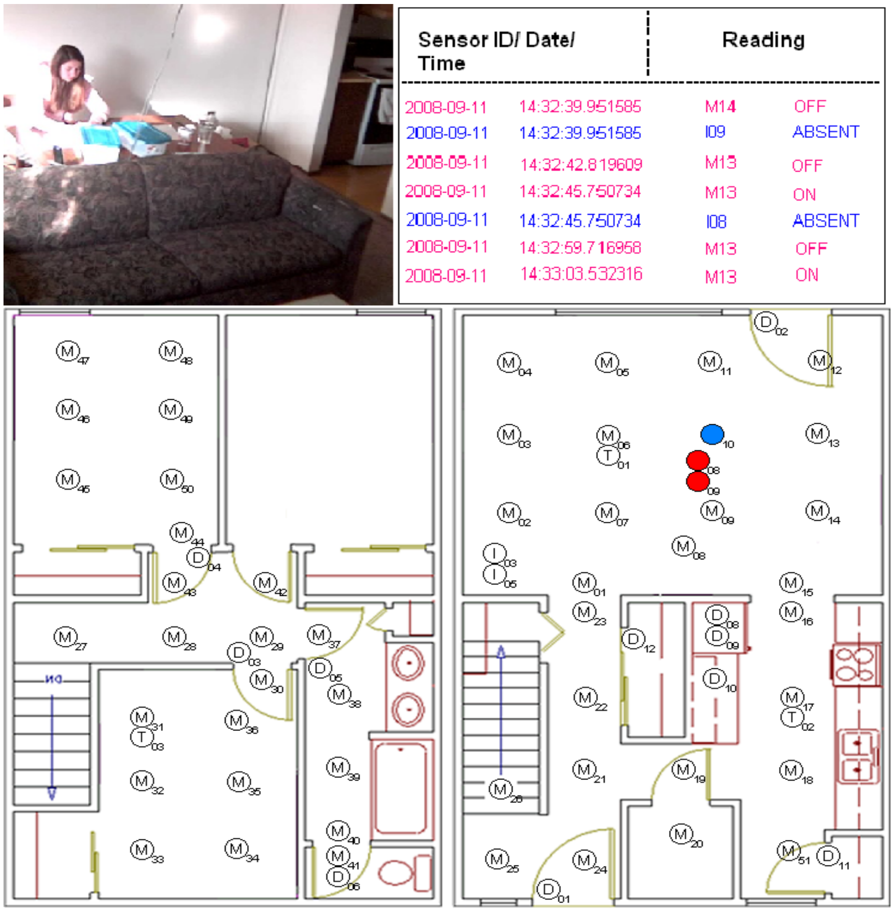
Figure 1. The CASAS smart apartment testbed.

3. Activity Recognition

While collecting sequences of sensor readings in a smart environment is valuable, determining what activities these sequences represent is a more challenging task. We employ probabilistic models to automatically recognize the ADLs activities when they are interleaved. Researchers have investigated the recognition of resident activities using a variety of mechanisms such as naïve Bayes classifiers, Markov models, and dynamic Bayes networks [2,3,4,5]. These approaches have been fairly effective for limited types of tasks. However, these limitations prevent the techniques from being applied in more realistic situations. In particular, all of the approaches to date make the assumptions that these activities are performed sequentially and independent of other activities; and that there is only one resident to monitor at a time.

In contrast, we are designing algorithms that probabilistically identify the activities while they are being performed in not only isolated (i.e. sequential), but also complex (i.e. interleaved and concurrent) manners in real life. Given a sequence of observed sensor events we can probabilistically determine which activity the sequence belongs to. We describe and analyze several probabilistic classifiers to perform this task.

4. Experiments



Sensor ID/ Date/ Time			Reading
2008-09-11	14:32:39.951585	M14	OFF
2008-09-11	14:32:39.951585	I09	ABSENT
2008-09-11	14:32:42.819609	M13	OFF
2008-09-11	14:32:45.750734	M13	ON
2008-09-11	14:32:45.750734	I08	ABSENT
2008-09-11	14:32:59.716958	M13	OFF
2008-09-11	14:33:03.532316	M13	ON

Figure 2. The web cam image in the upper left shows a participant performing the "Write birthday card" activity. The activity generates the sensor readings shown in the upper right and visualized at the bottom.

For this study, we selected 8 ADLs important for everyday functioning [6]. The selected activities include "Fill medication dispenser", "Watch DVD", "Water plants", "Converse on Phone", "Write Birthday Card", "Prepare meal", "Sweep and dust", and "Select an outfit". To validate our interleaved activity recognition algorithms we recruited 20 volunteer participants to perform these activities in the smart apartment. First, each of the participants performed these tasks sequentially. The participants were next instructed to perform all of these activities by interweaving them as they liked with a goal of being efficient in performing the tasks. In this case the participants jumped back and forth between tasks, sometimes even performing several tasks at once. The data collected for both of these runs was then manually labeled with the correct activity ids for training purposes.

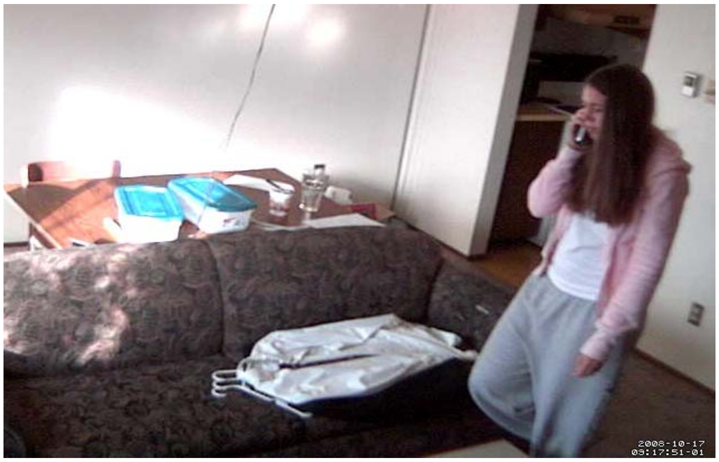


Figure 3. The participant is interweaving 2 tasks. Here she transitions from midway through the "Select an Outfit" activity to the "Phone Conversation".

We implemented several probabilistic classifiers to find out an approach that works best for identifying interleaved activities. These probabilistic models include a Naïve Bayes classifier, a Hidden Markov Model (HMM), a HMM with a time window, a frequency-based HMM with a sliding window and a frequency-based HMM with a shifting window. We used these models to automatically label every sensor event from the test set with an activity label so that the smart environment agent is aware of the current activity at any given time. In the results we report below, we show accuracy results generated using three-fold cross validation on the participant data.

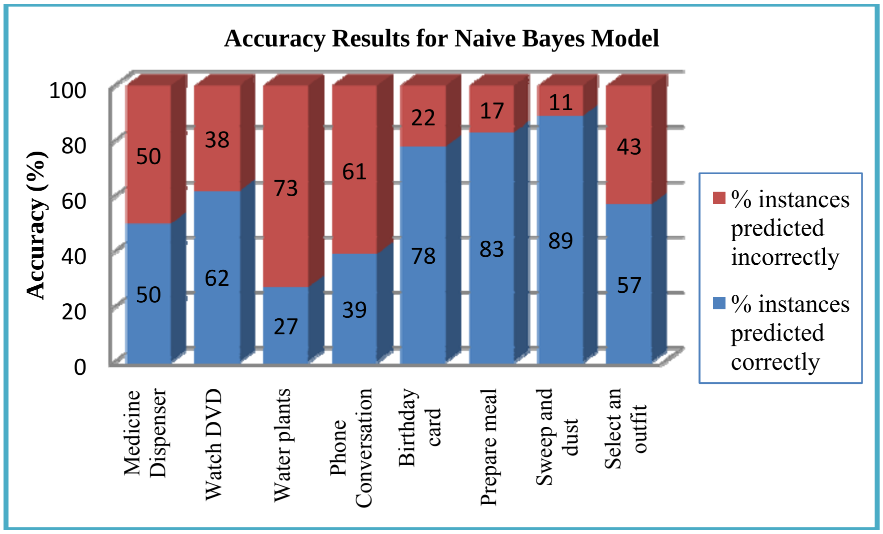


Figure 4. Bar graph showing performance of Naive Bayes classifier broken down by activity.

In spite of its simple design and over-simplified assumptions, naive Bayes classifiers (NBCs) often work much better in many complex real-world situations than expected. This classifier assumes variables to be independent and expresses the conditional distribution over the class variable C as:

$$p(C|F_1, \ldots.., F_n) = (1/Z)\ p(C) \prod n_i = p(F_i\ |C) \qquad (1)$$

where Z is a scaling factor dependent only on features F_1, F_2, …, F_n. For our research, we use the maximum a posteriori rule to pick the most probable activity. The classifier returns the class value given by:

$$\text{argmax}_C\ p(C)\ ((1/Z)\ p(C = c) \prod^n{}_{i=1} p(F_i\ = f_i|C = c) \qquad (2)$$

In our implementation, we treat every sensor as a feature and learn sensor value probability distributions for all activities. The independence assumption works well here because high probability values are generally concentrated over different sensors for different activities. The NBC results in an overall accuracy of 66.08% in predicting interleaved activities, as shown in Figure 4. As the NBC does not take transition probabilities into consideration, some activities could not be identified very well by the model. In order to make use of the information resulting in transition between states, we next implemented a Hidden Markov Model (HMM).

Our HMM algorithm represents each activity as a hidden state. Next, every sensor is treated as an observable state in the model due to the fact that every sensor in the environment generates observable readings. One collective HMM model is generated for all activities, with the horizontal and vertical transition probabilities learned from the training data. In the testing phase the new data is run through the HMM as a continuous stream of data and the activity (hidden node) with highest probability value (calculated using the Viterbi algorithm [7]) is returned as the activity label. Figure 5 shows a section of the HMM for interleaved activities and the accuracy results for our HMM are shown in Figure 6.

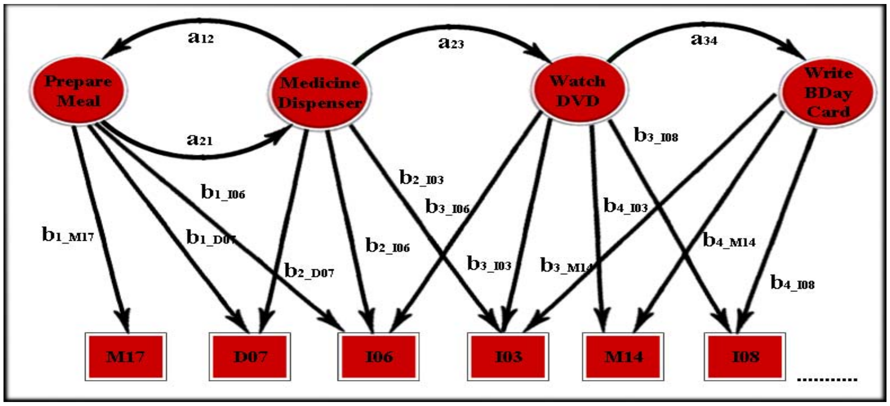


Figure 5. A section of a hidden Markov model designed to model interleaved activities. The circles refer to the activities (i.e., the hidden states) and the rectangles refer to the observable states. Value 'a_{ij}' refers to the probability of transitioning from activity *i* to activity *j*, and the $b_{i\_sensorId}$ values represent the emission probability of the possible observations in a particular hidden state.

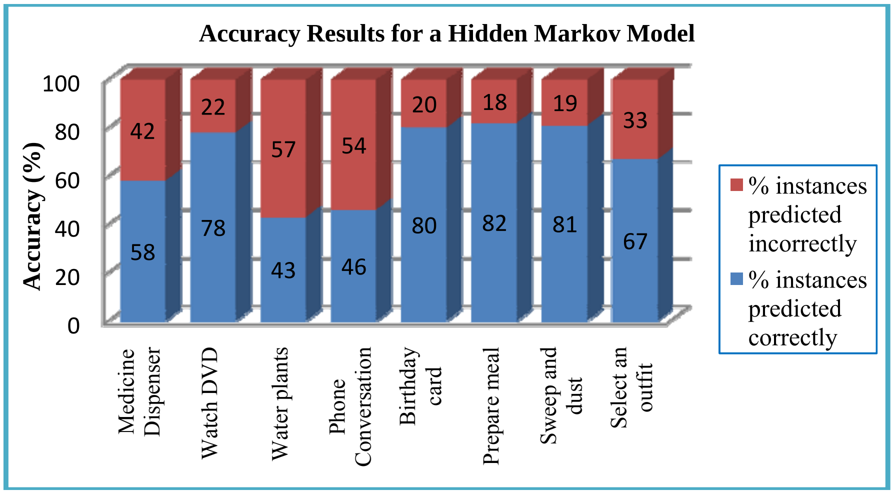


Figure 6. Bar graph showing performance of Hidden Markov Model in recognizing interleaved activities.

The average accuracy of our hidden Markov model in predicting interleaved activities is 71%, as shown in Figure 6. The model shows an overall increase of 10% in accuracy over the naïve Bayes model. A limitation of this plain HMM is that it moves very slowly from recognizing one activity to the next. As a remedy, we used a sliding window where the size of the window is governed by the number of sensor events it contains. This limits the history of sensor events that the model remembers at any given time which in turn helps in keeping the probability values low for all activities and in making a faster switch from the current to the next activity. This technique works in two phases: in the first phase, alternative window sizes are tested on 2/3 of the data; in the second phase, the automatically-selected window is used to label the rest of the data.

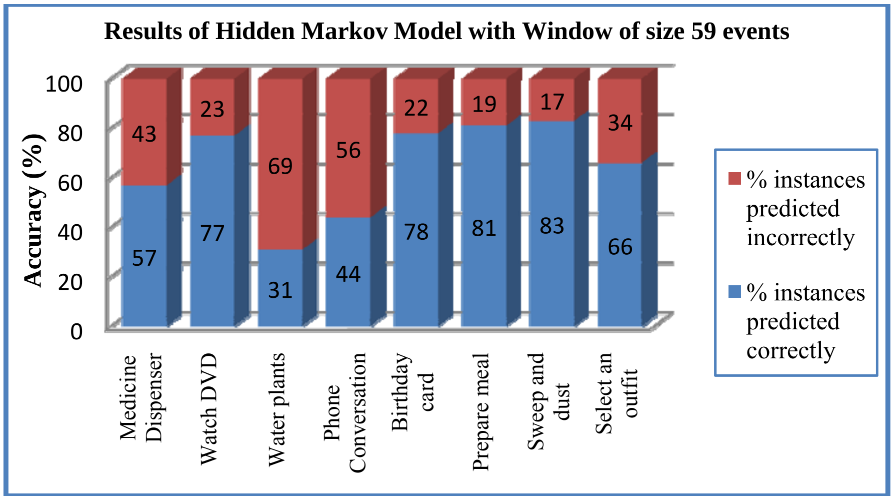


Figure 7. Bar graph showing accuracy of the model for a window size of 59 sensor events.

This model results in an overall average accuracy of 63.55% in recognizing activities for a window size of 59 sensor events. The activity “Sweep and dust” is predicted with the highest accuracy of 83%. This model performs poorly for activities “Water plants” and “Phone conversation” and shows lowest accuracies of 31% and 44%, respectively. This can be attributed by the fact that these activities contain fewer number of sensor events, and so a smaller window would have recognized these tasks more precisely. The window size thus calculated using this technique works for most, but not all type of activities.

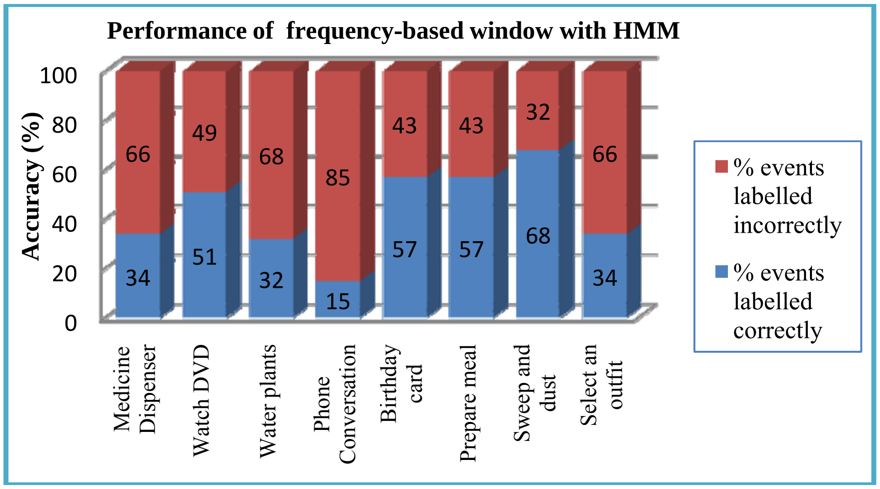


Figure 8. Bar graph showing performance of window based HMM using frequency.

Another approach that we tried to solve this problem was to use the HMM with a frequency-based window. Instead of labeling each sensor event with an activity label, however, this time we label the sensor event with the activity label of the most frequent activity in the entire window. This model shows a maximum overall accuracy of 56.75% in identifying activities for a window size of 8 sensor events. This technique suffered from the same problem of making a very slow transition from the current to next activity and did not perform as well as expected. At the time of transition, the previous activity always has a higher frequency leading to inaccurately labeling the new sensor event and thereby degrading the performance of the algorithm. Figure 9 summarizes the accuracy of all of the approaches to facilitate direct comparison.

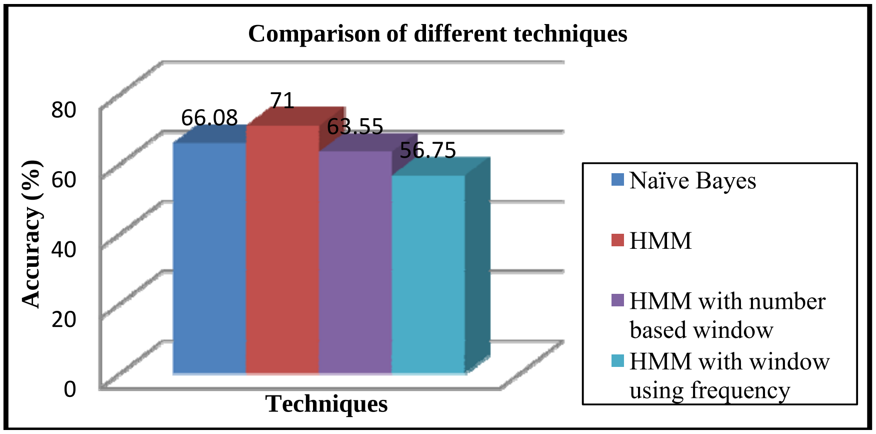


Figure 9. Bar graph comparing performance of all techniques in recognizing interleaved activities.

5. Conclusions and Future Work

In this paper we address the problem of recognizing interleaved activities in smart environments. Not only do we demonstrate that these activities can be recognized by sensors in physical environments, but we also show enhancements that help the classifiers improve recognition accuracy. This increased accuracy will be important as we move on to our next steps. In particular, we will next be investigating techniques for detecting missing or incorrect steps in the activity, and for recognizing activities when there are multiple residents in the environment.

This research lays the groundwork for tools that automatically monitor and assist individuals with special health needs. We believe these technologies are essential to provide accessible and low-cost health assistance in an individual's own home. Furthermore, investigating these issues will be imperative if we want to adequately care for our aging population and provide the best possible quality of life for them and, ultimately, for ourselves.

References

[1] M. Pollack. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Magazine*, 26(2):9-24, 2005

[2] C. Wren & E. Munguia-Tapia. Toward scalable activity recognition for sensor networks, *Proceedings of the Workshop on Location and Context-Awareness*, 2006

[3] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50-57, 2004

[4] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26(1):119-126, 2007

[5] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. *PervasiveHealth*, pages 258-263, 2008

[6] S.T. Farias, D. Mungas, B. Reed, D. Harvey, D. Cahn-Weiner and C. DeCarli. MCI is associated with deficits in everyday functioning. *Alzheimer Disease and Associated Disorders*, 20:217-223, 2006

[7] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory* 13(2):260–269, 1967.