ACTIVITY RECOGNITION IN COMPLEX SMART ENVIRONMENT SETTINGS

By

GEETIKA SINGLA

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

WASHINGTON STATE UNIVERSITY
Department of Electrical Engineering and
Computer Science

May 2009

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of GEETIKA SINGLA find it satisfactory and recommend that it be accepted.

Diane J. Cook ______________________________
(Chair)

Lawrence B. Holder ______________________________

Behrooz A. Shirazi ______________________________

ACKOWLEDGEMENTS

This research project could not have been possible without the guidance and support of numerous people. I would like to thank my supervisor, Dr. Diane J. Cook for her abundant help and invaluable support and guidance as I pursued my research. I also wish to express deepest gratitude to the members of the supervisory committee, Dr. Lawrence B. Holder and Dr. Behrooz A. Shirazi as without their knowledge and assistance this study would not have been successful.

Lastly, I would like to thank my family and friends for their encouragement and support throughout the duration of my studies.

ACTIVITY RECOGNITION IN COMPLEX SMART

ENVIRONMENT SETTINGS

ABSTRACT

by Geetika Singla, M.S.
Washington State University
May 2009

Chair: Diane J. Cook

Smart environments rely on artificial intelligence techniques to make sense of the sensor data and to use the information for recognizing and tracking activities. However, many of the techniques that have been developed are designed for simplified situations. In this thesis we investigate more complex situations like recognizing activities when they are interweaved in realistic scenarios and when the space is inhabited by multiple resident performing tasks concurrently. 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 sequential, interleaved and concurrent (multi-resident) activity recognition and evaluate various probabilistic techniques for activity recognition. In addition to demonstrating that these activities can be recognized by sensors in physical environments using Markov and Hidden Markov models, we also show variants of these models that help in improving the recognition accuracy. We validate our algorithm on real sensor data collected in the CASAS smart apartment testbed.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

One of the most dramatic changes in the population of the United States is the growing number of older persons in the population, both in absolute numbers and in the percentage of the total population they represent. The graying of America is a trend that is really only beginning. In 2010, when the leading edge of the baby boom cohort starts to hit 65 years of age, the relative size of the elderly population will begin to increase dramatically. According to US Census Bureau projections, the relative size of the older population in 2015 will reach 14% as compared to its current level of 12.4%. In 2030, older adults are projected to comprise almost 20% of the total population [1]. It is estimated that there are currently nearly 18 million older adults with dementia in the world, and by 2025, this number is expected to reach 34 million [2]. With the rapidly increasing aging population, the burden on the health care system is also increasing and measures must be taken in order to provide health care to the elderly.

One approach towards solving this problem is to promote healthy lifestyles at home, thus reducing the need for health care facilities and disease treatment in hospitals. With the convergence of technologies in machine learning and pervasive computing, home automation systems have emerged that make it possible for the elderly and disabled to live by themselves at home. The aging population has also generated a significant interest by the government as well as industry leaders to develop home automation systems for the elderly [3]. Recently there has been extensive research towards developing smart environments by integrating various machine learning and artificial intelligence techniques into home environments that are equipped with sensors and actuators. A smart environment is an intelligent environment, which perceives the state of the space using sensors, analyzes the state using learning and reasoning techniques, and provides assistance to the residents in their daily living to maintain safety of the environment and its residents.

Smart homes will make it possible for the elderly and people with disabilities to stay in their homes where they feel comfortable, instead of being moved to an eldercare facility. Researchers believe that the continuous assessment of physical activities of an individual is a useful basis for monitoring well being and detecting initial decline in health and functional ability [4, 5]. In order to function independently at home, inhabitants need to be able to complete key Activities of Daily Living (ADL) [5]. Hence, it is very important to be able to detect activities in a smart environment in order to realize such a system. The long-term aim of our project is to make sure that individuals are performing normal Activities of Daily Living and to assist them in performing these activities when needed to help them live independently in their own homes.

To be able to track an ADL is very important from the perspective of elder care but at the same time, manual assessment of ADLs is a fundamental problem in elderly care. Two specific sets of activities that describe the functional status of a person have been defined by psychologists [6]. These are Activities of Daily Living (ADLs, the basic tasks of everyday life) and Instrumental Activities of Daily Living (iADLs, activities representing interaction with the physical and social environment). The ADLs include activities like bathing, dressing, toileting, transferring, continence and feeding [6]. In contrast, iADLs fall within categories such as using telephone, shopping, food preparation, housekeeping, doing laundry, transportation, taking medications and handling finances [6]. We hypothesize that machine learning algorithms can be designed to monitor the state and completion of such ongoing activities in a smart space.

As a first step, in this thesis we try to answer few basic questions like whether activities can be detected precisely using simple sensors in a space and how effectively can they be recognized in real-world settings. In our work, we show that models of daily activities can be learned from sensor events collected by a smart environment. By learning models for each task, ADL initiation and completion can be automatically detected, even when the activity is performed in a real-world setting, the resident is switching between tasks, and when additional people are performing activities in the environment. If we can successfully recognize ADLs in these settings then we can use the technologies to perform automatic assessment of an individual's well being and provide the foundation for reminder-based interventions.

In the following sections, we will provide an overview of existing smart home technology in the field of activity recognition and will introduce our initiatives. In the next chapter, we will describe our experimental set up and the design and implementation of CASAS as a smart home environment which utilizes data mining and machine learning techniques to discover frequent patterns and recognize activities. In the next chapters, we will discuss various methods and approaches that we adopted to recognize activities in different scenarios and summarize the experiments and their results to assess the effectiveness of our approaches. In the last chapter we conclude with a summary of the research and a discussion of alternative and future research directions.

CHAPTER TWO

OVERVIEW AND RELATED WORK

Although there is a growing interest in adding intelligence to our living and working environments, only recently has the convergence of technologies in machine learning, pervasive computing, and sensor networks made the idea of smart environments a reality. 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. One of the most imperative parts of the smart environment technology is the ability to recognize activities. In this chapter, we discuss various data collection methods used by researchers for the purpose of activity recognition. We then discuss the probabilistic models used in related research works for recognizing activities. This is followed by an overview of the applications of activity recognition system in different applications domains. In the end, the chapter summarizes some of the problems and challenges associated with these research works and discusses how we plan to overcome these challenges.

Mihailidis, et al. [7] have proposed an intelligent environment for older adults with dementia that comprises of 3 modules: *tracking:* monitors the actions of the user by determining the spatial coordinates of the person's body in the environment; *planning*: determines what step the user is completing, whether the step being completed is correct, and which activity (i.e. sequence of steps) the person is attempting, and *prompting:* detects that if the user has made an error, such as completing a step out of sequence or missing a step altogether, and selects and plays a prompt.

Data collection techniques

Liao, et al. [8, 9] used traces of GPS data in their experiments to extract and label a person's activities. Their approach detects and classifies the significant places and activities of a resident by taking high-level context into account. They construct hierarchical activity model that encodes the complex relations among GPS readings, activities and significant places.

Significant work [10, 11, 12] has been done in the field of activity recognition using visual cues and cameras in computer vision. Vaswani, et al. [13] presented an approach to monitor activities using video data. They could learn the pattern of normal activities and detect abnormal events from a very low resolution video. Robertson, et al. [14] developed a system for human behavior recognition in video sequences by modeling it as a stochastic sequence of actions. They achieved action recognition via probabilistic search of image feature databases representing previously seen actions. Hongeng, et al. [15] present a new representation of activities by considering them as composition of action threads, each thread being executed by a single resident. Visual cues and videos are informative sources of data but on the other hand, they are very obtrusive and are not favored by older inhabitants. Moreover, storing and processing videos can be computationally very expensive if there is a need to work on data collected over a long period of time.

Attaching sensors to the body is a promising approach to acquire more precise data about objects under use, the human movement and the social environment. These sensors could be in the form of RFID tags placed on various objects in the smart space, and a RFID tag reader worn by the inhabitant [16, 17]. Object-interaction based activity recognition has been realized by Patterson, et al. [16] in a very realistic setting. In their experiment, they outfitted the kitchen with 60 RFID tags placed on every object touched by the user, to capture the identity of the objects being manipulated. Gu et al. have tried to solve a very similar problem in their epSICAR project [17] by placing RFID tags on various objects in the smart space. Subramanya, et al. [18] used a mountable sensor board to get asynchronous GPS measurements which they used to create a dynamic graphical model that estimates both activity and spatial context of the individuals.

Stikic, et al. [19] further strengthen the belief that the use of miniature sensors placed in the environment or worn by a person has great potential in effective and unobtrusive long term monitoring and recognition of ADLs. They built an effective and unobtrusive activity recognition system based on the combination of the data from two different types of sensors; RFID tag readers and accelerometers. Their experiments also show promising results calculated for a non-scripted datasets of 10 housekeeping activities performed by 12 subjects. Ravi, et al. [20] attempted to recognize activities using a single tri-axial accelerometer worn near the pelvic region.

Wu et. al [21] leverage sparse and noisy readings from RFID tagged objects, along with common-sense knowledge about which objects are likely to be used during a given activity, to bootstrap the learning process. They combine RFID and video data to jointly infer the most likely activity and object labels. Their experiments show that the combination of visual object recognition with RFID data is significantly more effective than the RFID sensor alone.

The systems that attach sensors to devices in the environment and to the human body in combination with sensors that observe the scene using audio, visual, and magnetic sensors are better informed of the environment. These sensors help in more accurately determining which object the inhabitants are using, their motion and interaction with environment at any given time and can be more powerful in recognizing human activities. But at the same time, use of such wearable sensors in smart space is obtrusive and they have to be worn by the inhabitants throughout the duration of their stay in the smart environment. Such devices are often not desirable to older adults who are the potential consumers of such technology.

Probabilistic Models

While collecting sequences of sensor readings in a smart environment is valuable, determining what activities these sequences represent is a more challenging task. Many researchers [22, 23, 24] exploited probabilistic models to recognize activities and detect anomalies to support individuals living at home with special needs. Hu et al. [22] propose a two-level probabilistic framework called CIGAR (Concurrent and Interleaving Goal and Activity Recognition) for recognizing both concurrent and interleaved activities. They use skip-chain conditional random fields (SCCRF) for modeling the interleaved tasks, and correlation graph for adjusting inferred probabilities which they use to model the concurrent tasks. In a similar work, Wu et al. [23] used Factorial Conditional Random Fields (FCRFs) for recognition of multiple concurrent activities. They also designed experiments to compare their FCRFs model with Linear Chain Condition Random Fields (LCRFs) in learning and performing inference with the MIT House n data set, which show that their model improves the F-score in concurrent activity recognition for up to 8%.

Gong, et al. [25] employed the Dynamic Probabilistic Networks (DPNs) for modeling temporal relationships which they used for behavior interpretation. They used Dynamically Multi-Linked Hidden Markov Model to interpret group activities involving multiple objects captured in an outdoor scene. Their model is very useful in precisely recognizing group activities in a noisy outdoor scene. Vail, et al. [24] consider activity recognition as a temporal classification problem and explore the differences in performance between the discriminatively trained Conditional Random Field and the generative Hidden Markov Model. They also examine the effect of incorporating features which violate independence assumptions between observations. Their experiments show that the discriminatively trained CRF performs (atleast as well as or) better than an HMM especially in cases where features depend on observations from many time steps.

Wu et al. [21] investigated the dynamic Bayesian network model to infer the most likely activity and object labels in their work. Wilson et al. [26] also used Dynamic Bayes networks to exploit the synergy between location and activity for simultaneous resident tracking and activity recognition. Oliver, et al. [27] have built a probabilistic machinery based on a tiered formulation of dynamic graphical models that we refer to as Layered Hidden Markov Models (LHMMs), that can provide real-time interpretations of human activity in and around an office. Through their research, they show that multi-layered architecture of their model makes it more robust to typical variations within office environments, such as changes of lighting and acoustics, and improves the performance of their model when transferred to new office spaces without the need of much tuning through retraining.

Most of the research work done so far focuses on recognizing activities in simple artificial scenarios and when activities are performed sequentially. In our research, we focus on recognizing activities in real-time situations like when different activities are interleaved together and when there are multiple residents in the smart space performing tasks concurrently, in addition to recognizing activities in simple scenarios. In our experiments, we use the Markov and the hidden Markov model for recognizing activities, and discuss their performance in different scenarios. We also use several variants of the HMM to improve the recognition accuracy, which are discussed in the following chapters.

Applications

There is a significant body of literature surrounding the ideas for designing smart environment software algorithms to track the location of residents, to generate reminders, and to react to hazardous situations [8, 28]. Smart environments have also been used to actually determine the cognitive impairment of the inhabitants. Carter and Rosen [29] demonstrate such an assessment based on the ability of individuals to efficiently complete kitchen tasks. Jimison, et al. [30] also provide such an assessment. In their case, individuals are monitored while playing computer games, and assessment is based on factors such as game difficulty, player performance, and time to complete the game.

Moncrieff's, et al. [31] implemented an emotive computing framework by using the concept of anxiety to record anomalies based on deviation from normal behavior. Their anxiety framework is a scalable, real-time approach and can accommodate interleaving event sequences. Barger, et al. [32], categorized the sensor data into individual's days into vacation (at home) and work days. Work by Cook, et al. [33] collected activity data from an apartment dweller and used this to determine increasing, decreasing, and cyclic trends in patterns. Once a baseline is established, this can be used to identify sudden changes. Luhr's approach [34] of learning inter-transaction association rules can also be helpful in identifying emerging and abnormal activities.

A limiting factor of these projects is that almost none are being tested on data collected from physical environments. A few test beds do exist in some form, although none are currently focusing on research for automated functional assessment and intervention. These include MavHome project [35], the Gator Tech Smart House [36], the iDorm [37], the Georgia Tech Aware Home [38], and the University of Colorado Adaptive Home [39]. As a result of this and related work, researchers are now beginning to recognize the importance of applying smart environment technology to health assistance [32, 40, 41, 42] and companies are recognizing the potential of this technology for a quickly-growing consumer base.

In our research, we focus on performing activity recognition that is not only accurate, but that also requires a minimum number of sensor devices. It can be cumbersome for the resident to wear many sensors and battery packs mounted over the body. Additionally, we also wish to minimize the overall system cost. In our work, we use generic off the shelf sensors like motion, light, temperature, humidity and simple item sensors. These sensors are inexpensive and are easy to install and are readily available in general stores. The use of video cameras in our research is limited to being only an additional source of information to detect if something goes wrong in data collection. We do not use data from video cameras in aiding our task of activity recognition. Also, for our research work, we rely only on simple sensors that can be deployed in our test bed so that the space looks more homely and there is no need for the residents to wear any sensors. Our work uses data mining and machine learning techniques to model the ADLs from the test data and then analyzes the test data to match the models thus created. In our research, we use the existing state-space models with some modifications to recognize activities and individuals and assess their performance in various scenarios.

Our overall objective is to design software algorithms that will monitor the overall functional wellbeing of individuals at home by detecting ADLs that are being performed by residents in a smart environment. We also test our working hypothesis that smart environment-based measurement techniques can accurately detect completed ADLs. We ultimately use this capability to identify the current step the individual is performing within an ADL and determine which steps of the ADL were skipped or performed out of order. Specifically, after each sensor event we will generate a label for the activity (or set of activities) that the participant is performing, and will use algorithms to identify the current state of the ADLs that were performed and by whom. Details on how methods for performing activity recognition are provided in the next chapter of this thesis.

CHAPTER THREE

EXPERIMENTAL SET UP

The activity recognition algorithms we introduce through this work are part of the CASAS smart environment software architecture. CASAS is an integrated set of components and is composed of various parts that work together to accomplish an interwoven set of tasks, including recognizing activities of daily living, automating them completely or partially, predicting and scheduling automated activities and also adapting to explicit user feedback or observed changes in resident behavior.

In order to evaluate our algorithms, we test them using data collected from volunteer participants performing activities in our smart environment test bed. This test bed is a smart apartment on the WSU campus. The apartment includes three bedrooms, one bathroom, a kitchen, and a living/dining room. The layout of the apartment is shown in Figure 1. 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 relatively easy to monitor remotely.

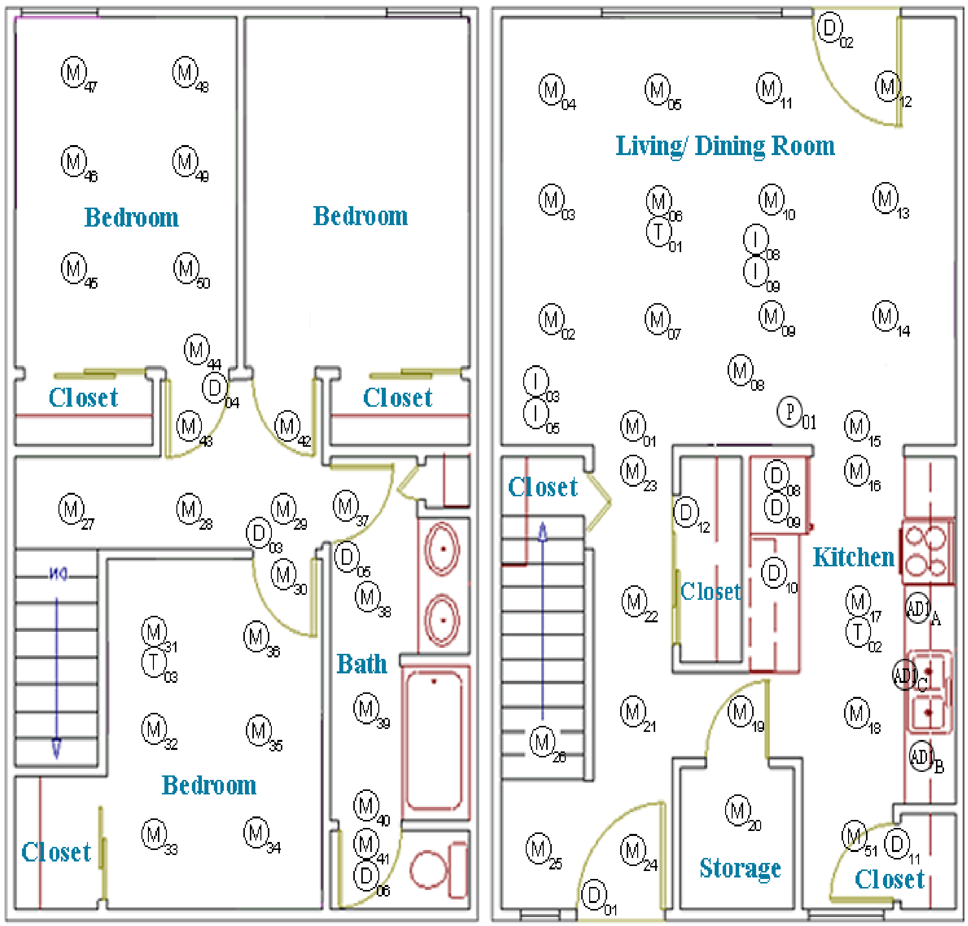


Figure 1 – CASAS Testbed, a three bedroom smart apartment
Sensor Key

M	Motion Sensor (On/ Off sensor)
I	Item Sensor (Absent/ Present sensor)
D	Door Sensor (Open/ Close sensor)
P	Phone sensor (On/ Off sensor)
T	Temperature Sensor (Temperature Value every 10 minutes)
AD1	Gas burner, A and Water Sensor, B and C (analog sensors)

We tagged many items that were used in the experiment with pressure sensors. Some of such items include containers brown sugar, oatmeal, and raisins, a cup, a bowl, a measuring cup, a medication dispenser, a birthday card, a phonebook, and a DVD case. Because some activities require residents to remove items from kitchen cabinets, we also used a cabinet sensor triggered an event when the cabinet was opened or closed. Sensor details for one such kitchen cabinet is shown in Figure 2. An experimenter switch was used to record the starting and ending of each activity during the experiments.

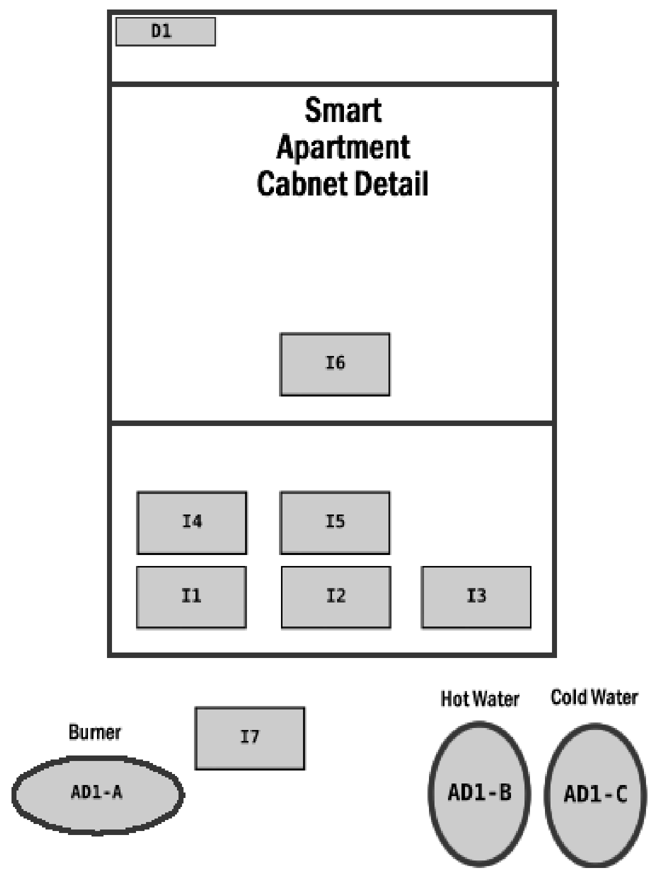


Figure 2 – Detailed view of a kitchen cabinet in the Smart Apartment. I1..I6 represent Item sensors used to record the presence or absence of specific

items placed inside the cabinet. AD1-A, B and C are analog sensors used to monitor gas burner and water usage.

While many contributing smart home technologies are in place, one of the most important issues that have not been explored by smart home researchers is the smart home's ability to recognize the ongoing activities in realistic, complex smart environment situations. Activity recognition in environments such as smart homes is a crucial issue as once the ongoing activities are known, this information can be used to assist the inhabitants in performing their daily tasks, alarm them of hazardous situations and monitor their heath.

To achieve this goal, we employ a combination of data mining and machine learning algorithms to predict the ongoing activities in our smart test bed. We focus on predicting activities in three different realistic scenarios and evaluate the performance of different approaches in all three cases. We first work on recognizing stand alone activities performed by a single inhabitant in isolation. We made use of temporal information in predicting the activities when performed sequentially and in isolation from other activities. In real life, we do not always perform activities in a sequential manner; rather, we try to perform more than one task simultaneously by interweaving them so as to be more efficient. To accommodate such situations we also focused on identifying activities when they are performed by an individual in an interwoven fashion.

The third part of our work focuses on identifying activities in the case where multiple residents perform activities in parallel in the same environment. More than one resident living in a space is a very common situation. We investigate techniques for recognizing activities in this scenario as well when we had 2 participants in our smart home test bed, who are performing different tasks, both in parallel and at different times. The following chapters provide more insight to the issues we addressed, the approaches we consider, our experiments and their results.

CHAPTER FOUR

SEQUENTIAL ACTIVITY RECOGNITION

In order to address the problem of activity recognition, the simplest problem to be considered is to recognize tasks when they are performed in isolation. This refers to the case when inhabitants in a space concentrate on performing only one activity at a time and when there is only one resident in the space at a time. While dealing with this challenge, we design various algorithms that work well in modeling activities and recognizing them in a real-world environment.

We selected several ADLs for testing which are important from the perspective of daily living [43] and can be performed independently. The activities we have selected include both basic ADLs which are disrupted in early-stage dementia [44, 45] and instrumental ADLs which are disrupted in the later stages of dementia [44, 45]. We selected 5 activities for this experiment which include –

1. *Making phone call:* Here the participant obtains a specified number from the phonebook located at the dining room table and calls the number. The participant listens to the recorded message and writes down the cooking directions.

2. *Washing Hands:* The participant washes hands in the kitchen sink.
3. *Cooking:* Here the participant uses the ingredients located in the kitchen cabinet and cooks oatmeal according to the recorded directions.
4. *Eating:* The participant takes oatmeal and some medicine to the dining room table and eats them along with a glass of water.
5. *Cleaning up:* For this activity, the participant cleans the dishes in kitchen sink, and returns all items and ingredients used to their respective locations.

We recruited 20 undergraduate student participants to perform each of these 5 tasks in the smart apartment. The activities were performed separately, with no interleaving or interruptions. We conducted the experiment in the CASAS test bed apartment where various sensors were deployed to log the movements of the participants, as described in Chapter 3. Sensor events were recorded for data collection when these participants were performing the tasks. Data were recorded for each of the 5 activities performed by all 20 volunteers. Hence, we had a total of 100 data sets, 20 for each activity. The script that was given to the participants to perform these tasks sequentially can be found in Appendix A.

The data collected from this study was manually labeled. Specifically, each set of data that belongs to one activity was labeled with the corresponding activity id. The average times consumed in each of these 5 activities were 2 and half minutes, 48 seconds, 7 minutes, 2 minutes and 5 minutes, respectively. The average number of sensor events in the data sets for the activities are 46, 20, 93, 38 and 66 sensor events, respectively. Figures 3 and 4 show images from the "washing hands" and "cooking" activities together with a sample of the sensor events these sequences generate and a visualization of the sensor events.

In our work, we employ Markov models for automatic recognition of the ADLs.

Experimental Setup

```
sensor ID          |
   date / time                      | reading
--------------------------------------------
12048146000000B2   |
           2008-02-12 10:50:45.673225   | ON
12D27E460000000D   |
           2008-02-12 10:50:48.903745   | ON
12048146000000B2   |
           2008-02-12 10:50:49.339849   | OFF
2084A30D0000039B   |
           2008-02-12 10:50:53.27364    |  0.0459382
2084A30D0000039B   |
           2008-02-12 10:51:05.6252     |  0.158401
```

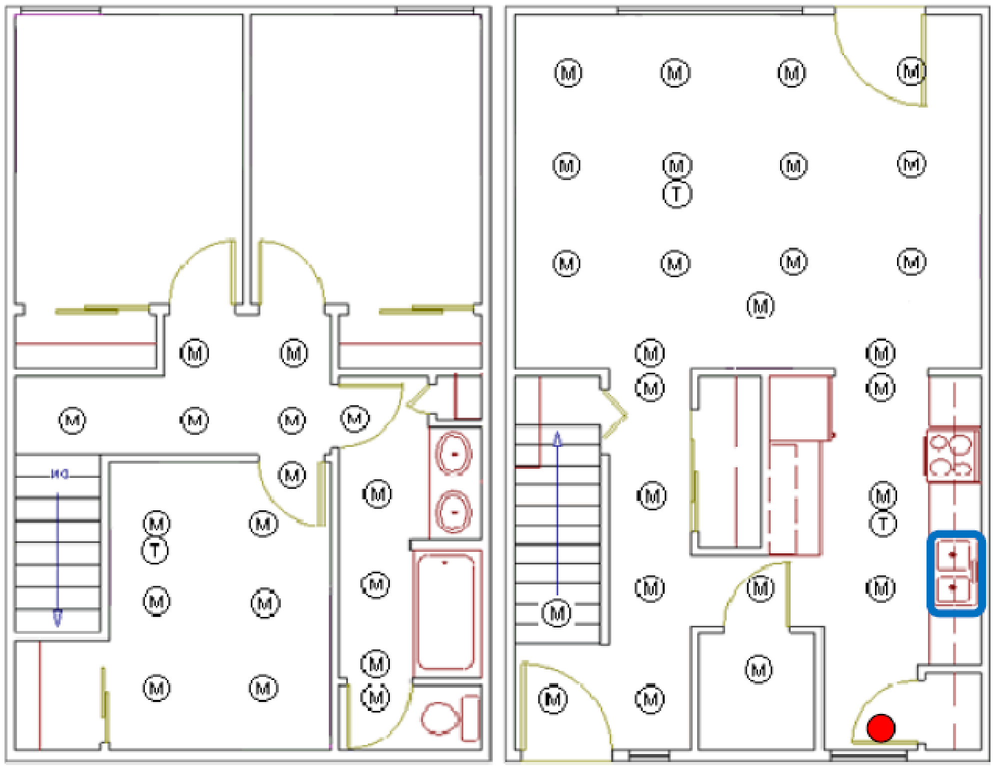


Figure 3 - Recognition of the "washing hands" activity in the smart apartment. The web cam image in the upper left shows a student participant performing the "washing hands" task. The activity triggers the sensor readings shown in the upper right (the first three readings correspond to motion sensors and the last two correspond to non-zero water flow values). A visualization of the sensor activity for the "washing hands" task is shown at the bottom.

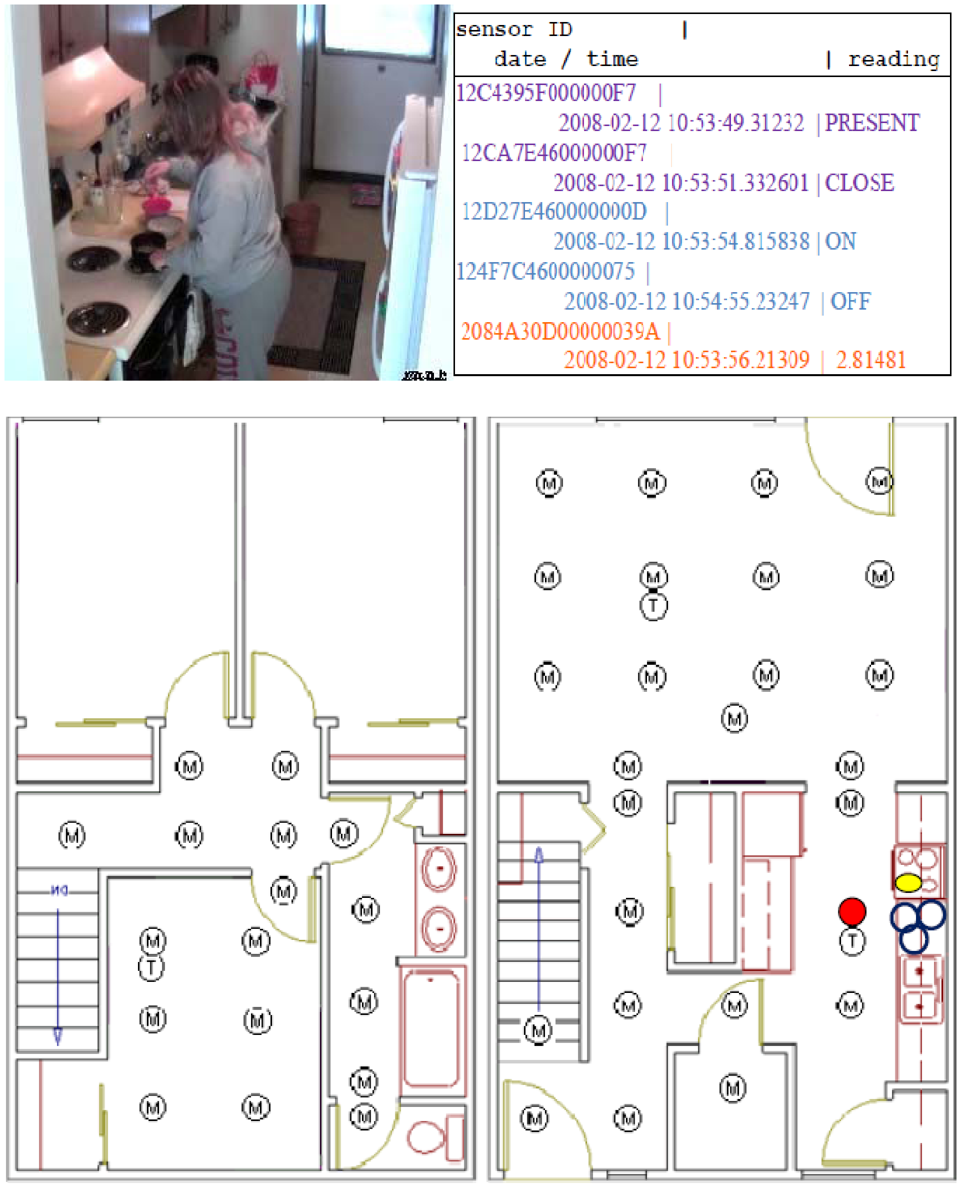


Figure 4 - Recognition of the "cooking" activity in the smart apartment. The first sensor reading indicates that one of the tagged items is being used. The next entry indicates that the cabinet door was just closed. The following two entries reflect that a motion sensor was activated then deactivated, and the last

entry shows a non-zero reading for the stove burner. A visualization of the sensor values that are active at this point during the "cooking" task is shown at the bottom.

3.1 Markov Model

The Markov model is derived from an assumption called the Markov property, which allows this system to be analyzed. The Markov property states that the future evolution of the system is independent of its history, but only depends on the current state and most recent action [46]. The description of the present state fully captures all the information that could influence the future evolution of the process and future states are reached through a probabilistic process instead of a deterministic one. At each step the system may change its state from the current state to another state, or remain in the same state, according to a certain probability distribution.

Given information about the number of tasks to identify and available training data, our algorithm will construct a Markov model for each activity and learn the probabilistic transitions between states. While testing, our algorithm constructs a model of the given sequence of observed sensor events and probabilistically determines which previously constructed model best supports the sequence. The activity corresponding to the most probable model is output as the activity represented by the given set of sensor events. This most probable model can be computed using the formula shown in Equation 1.

$$\operatorname{argmax} p(a \mid e_{1\ldots t}) = p(e_{1\ldots t} \mid a) \cdot p(a) \qquad (1)$$

Here, p(a) is the prior probability of class *a*, or the established likelihood that activity *a* will occur before the arrival of new evidence or information. It is calculated as the ratio of instances for which the class label is class *a*. For our experiment, we have 5 target classes, each representing an activity. The prior probability is uniformly distributed among these 5 activities as each of the activities was performed by every participant, which makes the original belief equal for all activities.

The term $p(e_{1\ldots t} \mid a)$ in Equation 1 represents the probability of observing evidence $e_{1\ldots t}$ for sensor events belonging to class a. It is calculated as the sum of the likelihood of being in a state after processing the sequence of sensor events $e_{1\ldots t}$, summed over all states. The formula in Equation 2 is used to update the likelihood of every state whenever a new sensor event is processed.

$$p(S_{t+1} \mid e_{1\ldots t+1}) = \alpha\, p(e_{t+1} \mid S_{t+1}) \sum_{S_t} p(S_{t+1} \mid S_t) \cdot p(S_t \mid e_t) \qquad (2)$$

Here, $p(S_{t+1} \mid S_t)$ represents the transition probability of moving from the previous state S_t to the current state S_{t+1}, $p(S_t \mid e_t)$ represents the probability of being in the previous state given the sequence of evidence up to event e_t, and α is the normalizing constant.

Using the sequential probability distribution that can be directly computed from the Markov model, we can observe a sequence of sensor events and identify the model (and the task that the model represents) that yields the highest probability of corresponding activity to the observation sequence. Specifically, after each sensor event we will generate a label for the activity (or set of activities) that the participant is performing, and will use a forward probability-propagating algorithm (as mentioned above) to identify the belief state (or current state) of the corresponding activities.

In constructing the model for an activity, we treat every sensor as a state as shown in Figure 5. For example, for a sequence of sensor events [Motion 14, Motion 15, Motion 16, Motion 17, Water On, Water Off], the probability that the model "washing hands" in Figure 5 supports the sequence is calculated as follows –

Step1: First, obtain the prior probability of all states for each activity. In this case, we calculate the prior probability of a state as the ratio of the number of sensor events representing that state in an activity to the total number of sensor events recorded for that activity.

Step2: Obtain the probability of transitioning from a state to every other state for each activity. The transition probability is calculated as the ratio of the number of transitions made from the previous state to the new state, to the total number of sensor events recorded for the previous state for the given activity. For example – The transition probability of moving from sensor M14 to M15 for the activity "washing hands" turns out to be 0.47.

Step 3: Obtain the likelihood of being in every state for each activity. This is calculated as the product of the prior probability of the initial state and the transition probability of moving from the initial state to the final state. This value is then multiplied with the probability of the observing the current evidence for the given activity.

Step 4: The probability that a model supports the given sensor sequence is calculated for each activity (or each model) as the sum of the likelihood values calculated for each state in that activity.

The probability that the "washing hands" model supports the sequence [Motion 14, Motion 15, Motion 16, Motion 17, Water On, Water Off] is 0.1, which is greater than the values from the models for cooking, making a phone call, or cleaning up. Similarly, we can probabilistically determine the belief state, or the most likely state of the model that is currently being observed.

In addition, few approaches are designed to make use of the timing of the activity and steps within the activity. The duration of each sub-task in an activity can be used as additional information to distinguish between overlapping activities or different activities which trigger a similar set of sensors. In order to incorporate this timing information, our models annotate each state description with a normal distribution representing the likely start time and duration of the activity initiation and of each step comprising the activity. We calculate the probability of time matches using the definition of the normal distribution and now this value also contributes to the probability of a model matching an activity in addition to the probability based on the sensor events.

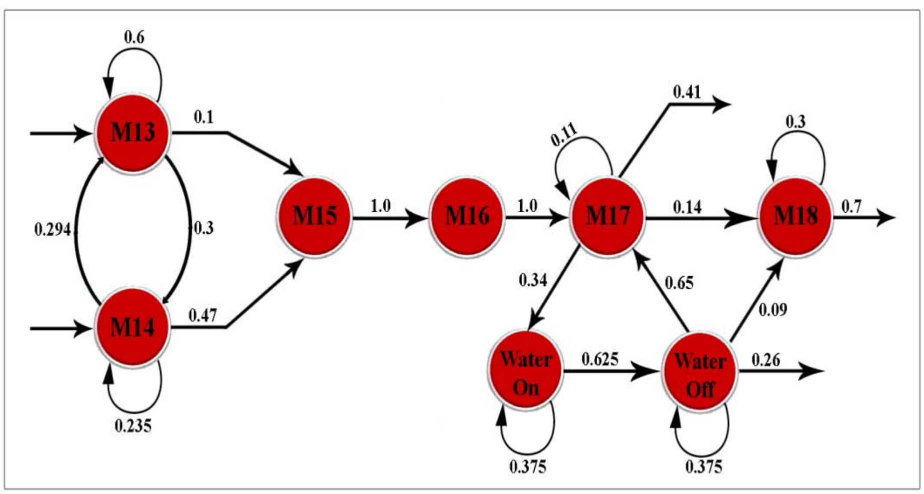


Figure 5 – Markov model representing the “washing hands” activity

We separated the activities into distinct event streams for training and testing. Markov models were generated based on the values provided in the training data for each of the distinct activities and were to automatically label the sensor event streams from the test set. In the results reported below, we show accuracy results generated using three-fold cross validation on the participant data.

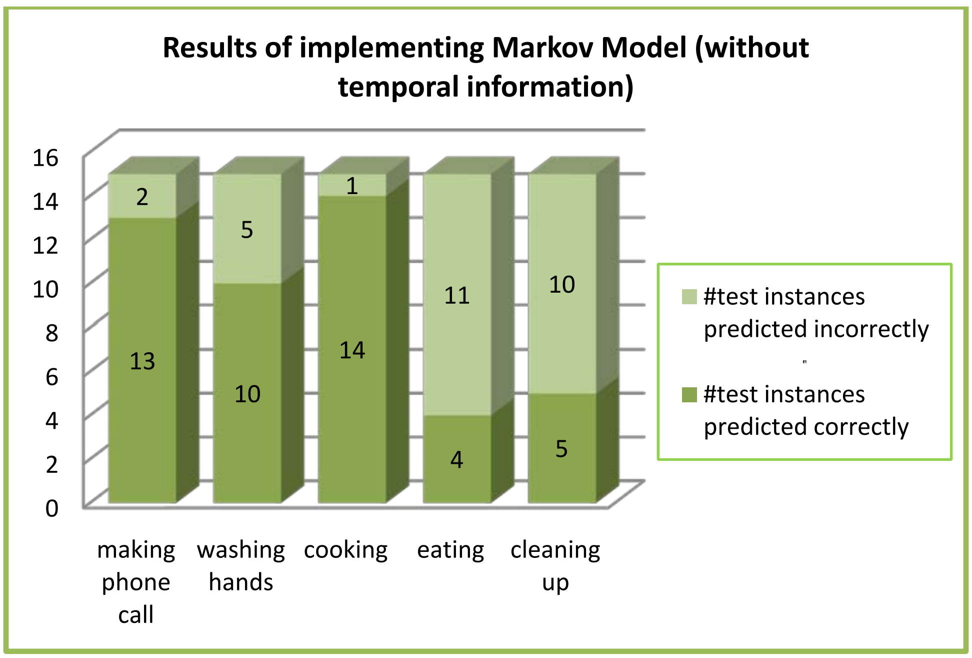


Figure 6 - Bar graph showing results of using Markov model in identifying activities (without using any timing information).

The five activities ("making phone call", "washing hands", "cooking", "eating" and "cleaning up") are represented on the x-axis and the y-axis depicts the accuracy in their prediction. This model shows an average overall accuracy of 61.33% on the test set, with individual accuracies of 86.67%, 66.67%, 93.34%, 26.67% and 33.34% respectively for the five activities. It can be observed that the model shows the lowest accuracy in predicting the "eating" and "cleaning up" activities. The "eating" task specifically is the shortest activity and it does not involve any sensors that belong exclusively to this task, which reduces its recognition accuracy. As the "cleaning up" activity is performed in the same parts of the apartment as the "washing hands" activity, they trigger similar sensors and hence the Markov models generated for these two activities overlap quite a bit. The "washing hands" task is also a shorter activity; the corresponding set of sensor events for this task actually forms a subset of the set of sensor events that comprise the "cleaning up" activity. As a result, the "cleaning up" activity is thus often incorrectly predicted as a "washing hands" activity by the model.

3.2 Markov Model Augmented With Temporal Data

In the second part of our experiment, we augment our models with temporal information by associating with each model state a normal distribution of the time spent in the corresponding sub-task. This helps in distinguishing between overlapping activities like in the above case as evidenced by the results below.

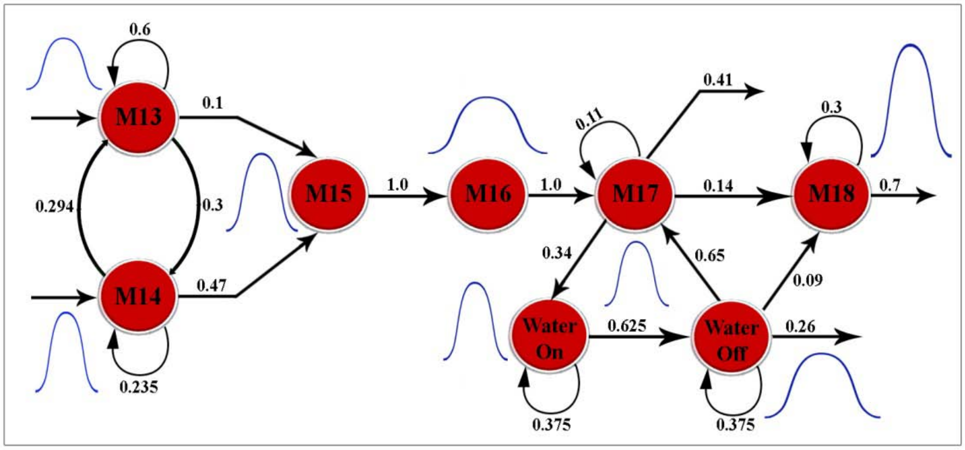


Figure 7 – Markov Model with each state annotated with normal distribution over time.

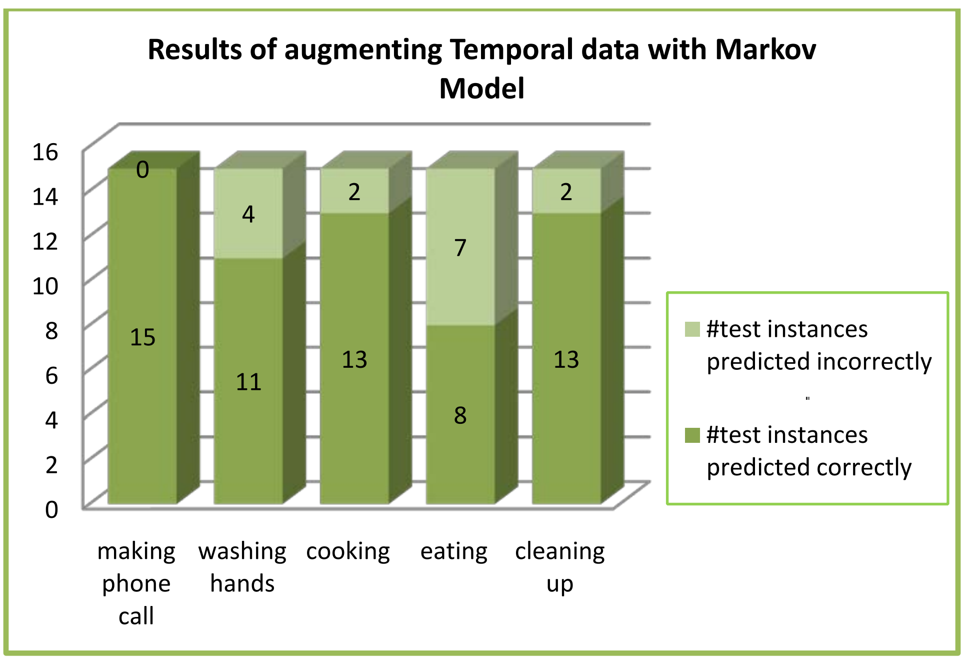


Figure 8 - Bar graph showing results of using Markov model with temporal information.

When augmented with temporal information, the Markov models show an overall accuracy of 80.01%. This represents an average improvement of 18% over the previous model. It can be noticed from the results in Figure 8 that adding temporal information to the Markov models greatly enhanced their accuracy of prediction, particularly for the similar activities that created confusion for the earlier model. The accuracy of predicting the activity "cleaning up" showed up a maximum increase from 33.34% to 86.67%. The bar graph in Figure 9 above brings together the two approaches and shows a contrast between the performance of the Markov models with and without temporal information to facilitate easy comparison. Considering that the two approaches work on the same data set and the experimental setup is almost identical for the two experiments, we performed a paired t-test to find the statistical significance of the difference between the performances of two approaches. The significance for these two algorithms is $p<0.070$ which means that augmenting temporal information with the Markov Model helps in improving the recognition accuracy though not to a significant degree.

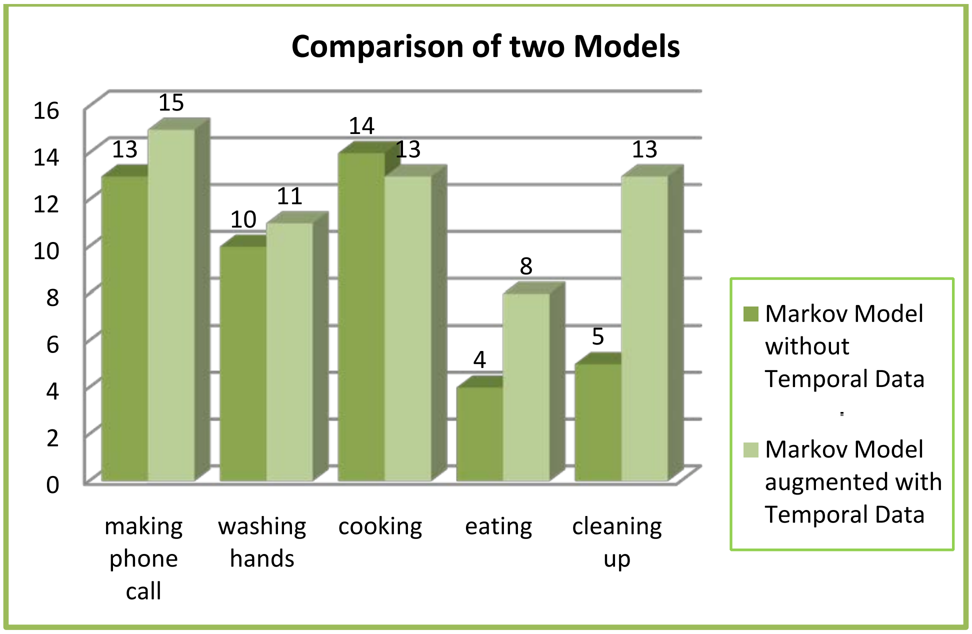


Figure 9 - Bar graph showing comparison models with and without temporal information

As an attempt to further improve the performance of the model, we formulated a number of abstractions which represented the states in our Markov Model. These abstract states were formulated based on location or sub-task of an activity. In our experiment, we selected "Living Room", "DiningRoom Phone", "Dining Room Eat", "Kitchen Water", "Kitchen Burner", "Kitchen Items", "Kitchen" and "Medicine" as the states. Each of these abstractions represents a state in the Markov Model as shown in Figure 10.

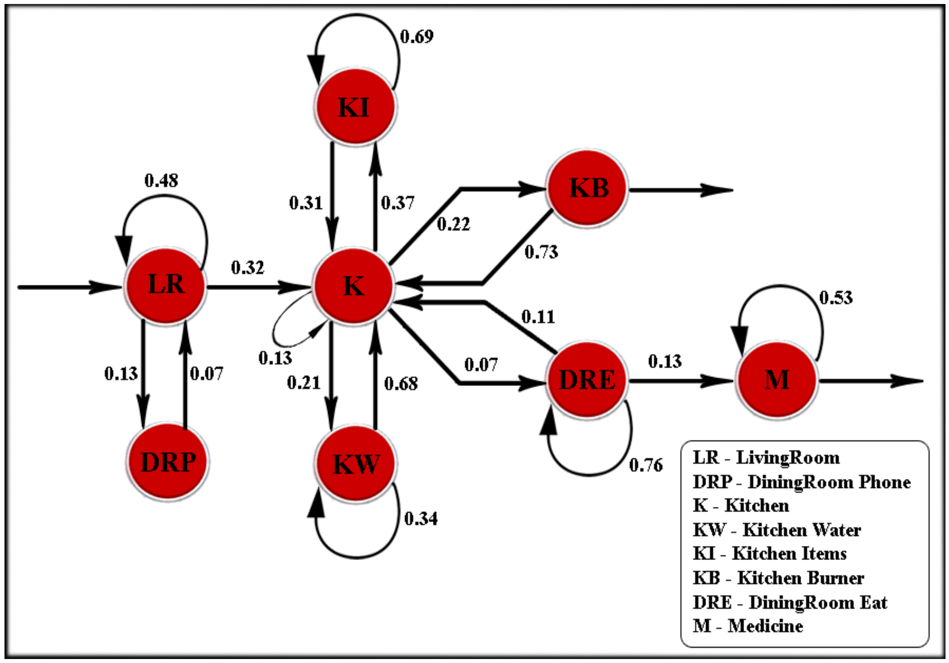


Figure 10 - Markov Model with abstract states for "cooking" activity

Using abstract states instead of individual sensors as states in the Markov model greatly helped in improving the performance of the model. The abstract states concealed the details of each and every sensor from the model and rather collected a set of sensor events to form an abstract state. This reduced the overall number of states in the model for each activity and thus helped the model in learning better and more appropriate probability values. The algorithm now updates the probabilities according to both, the transition probabilities from previous state to the next state, and the probability distribution over specific sensor values for each abstract state. Despite having abstract states, the probability distribution is still associated with the actual sensor values that correspond to these states. The models generated using this technique were more robust against noise and were capable of handling uncertainty in sensor readings. The use of abstract states enhanced the accuracy of the model to 95.00%. The accuracy of predicting the activity “eating” and “cleaning up” showed up a maximum increase from 53.34% to 93.34% and 86.67% to 93.34%. The chart in Figure 11 summarizes the results of this approach.

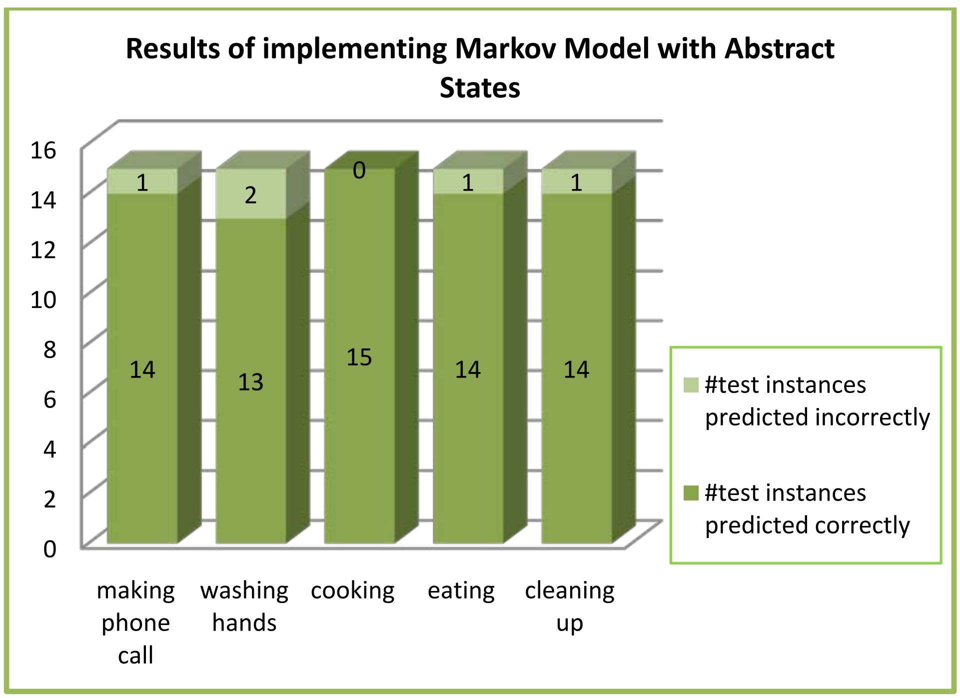


Figure 11 – Bar graph showing results of implementing Markov Model with abstract states.

The implementation of Markov model with abstract states shows an improvement of 34% over the plain Markov model and 15% over the previous model using temporal information with every sensor as a state. The paired t-test conducted to statistically compare the performance of the above approach with the Markov model, resulted in a p-value of $p<0.035$ which implies that the Markov model with abstract states significantly outperforms the Markov model with a confidence of more than 95%.

These techniques describe approaches to recognizing activities that are performed by residents of smart environments. Not only do we demonstrate that these activities can be recognized by sensors in physical environments using Markov models, but we also show that the recognition accuracy is greatly improved through the use of temporal event duration information. As we move on to more complex situations for activity recognition, we investigate techniques for detecting activities when activities are interrupted and interleaved and for recognizing activities when there are multiple residents in the environment.

CHAPTER FIVE

RECOGNIZING INTERLEAVED ACTIVITIES

A system that can detect and track a large number of day-to-day human activities is of both conceptual and practical interest. But tracking daily activities brings challenges along with opportunities. The main challenges are that the number of activities to be detected is very large and activities are often performed in not only isolated (i.e. sequential), but also complex (i.e. interleaved and concurrent) manners in real life.

Little work has been done in addressing the problem of activity recognition in such complex situations. Researchers have tried to build systems that attach sensors to devices in the environment and to the human body in combination with sensors that observe the scene using audio, visual, and magnetic sensors [19]. These sensors help in more accurately determining which object the inhabitants are using, their motion and interaction with environment at any given time and can be more powerful in recognizing human activities. But at the same time, use of such wearable sensors and video cameras in smart space is obtrusive and these are not desirable to older adults who are the potential consumers of such technology.

In our research, we focus on performing activity recognition that is not only accurate, but that also requires a minimum number of sensor devices as it can be cumbersome for the resident to wear many such sensors and battery packs mounted over the body. Our approach is fundamentally different from other approaches in its use of probabilistic models and dependence on mainly the motion sensors in addition to few item and door sensors. The problem we address in this section is to recognize activities when they are performed in an interleaved fashion. For this study, we selected 8 ADLs important from the perspective of fundamental functioning and instrumental ADLs which enable individuals to live independently in a community. These activities are as follows –

1. *Filling medication dispenser:* Here the participant removes the items from kitchen cupboard and fills the medication dispenser using the space on the kitchen counter.

2. *Watching DVD:* The participant selects the DVD labeled "Good Morning America" located on the shelf below the TV and watches it on the TV. After watching it, the participant turns off the TV and returns the DVD to the shelf.

3. *Watering plants:* For this activity, the participant takes the watering can from the supply closet and lightly waters the 3 apartment plants, 2 of

which are located on the kitchen windowsill and the third is located on the living room table. After finishing, he/she empties any extra water from the watering can into the sink and returns the watering can to the supply closet.

4. *Conversing on Phone:* Here the participant answers the phone when it rings and hangs up after finishing the conversation.

5. *Writing Birthday Card:* The participant writes a birthday wish inside the birthday card and a check in a suitable amount for a birthday gift, using the supplies located on the dining room table. He/she then places the card and the check in an envelope and appropriately addresses the envelope.

6. *Preparing meal:* The participant uses the supplies located in the kitchen cupboard to prepare a cup of noodle soup according to the directions on the cup of noodle soup. He/she also fills a glass with water using the pitcher of water located on the top shelf of the refrigerator.

7. *Sweeping and dusting:* For this task, the participant sweeps the kitchen floor and dusts the dining and the living room using the supplies located in the kitchen closet.

8. *Selecting an outfit:* Lastly, the participant selects an outfit from the clothes closet to be worn by a male friend going on an important job interview. He/she then lays out the selected clothes on the living room couch.

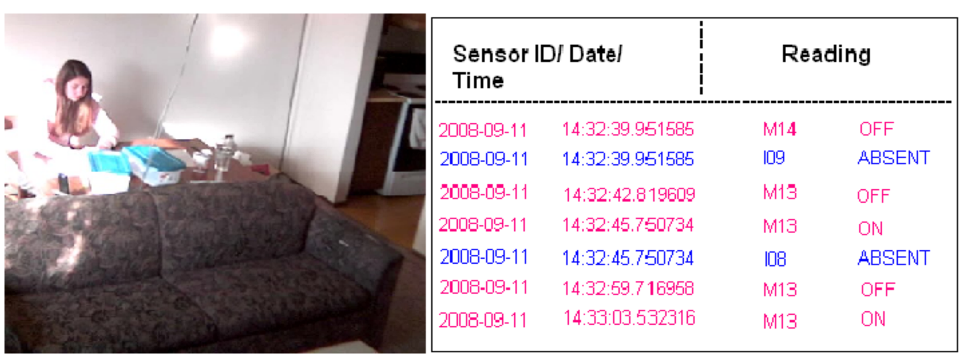
One challenge in using machine learning-based techniques to recognize interleaved activities is that they require the training dataset to contain instances of the interleaved activity to be predicted. However, there is a large number of ways in which daily activities can be interleaved, interrupted, and performed in parallel, and the ways they are interleaved may differ from person to person. This implies that the training dataset for learning such complex activity models has to be very large and must represent all possible ways of interleaving tasks so as to be able to correctly identify every activity.

Experimental Setup

To address this issue, we recruited 20 participants to perform the 8 activities mentioned above. First, the participants performed each activity in isolation. In other words, each of the participants first performed these tasks one at a time in sequential order. The participants were then instructed to perform all of these activities by interweaving them in any fashion as they like with a goal of being efficient in performing the tasks. From the initial run of performing the activities in sequential order, we obtained a set of sensor events for every activity which could be used to generate a model of every individual task. In the second run of performing the tasks in an interwoven manner, the order in which different activities were performed and interleaved was left to the discretion of the participant. As different participants interweaved the tasks differently, the data set thus obtained was significantly richer.

The data collected for both of these runs was then manually labeled. Specifically, each sensor event was labeled with the corresponding activity id. The average times taken by the participants to perform these activities were 3 and half minutes, 7 minutes, 1 and half minutes, 2 minutes, 4 minutes, 5 and half minutes, 4 minutes and 1 and half minutes, respectively. The average number of sensor events in the data sets for the activities are 31, 59, 71, 31, 56, 96, 118, and 34 sensor events respectively.

The script that was given to the participants to perform the interwoven tasks can be found in Appendix B.



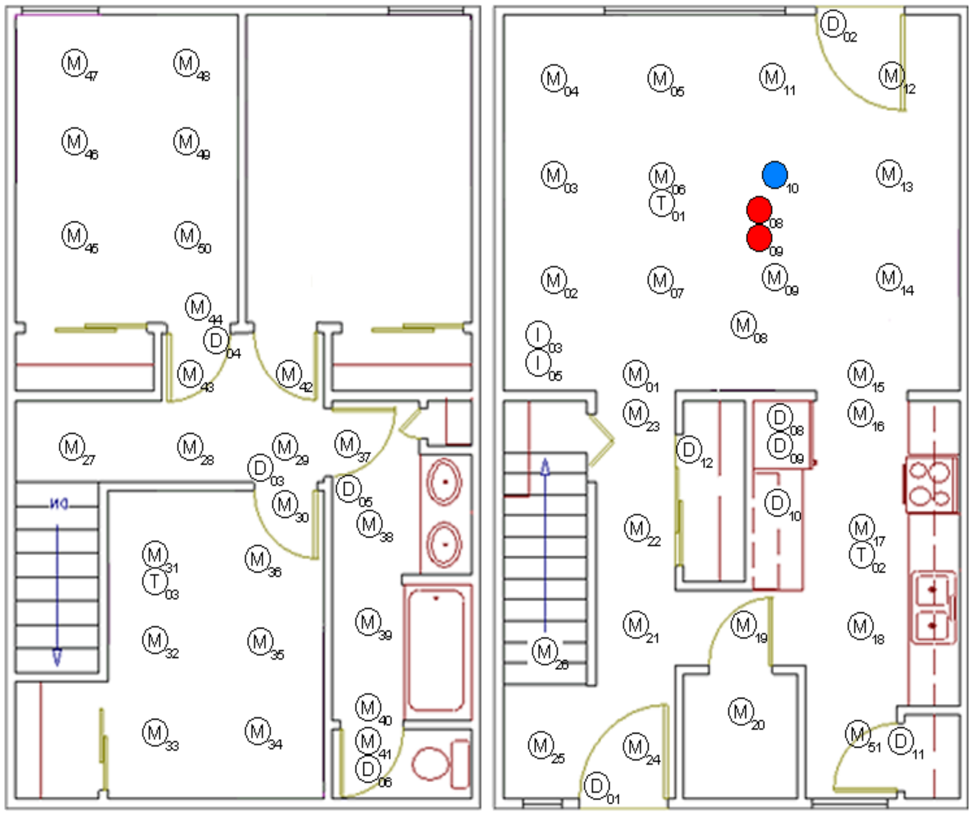


Figure 12 - “Writing birthday card” activity in the smart apartment. The web cam image in the upper left shows a student participant performing the task. The activity results in the sensor readings shown in the upper right. These readings correspond to motion sensors and item sensors (on birthday card and writing material) being triggered. A visualization of the sensor activity for the “Writing birthday card” task is shown at the bottom.

Figure 13 – The participant is interweaving 2 tasks. Here she transitions from midway through the "Selecting an outfit" activity to the 'Conversing on phone" activity.

Our aim is to label every sensor event with an activity label so that the smart environment agent is aware of the current activity at any given time and can use that information for monitoring and assisting individuals with special needs. We tried several probabilistic classifiers to find out an approach that works best for identifying interleaved activities. These probabilistic models include naïve Bayes, hidden Markov hodel, HMM with a time window, frequency-based HMM with a sliding window and frequency-based HMM with a shifting window. Some of these classifiers like naïve Bayes classifier first pre-process the sensor readings into feature counts, and then applies the classifier to label the activities. In contrast, other classifiers like the Markov model and the hidden Markov model classifiers make use of the associated temporal information to probabilistically infer the activity labels given the observations in the form of sensor events.

4.1 Naïve Bayes

A Naive Bayes classifier is a probabilistic classifier based on Bayes' theorem [47]. It works on an assumption that the effect of a variable value on a given class is independent of the values of other variables. This assumption is called class conditional independence and is made to simplify the computation. The essence of the Bayesian approach is to provide a mathematical rule explaining how the existing beliefs should be changed in the light of new evidence. In other words, it allows scientists to combine new data with their existing knowledge or expertise. Mathematically, rule is given in Equation 3.

$$P(A=a \mid e) = \frac{P(e \mid A=a)\, P(A=a)}{P(e)} \tag{3}$$

In other words,

$$\text{posterior prob} = \frac{\text{likelihood} * \text{prior prob}}{\text{marginal likelihood}}$$

where $P(A=a|e)$ denotes the probability that random variable A has value a given evidence e. The denominator is just a normalizing constant that ensures the posterior values add up to 1; it can be computed by summing up the numerator over all possible values of A.

In our implementation of the Naïve Bayes model, we treat every sensor as a feature descriptor and learn probability distributions over feature values for all activities. Events such as tripping a motion sensor, item sensor, door sensor or phone sensor denote the evidence. The frequencies of these events as they occur during each activity govern the likelihood of that activity given the evidence. We use the maximum a posteriori rule to pick the most probable activity when using the Naïve Bayes classifier. The classifier returns the class value given by the formula in Equation 4.

$$\mathrm{argmax}_{a \in A} \ (A=a \mid e) = \frac{P(e \mid A=a)\, P(A=a)}{P(e)} \qquad (4)$$

As the denominator is a constant, the formula can be reduced to the one shown in Equation 5.

$$\mathrm{argmax}_{a \in A} \ (A=a \mid e) = P(e \mid A=a)\, P(A=a) \qquad (5)$$

In spite of its simple design and simplifying assumptions, Naive Bayes classifiers often work much better in many complex real-world situations than expected [48, 49]. The naïve Bayes independence assumption also works well in our case as high probability values are generally concentrated over different states (i.e., sensors) for different activities.

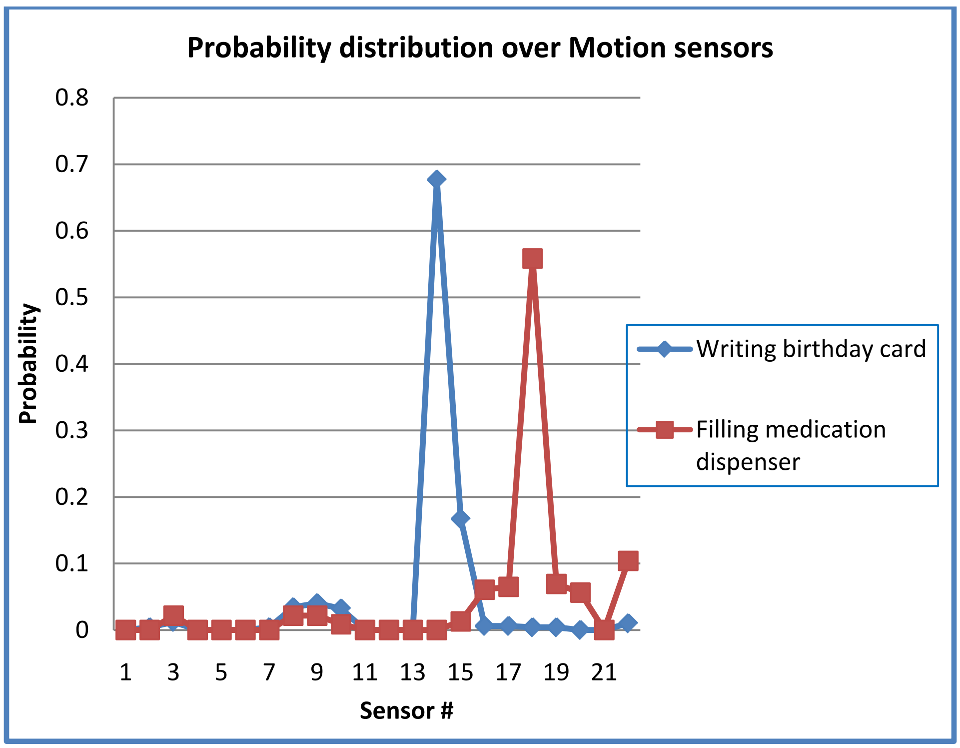


Figure 14 – Concentration of probability values for activities "Writing birthday card" and "Filling medication dispenser".

We use the data collected from sequential execution of the activities to train the naïve Bayes model and calculate the probability values. The interweave data is then used for testing the performance of the model by using the pre-calculated probability values associated with the states to compute the likelihood of each activity. The activity with the maximum likelihood is output as the activity label associated with the current sensor event.

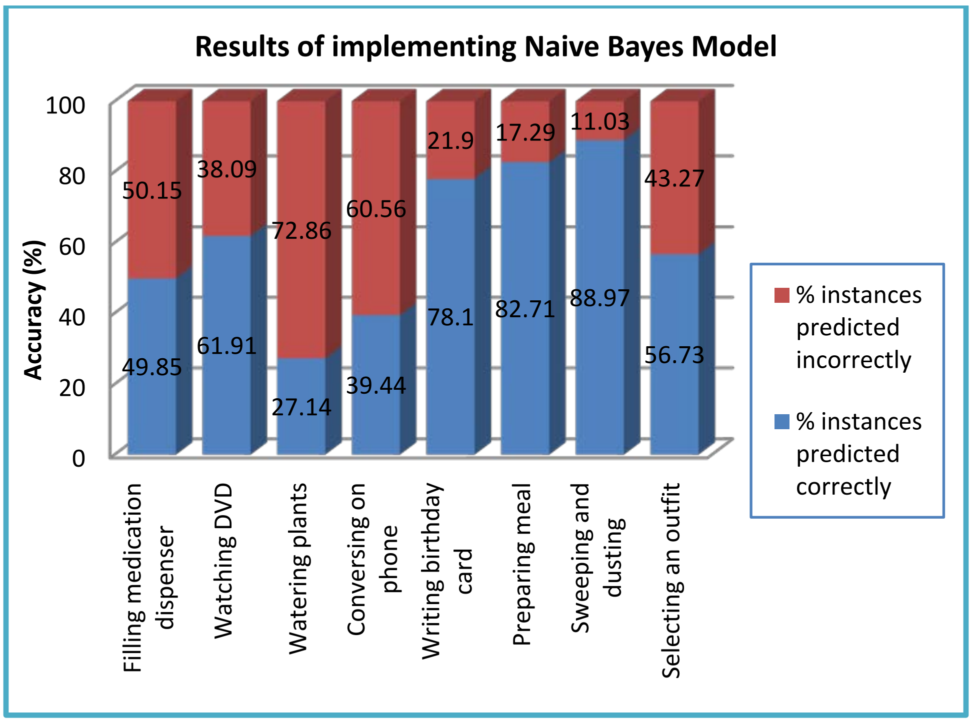


Figure 15 – Bar graph showing performance of naive Bayes Model broken down by activity.

The model results in an overall accuracy of 66.08% in predicting activities. The accuracy values for the 8 activities are 49.85%, 61.91%, 27.14%, 39.44%, 78.10%, 82.71%, 88.97% and 56.73%, respectively.

The model shows a relatively poorer performance in identifying the two activities "Watering plants" and "Conversing on phone". The activity "Watering plants" does not have any particular set of sensors associated with it. For this task, the resident moves around in the apartment, watering 3 plants placed at 3 different locations which results in logging many motion sensor events placed throughout the apartment. Due to this nature of the activity, the sensor log for this task lacks the dominance of any particular set of sensors. Similarly, for the "Conversing on phone" task, only the phone sensor is tripped whenever the resident receives a call on the phone. However, different people sit and talk over phone at different locations.

In addition, when they interwove this task with other activities, different participants chose to do it very differently. For example, one participant chose to talk over the phone while cooking while another was conversing on phone while writing the birthday card. This resulted in a lack of any consistent pattern for this activity. As the naïve Bayes model considers various attributes to be independent and does not take transition probabilities into consideration, such activities could not be identified very well by the model. Adding timing information and transition probabilities to the naïve Bayes model might be helpful in improving the performance of this model.

In order to make use of the information resulting in moving from one state to another, we next implemented a hidden Markov model.

4.2 Hidden Markov Model

A hidden Markov model (HMM) is a statistical model in which the underlying model is a stochastic process that is *not* observable (i.e. hidden) and is assumed to be a Markov process which can be observed through another set of stochastic processes that produce the sequence of observed symbols. A HMM assigns probability values over a potentially infinite number of sequences. But as the sum of the probabilities must be one, the distribution described by the HMM is constrained. This means that the increase in probability values of one sequence is directly related to the decrease in probability values for another sequence.

In case of a regular Markov model, all states are observable states and are directly visible to the observer. Thus, the only other parameter in addition to the prior probabilities of the states is the state transition probabilities. In the case of a hidden Markov model, there are hidden states which are not directly visible, and the observable states (or the variables) influence the hidden states. Each state is associated with a probability distribution over the possible output tokens. Transitions from any one state to another are governed by a set of probabilities called *transition probabilities*. Thus, in a particular state an outcome can be generated according to the associated probability distribution.

HMMs are known to perform very well in cases where temporal patterns need to be recognized which aligns with our requirement in recognizing interleaved activities. Figure 16 represents the general architecture of a HMM where each circle represents a random variable. The random variable $x(t)$ is the hidden state and the random variable $y(t)$ is the observable state at time t. The arrows are used to denote conditional dependencies.

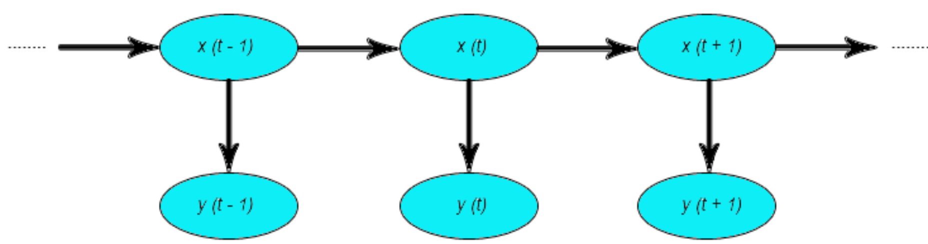


Figure 16 – General architecture of a Hidden Markov Model

The conditional probability distribution of any hidden state $x(t)$ at time depends only on the value of its preceding hidden state i.e. $x(t - 1)$ i.e. the values at any time before time $t - 1$ have no influence on the value of state at time t which essentially is the Markov property. Also, the value of the observable state $y(t)$ depends only on the value of the hidden state $x(t)$ given at time t.

From the training data set, we know the sensors that are used for each of the activities, and hence we treat them as the parameters of our model. Using this sequence of observations, our aim is to find the most likely sequence of hidden states that could have generated the given output sequence. We used the Viterbi algorithm [50] to solve this problem.

In our implementation of the hidden Markov model, we treat every activity as a hidden state. As a result, our HMM includes 8 hidden states, each of which denotes one of the 8 modeled activities. Next, every sensor is treated as an observable state in the model due to the fact that every sensor which is used is observable in our dataset. The challenge here is to identify the sequence of activities (i.e., the sequence of visited hidden states) that corresponds to a sequence of sensor events (i.e. the observable states). For this, we calculate based on the collected data the *prior probability* (or the start probability) of every state which represents the belief about which state the HMM is in when the first sensor event is seen. For a state (or an activity) *a*, this is calculated as the ratio of instances for which the activity label is *a*. We also calculate the *transition probability* which represents the change of the state in the underlying Markov model. For any two states *a* and *b,* the probability of transitioning from state *a* to state *b* is calculated as the ratio of instances having activity label *a* followed activity label *b,* to the total number of instances. The transition probability signifies the likelihood of transitioning from a given state to any other state in the model and captures the temporal relationship between the states. And lastly, the *emission probability* represents the likelihood of observing a particular sensor event for a given activity. This is calculated by finding the frequency of every sensor event as observed for each activity.

The likelihood of an activity *a* is calculated according to the formula shown in Equation 6.

$$\text{likelihood(a)} = \sum_{b \in A} [\ \text{PriorP(b)} \ . \ \text{TransitionP(b, a)} \ . \ \text{EmissionP(a, e)}\ ] \qquad (6)$$

where PriorP(b) is the prior probability of activity *b*, TransitionP(b, a) is the probability of transitioning from activity *b* to activity *a*, and EmissionP(a, e) is the emission probability of evidence *e* being obeserved for the activity *a*; summed over all activities. The most likely activity label for the current sensor event given the history of sensor events seen before is calculated by finding the activity with highest likelihood by using the formula in Equation 7.

$$\text{argmax [likelihood(a)]; where a} \in \text{A} \qquad (7)$$

The prior probability of every activity is updated to the new likelihood value calculated for that activity (as above) whenever a new sensor event is processed.

Unlike the flat Markov model approach where one model was generated for each activity, here only one collective HMM model is generated for all activities. Given new sensor data, the flat Markov model would match the new data against every model and return as the activity label the label of the model that most closely aligned with the sensor data. In the case of the HMM, the new data is run through the HMM as a continuous stream of data and the activity (hidden node) with highest probability value is returned as the activity label. Figure 17 shows a section of the HMM for interleaved activities.

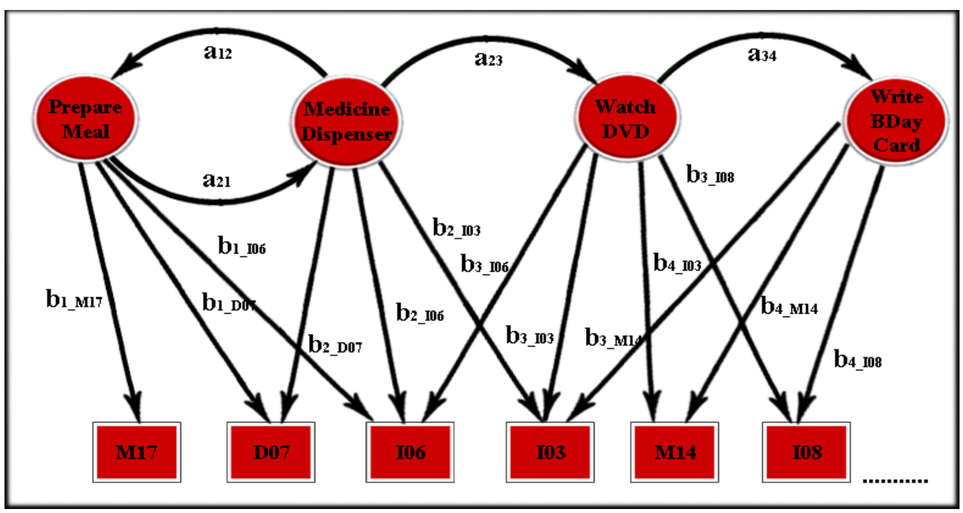


Figure 17 – A section of Hidden Markov Model for interleaved activity data. The circles refer to the activities i.e. the hidden states and the rectangles refer to the observable states. The 'a_{ij}' values refer to the transition probability of

transitioning from activity *i* to activity *j*, and the $b_{i\_sensorId}$ values represent the emission probability of the possible observations.

We trained our model on the interleaved activity data itself in order to learn transitions that residents made from one activity to another. Using 3-fold cross validation to evaluate the performance of the HMM, the resulting average accuracy was 71.01%. Using a HMM results in an overall accuracy increase of 5% over the Naïve Bayes model. The chart in Figure 18 shows the accuracy values for different activities as identified by the HMM.

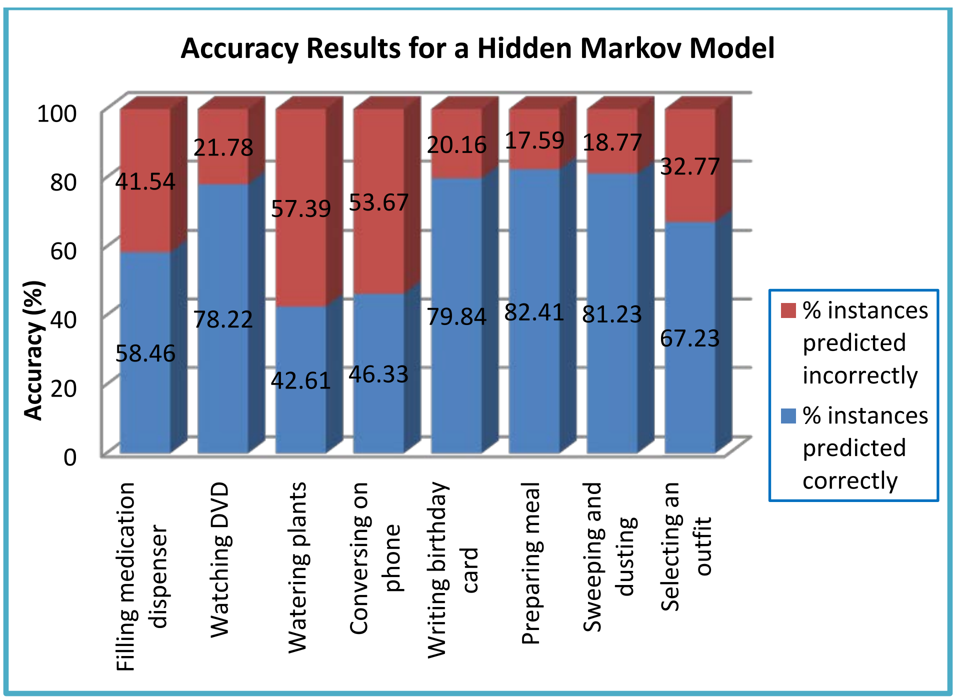


Figure 18 – Bar graph showing performance of the hidden Markov model in recognizing interleaved activities broken down by activity.

The model shows a maximum increase of 16% each for the activities "Watching DVD" and "Watering plants", and 10% increase in the accuracy for the activity "Selecting an outfit". The activities "Watering plants" and "Conversing on phone" still perform poor as compared to other activities but their performance was enhanced by 16% and 9% respectively by using the Hidden Markov Model. We performed paired t-test to compare the performance of naïve Bayes model and the hidden Markov model. The t-value obtained from the test was $p<0.035$. This implies that the hidden Markov model outperformed the naïve Bayes model with a confidence of more than 95%.

A limitation of this model is that it makes very slow transitions from one activity to another. Consider the case when the system is currently in some state indicating a_1 as the most likely activity but the next sensor event belongs to some other activity a_2. In such a scenario, it takes several sensor events for the system to slowly decrease the probability of activity a_1 and increase the probability of activity a_2 to make a transition from a_1 to a_2. Our data of interleaved tasks is very rich in such cases, where the residents were constantly switching between tasks in order to interweave the tasks and be efficient at the same time. This resulted in degradation in performance of the system.

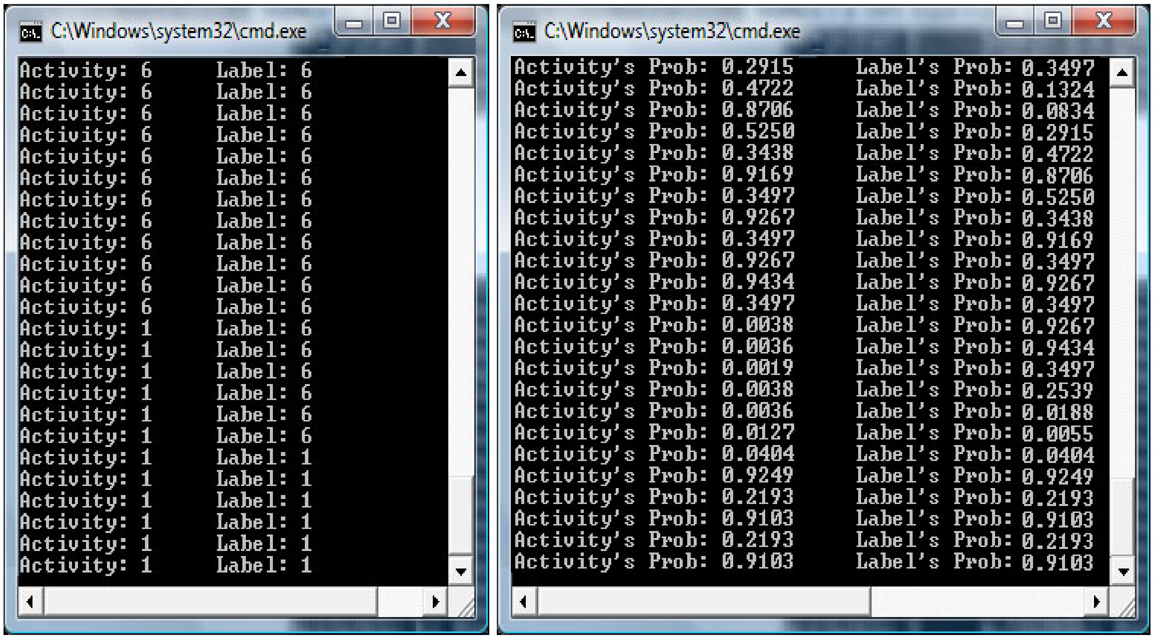


Figure 19 – Screen shot of the execution of the HMM showing a delayed transition from activity 6 to activity 1(left); the probability values of the two activities (right) demonstrating the gradual decrease in the probability of the previous activity and a gradual increase in the probability of the current activity. The model incorrectly labels some of the next activity's initial sensor events before making a transition.

As a remedy, we implemented a HMM with a sliding window. This system uses a sliding window over the data and remembers only those events which belong to the window. Using a sliding window, the probability values are calculated based only on the sensor events contained in the window and the earlier sensor events are flushed out before starting the calculation for a new window. This helps in keeping the probability values low for all activities, thereby helping in making a faster transition from current to the next activity. The next section gives a detailed description of this technique and results of this implementation.

4.3 HMM with Number Based Sliding Window

This technique uses a number based sliding window over HMM to limit the history of sensor events that the model remembers at any given time. Any probability values calculated previously are flushed out whenever the model starts processing a new window. The window slides by one sensor event every time (so as to label every sensor record) and uses only those senor events from the past which fall within the window to recalculate the probability values. The challenge that this model faces is to determine an appropriate window size. The window size can be based on either of 2 factors:

1. Time-based window – the size of the window is dependent upon time. Using this definition, a window contains all of the sensor events that occur within some time frame.

2. Sensor events count-based window – the size of the window is governed by the number of sensor events. Using this definition, a window contains

a fixed number of sensor events irrespective of the span of time that elapsed from beginning to end of the sensor event sequence.

We tried both types of windows and determined experimentally that a window based on the number of sensor events is more consistent and works better than a time-based window. There are activities in which a resident makes very few movements and so there is larger time gap between two sensor events due to which the overall number of sensor events logged for such tasks is very small. For example, in the case of the "Conversing on phone" task, only one sensor event is recorded when the resident starts the phone conversation and one event is recorded at the end of the conversation, although there can be a few more events recorded if the resident is moving while talking. For such activities, a time-based window shows very poor performance as the whole activity shrinks into very few windows. On the other hand, a number based window is consistent in terms of the number of sensor events it processes and hence performs better and more uniformly for all activities.

The second challenge in using a sliding window is determining an appropriate window size. In order to automate window size selection, we divide our data into 2 parts. The first 2/3 of the data is used to find a window size that performs well, by training and testing the model for all possible window sizes on this 2/3 data using 3-fold cross validation. The window size that yields the best average performance over the 3 passes on the sample data is selected. Next, training and testing is run again on the entire data with this window size to label every sensor event. The chart in Figure 20 shows the performance of the HMM with time-based window for all window sizes.

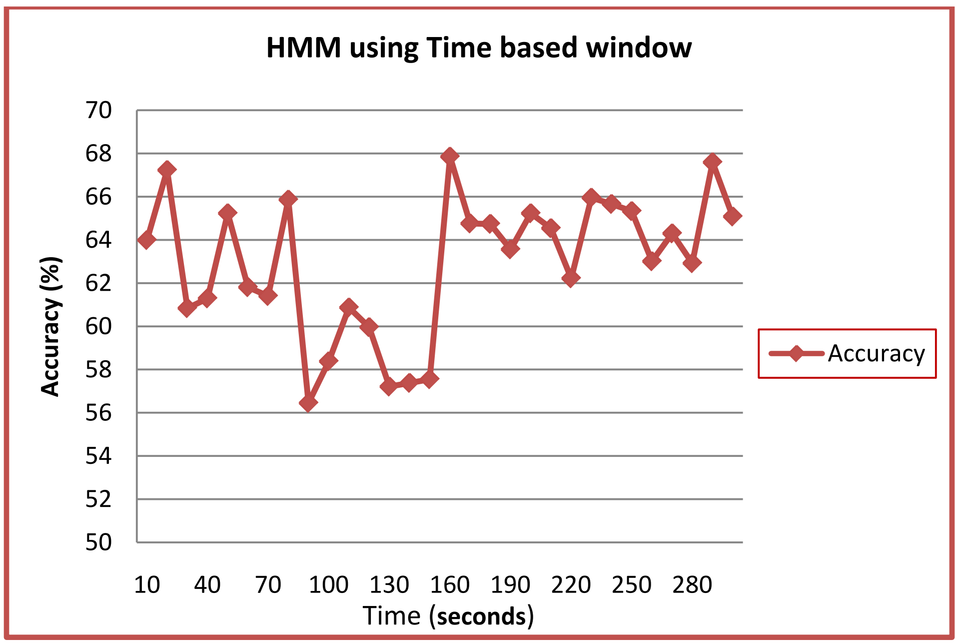


Figure 20 – Performance of the model for all window sizes.

The model shows relatively better performance for higher window sizes. The best performance is observed for a window that spans 160 seconds. The results indicate that the performance drops down for window sizes between 2 to 3 minutes. This implies that a time window of size 2 to 3 minutes will not provide appropriate information for our probability calculations on this particular dataset. The accuracy rises again when the window size is increased to 3 minutes and higher. Because the best window size may vary from one dataset to another, the window size selection step should be performed for each new dataset to obtain best results.

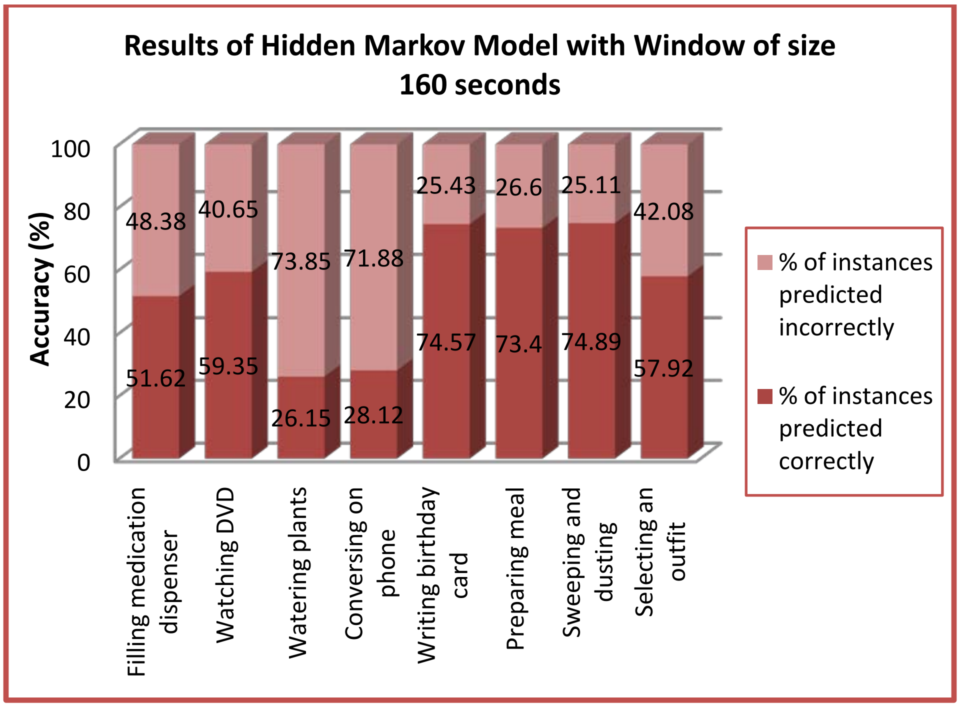


Figure 21 – Accuracy values for all activities as predicted by the HMM for time based window of 160 seconds.

Using the automatically-selected window size of 160 seconds, the resulting average accuracy in recognizing activities turns out to be 56.20%. The bar graph in Figure 21 shows the accuracy values broken down by activity. The "Sweeping and dusting" activity is identified most accurately, yielding an accuracy of 74.89%, whereas the "Watering plants" and "Conversing on phone" activities show minimum accuracies of 26.15% and 28.12%, respectively.

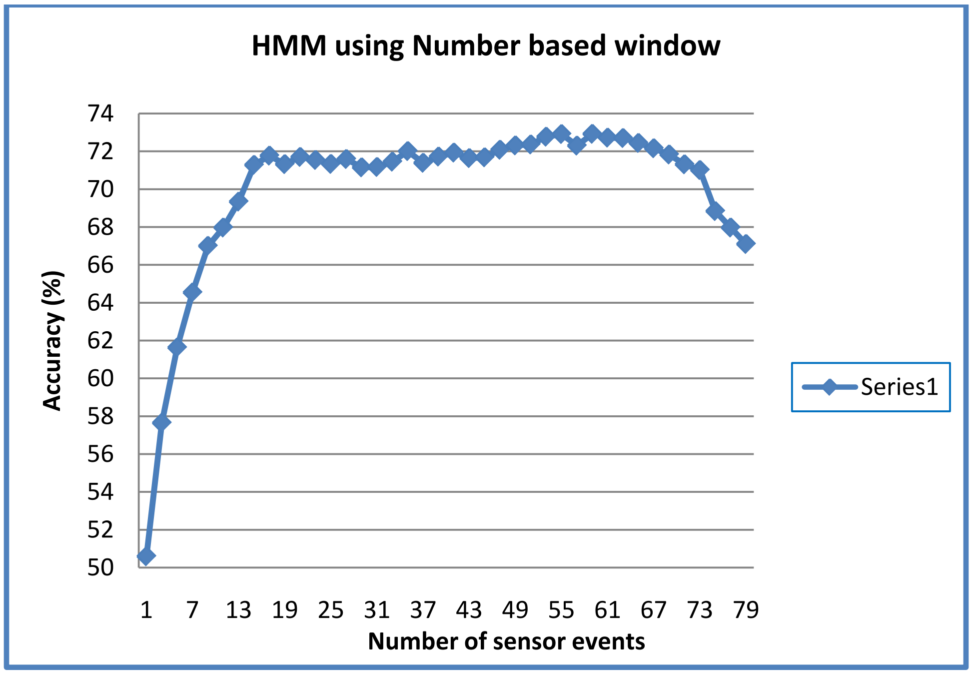


Figure 22 – Line graph showing accuracy of the model for all window sizes based on numbers.

The graph in Figure 22 shows the accuracy of the HMM for all count-based window sizes. The performance increases as the window size increases and reaches the maximum for a window size of 57 sensor events. Performance starts falling again when the window size is increased further. The model gives an overall average accuracy of 63.55% in recognizing activities, which is a more than 6% increase as compared to the time-based window model. The activity "Sweeping and dusting" is predicted with the highest accuracy of 82.88% and the activities "Watering plants" and "Conversing on phone" show lowest accuracies of 30.69% and 43.76%, respectively.

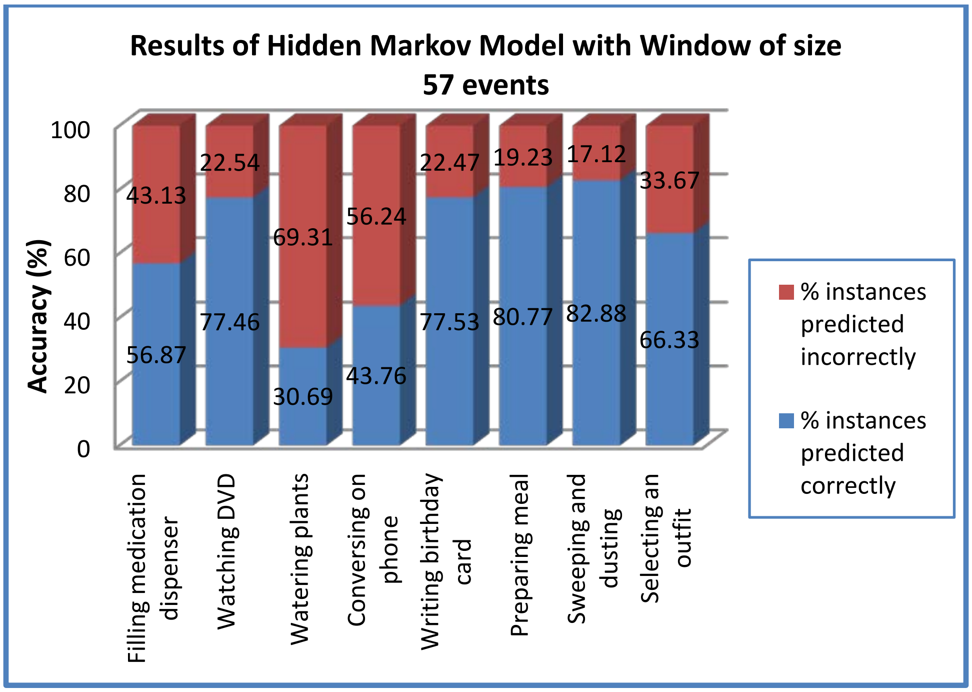


Figure 23 – Accuracy of the model for a window size of 57 sensor events.

Overall, the count-based window model outperforms the time-based window model. The activities with the lowest performance, "Watering plants" and "Conversing on phone", show an increase in accuracy by 4% and 6%, respectively, when the count-based window model is used. The activity "Watching DVD" shows the highest increase in accuracy by 24%. Other activities that perform better with the number based model are "Selecting an outfit" and "Sweeping and dusting" which show an increase in performance by 10% and 12% respectively. Figure 24 shows the comparison between performances of the two models.

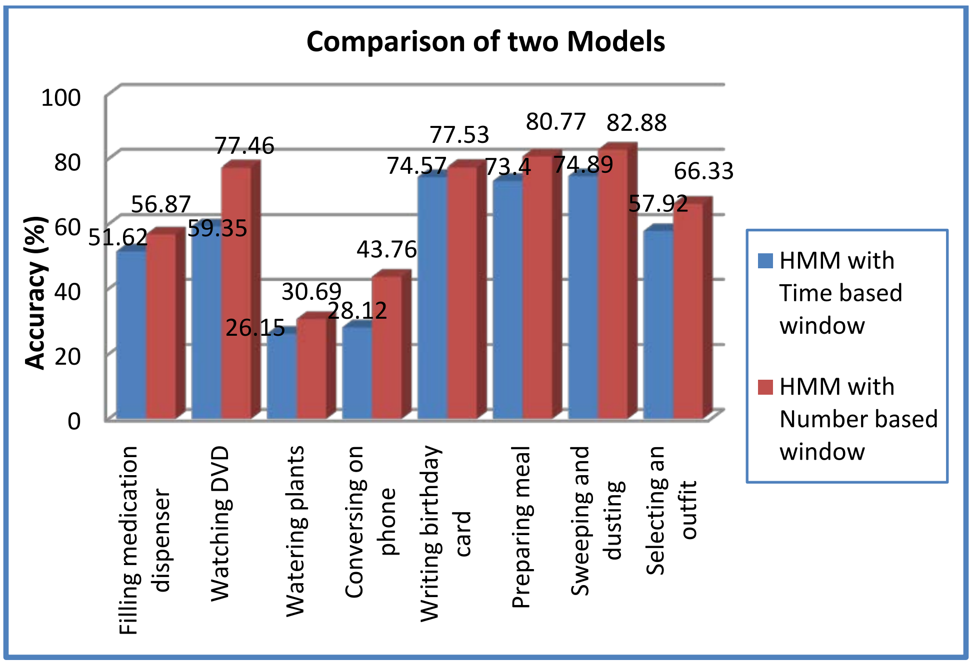


Figure 24 – Comparison between the two models. HMM with a number based window model outperforms the HMM with a time based window model for almost every activity.

We also performed a paired t-test to find the statistical significance of the difference between the performances of two approaches. The p-value for these two algorithms came out to be $p<0.013$ which implies that the HMM with a number-based window model outperforms the HMM with a time-based window with a confidence of more than 95%.

4.4 HMM with Frequency

Another approach that we tried to solve this problem was to use the HMM with a count-based window (as the count-based window seemed more successful). Instead of labeling each sensor event by the most probable activity label, however, this time we label the sensor event with the activity label of the most frequent activity in the window. The window is sliding window and it slides down to the next sensor event once the previous event has been labeled.

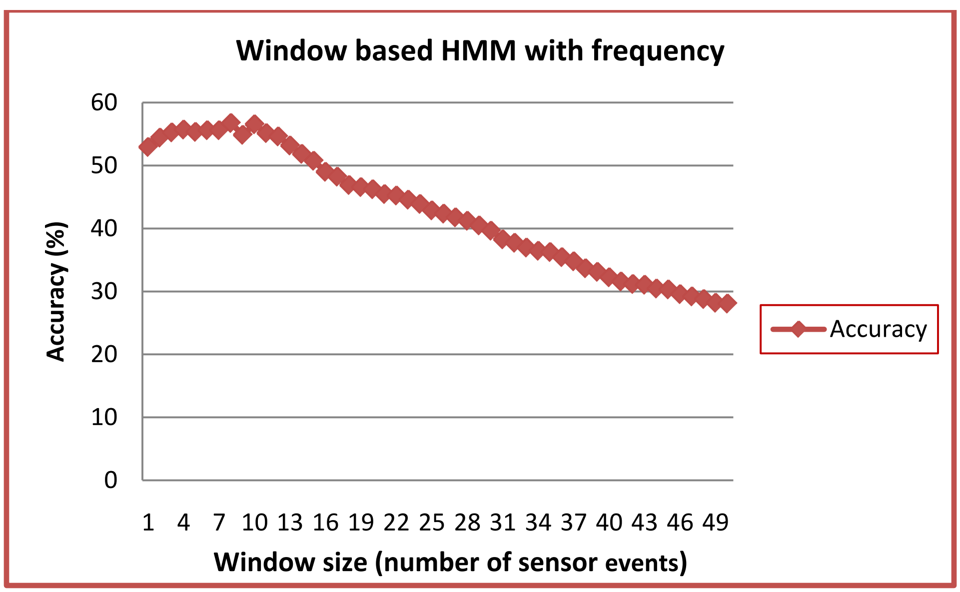


Figure 25 – Performance of a HMM using number based window where next sensor event is labeled by most frequent activity in the window.

This model was trained and tested on the interweave data using 3-fold cross-validation. The technique shows better results for window sizes up to 14 sensor events, after which the accuracy of the model starts degrading. A maximum overall accuracy of 56.75% in identifying activities is shown for window size of 8 sensor events. As the window size increases, sensor events with labels for different activities fall in the same window which results in many frequent activity labels in the window. In many cases, the previous activity labels dominate the new activity label when the window is larger, thereby degrading the performance of the model. The bar graph in Figure 26 shows the performance of the technique broken down by activity.

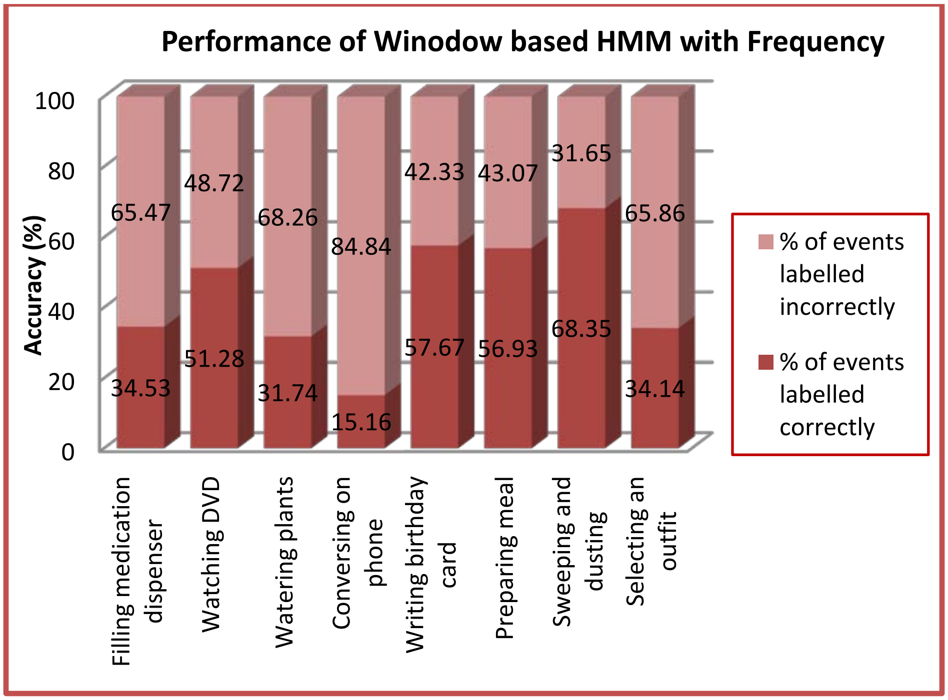


Figure 26 – Bar graph showing performance of window based HMM using frequency, broken down by activity

The technique works in two phases: in the first phase, an appropriate window size is calculated by training and testing the model on 2/3 of the total data. Training and testing is performed again on the entire data in the second phase using the pre-determined window size and performance is estimated using 3-fold cross validation method. The best window size selected by this technique is 8 sensor events long. For this window size, the activity "Sweeping and dusting" shows maximum accuracy of 68.35%, whereas the activity "Conversing on phone" shows the least accuracy of

15.16%. The "Conversing on phone" activity has only the phone sensor associated with it and thus it is hard to be selected as the most frequent activity which makes it perform very poorly.

This model did not perform very well for the interleaved activity dataset. As this model is based on selecting the most frequent activity from the window, it becomes all the more important to find a window size that works for all activities when interweaved together which is very hard to determine. We tried another approach which uses a shifting window, as described in the following section.

4.5 HMM with a Shifting Window

This approach is similar to the one described in the previous section. The difference is that in this technique, we label the whole window as only one activity and shift the window by window size. This approach differs from all previous approaches in the sense that we label every window with an activity label instead of labeling every sensor event. Results of implementing this model show that the time-based window performs better in this case. The model gives a maximum accuracy of 84.18% for a time window of 190 seconds.

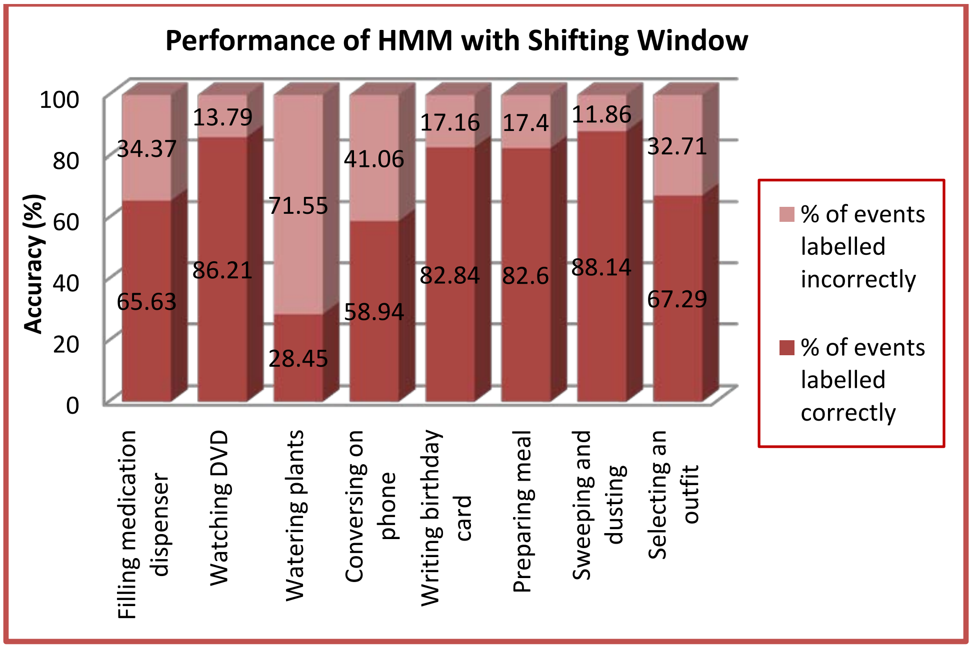


Figure 27 – Bar graph showing accuracy of HMM with a shifting window broken down by activity.

As shown in Figure 27, accuracy values obtained by this model are 65.63%, 86.21%, 28.43%, 58.94%, 82.84%, 82.6%, 88.14% and 67.29%. The activity "Conversing on phone" is recognized at an accuracy rate of 58.94% which is an improvement of 44% over the previous model and 10-15% over the other models mentioned earlier in the chapter. The "Watching DVD", "Writing birthday card", "Preparing meal", and "Sweeping and dusting" tasks are also identified very accurately by the model. The time-based window performs better in this case for activities like "Conversing on phone". The reason is that even though the time window for this task contains very few events, the activity is labeled based on frequency of the probable label. In the case of windows with few sensor events, it is more likely for a correct label to become the most frequent label.

The count-based window for the same technique shows a maximum accuracy of 81.61% for a window count of 89 sensor events. The individual accuracy values for the 8 tasks are 58.39%, 79.98%, 34.30%, 43.28%, 75.99%, 82.42%, 83.10% and 64.79%, respectively. The time-based window outperforms the count-based window for all activities by a small percentage. The bar graph in Figure 28 compares the performance of the two models.

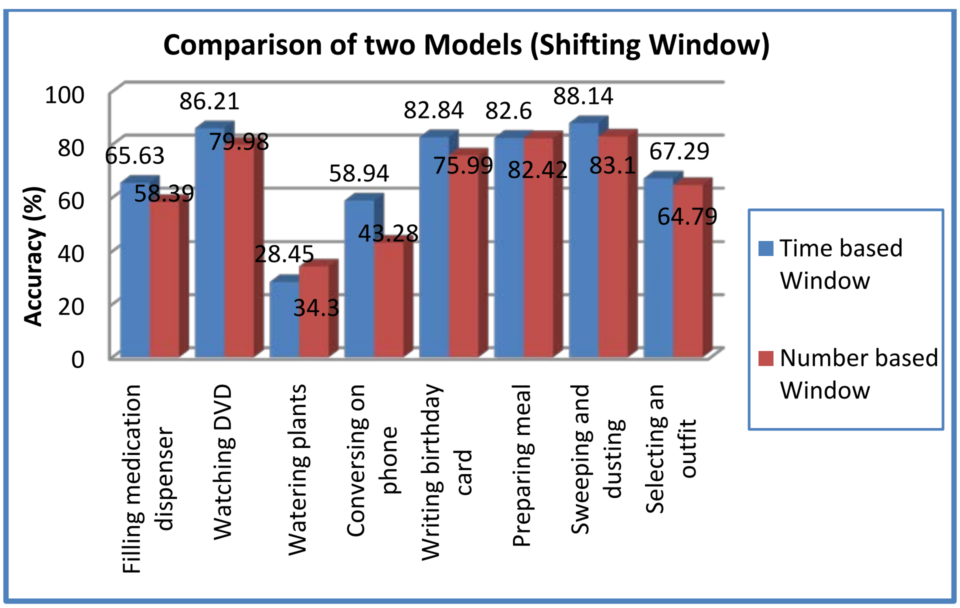


Figure 28 – Comparison of the performance of a time-based and a number-based window for HMM using a shifting window technique.

The statistical significance of comparison for these two approaches came out to be $p<0.033$ which implies that HMM with count-based shifting window outperforms the HMM with time-based shifting window with a confidence of more than 95%.

We also performed the paired t-test to compare the performance of the HMM with a time-based shifting window, with the plain HMM. The p-value for these two algorithms is $p<0.166$ which implies that the HMM with shifting window technique improves the recognition accuracy, though not very significantly. HMM with a shifting window technique performs better in comparison to all other techniques. But this approach does not completely address our problem, as it labels the whole window with one activity label instead of every sensor event. Figure 29 compares the performances of all techniques described in this chapter.

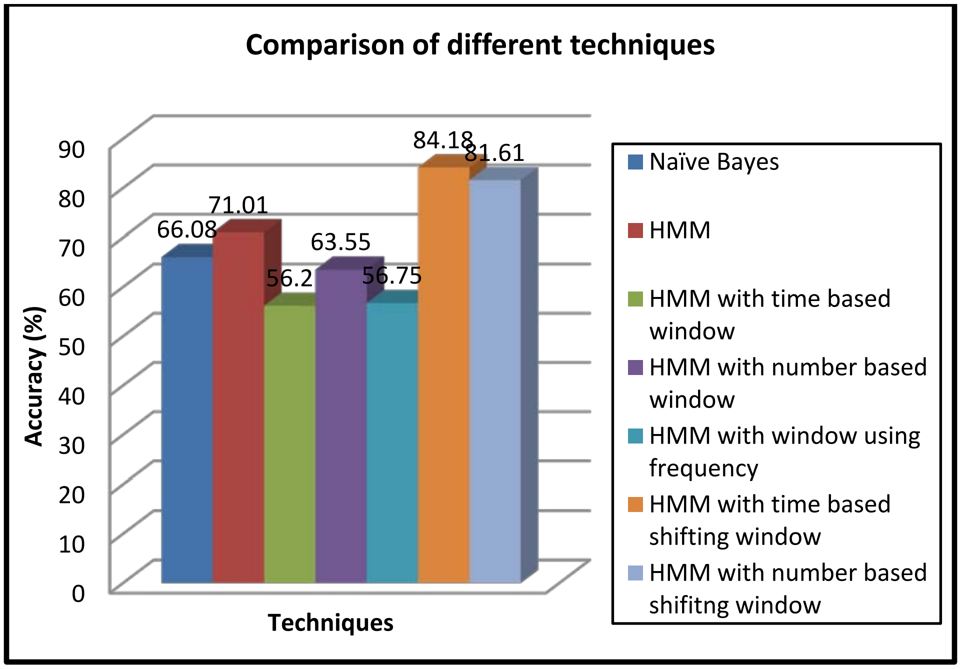


Figure 29 – Comparison of performance of all techniques in recognizing interleaved activities.

This chapter discusses various techniques for recognizing interleaved activities and from all the experiments performed, it can be said that the hidden Markov model turns out to be the most efficient technique for recognizing activities. In the next chapter, we discuss techniques for recognizing activities when the space is inhabited by multiple resident performing tasks concurrently.

CHAPTER SIX

RECOGNIZING CONCURRENT ACTIVITIES

So far we discussed approaches for recognizing mutually exclusive activities which are spread linearly over time. These activities did not have any co-temporal relationships between different activities. The algorithms described in previous chapters worked on the assumption that at most one activity would occur at any given time. In practice, however, people carry out multiple activities in parallel, possibly in different parts of a home. Additionally, previous approaches consider only one inhabitant occupying the smart space. In other words, at any given time, most approaches assume that there is only one person performing an activity in the smart space. There has not been any significant exploration on detecting parallel and concurrent activities in real life scenarios.

We already discussed the problem of activity recognition for interleaved tasks where different activities are performed in an interwoven and non-consecutive fashion. In this chapter, we consider the multi-resident case where the smart space is occupied by more than inhabitant. The residents interact with the environment to perform various assigned tasks in parallel (in the same time slice) in the same or different locations of the test bed. They also interact with each other and come together to perform several tasks collectively. In this chapter, we explore several activity recognition approaches that can handle such complex social and behavioral situations.

We conducted a multi-resident activity study in our CASAS test bed and recruited 40 individuals to perform the activities. The smart space was occupied by 2 volunteers at the same time which performed the assigned tasks concurrently, making it a multi-resident environment. The data collected from this study was then manually labeled. Specifically, each sensor event in the data was annotated with the activity Id (the activity to which the sensor event belongs) and the person Id (i.e. person A or person B). A total of 15 activities were selected for this experiment, as listed below:

Person A:

1. *Filling medication dispenser:* For this task, the participant fetches the medication dispenser and bottles of medicine from the kitchen cupboard and fills the dispenser using the space on kitchen counter.
2. *Moving furniture:* When the participant is requested for help by Person B, he goes to the living room to assist Person B in moving furniture. The participant returns to filling the medication dispenser after helping the other resident.

3. *Watering plants:* Here the participant waters the plants located on the coffee table and the side table in the living room using the watering can located in the hallway closet.
4. *Playing checkers:* The participant retrieves the checkers game from the hallway closet and then sits at the dining room table to play checkers.
5. *Preparing dinner:* For this activity, the participant sets out ingredients for dinner on the kitchen counter using the ingredients located in the kitchen cupboard.
6. *Reading magazine:* The participant begins by reading magazine while sitting in the living room. When person B asks for help, the participant goes and helps Person B in locating and dialing a phone number. After helping person B, the participant returns to the couch and continues reading magazine.
7. *Gathering and packing picnic food:* The participant gathers 5 appropriate items from the kitchen cupboard and packs them in a picnic basket. He/she helps Person B in finding dishes when asked for help. After packing is done, the participant brings the picnic basket to the front door.

Person B:

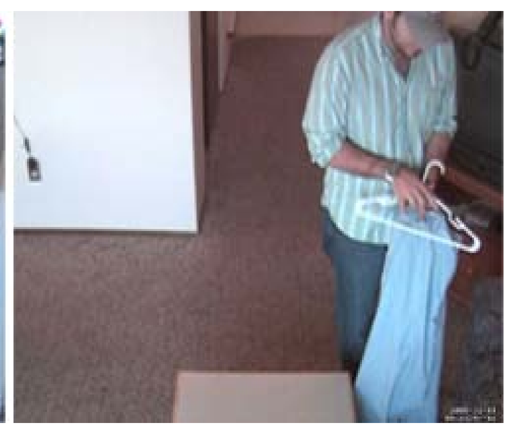
1. *Hanging up clothes:* Here the participant hangs up the clothes laid out on the couch in the living room, inside the closet located in the hallway.

2. *Moving furniture:* The participant begins by moving the couch to the other side of the living room. He/she then requests Person A for help in moving furniture. The participant (alone or with the help of Person A) also moves the coffee table accordingly.
3. *Reading magazine:* The participant sits on the couch and reads the magazine located on the coffee table.
4. *Sweeping floor:* For this task, the participant fetches the broom and the dust pan from the kitchen closet and sweeps the kitchen floor.
5. *Playing checkers:* Here the participant joins Person A in playing checkers at the dining room table.
6. *Setting table:* The participant sets the dining room table using dishes located in the kitchen cupboard.
7. *Paying bill:* The participant begins by retrieving check, pen, and an envelope from the cupboard underneath the television. He/she then tries to look up the number for Avista Utilities from the telephone book. The participant later asks Person A for help in locating and dialing the phone number. After being helped, the participant listens to the recording to find out bill balance and address for Avista Utilities. He/she then fills out a check to pay the bill, puts the check in the envelope, addresses the envelope accordingly and places it in the outgoing mail slot.

8. *Gathering and packing picnic supplies:* Finally, the participant retrieves Frisbee and picnic basket from the shelf in hallway closet and places these items on the dining room table. He/she then retrieves the dishes from kitchen cupboard and packs them in the picnic basket.

Experimental Setup

For 4 of these activities (i.e. "Moving furniture", "Playing checkers", "Paying bill" and "Packing picnic supplies"), the two residents came together and helped each other to collectively accomplish the task. The script used to conduct this experiment can be found in Appendix C.



Sensor ID/ Date/ Time		Sensor Reading	
2008-11-10	14:29:33.1159	M17	ON
2008-11-10	14:29:34.108609	M16	ON
2008-11-10	14:29:35.33645	M17	OFF
2008-11-10	14:29:37.744859	M15	ON
2008-11-10	14:29:40.285569	M03	ON
2008-11-10	14:29:42.10099	M02	ON
2008-11-10	14:29:42.73407	M01	ON
2008-11-10	14:29:43.515799	M23	ON
2008-11-10	14:29:43.77201	M02	OFF
2008-11-10	14:29:43.976389	M15	OFF
2008-11-10	14:29:44.66421	D12	OPEN
2008-11-10	14:29:46.040679	M01	OFF
2008-11-10	14:29:46.72827	M23	OFF
2008-11-10	14:30:27.88842	M17	ON
2008-11-10	14:31:05.212919	M17	OFF
2008-11-10	14:31:07.15455	M17	ON
2008-11-10	14:31:27.565	M16	ON
2008-11-10	14:31:27.90807	D12	CLOSE

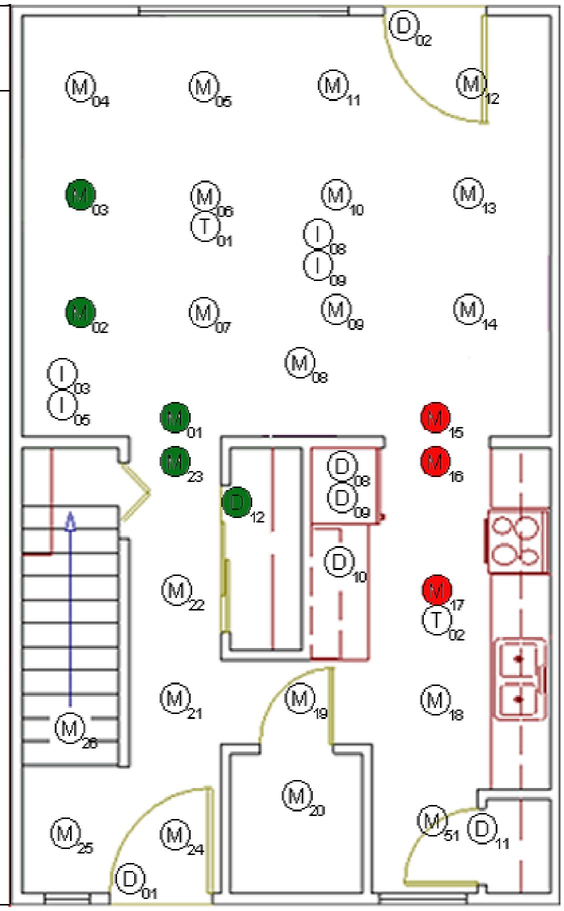


Figure 30 – The figure (top) shows two residents, person A and person B, performing two tasks, “Filling medication dispenser” and “Hanging up clothes”, respectively. The lower left figure shows the sensor readings recorded for these tasks, the readings shown in red correspond to person A and the readings shown in green color correspond to person B. The lower right figure visualizes these events on the test bed layout with red and green colors showing events belonging to person A and B, respectively.

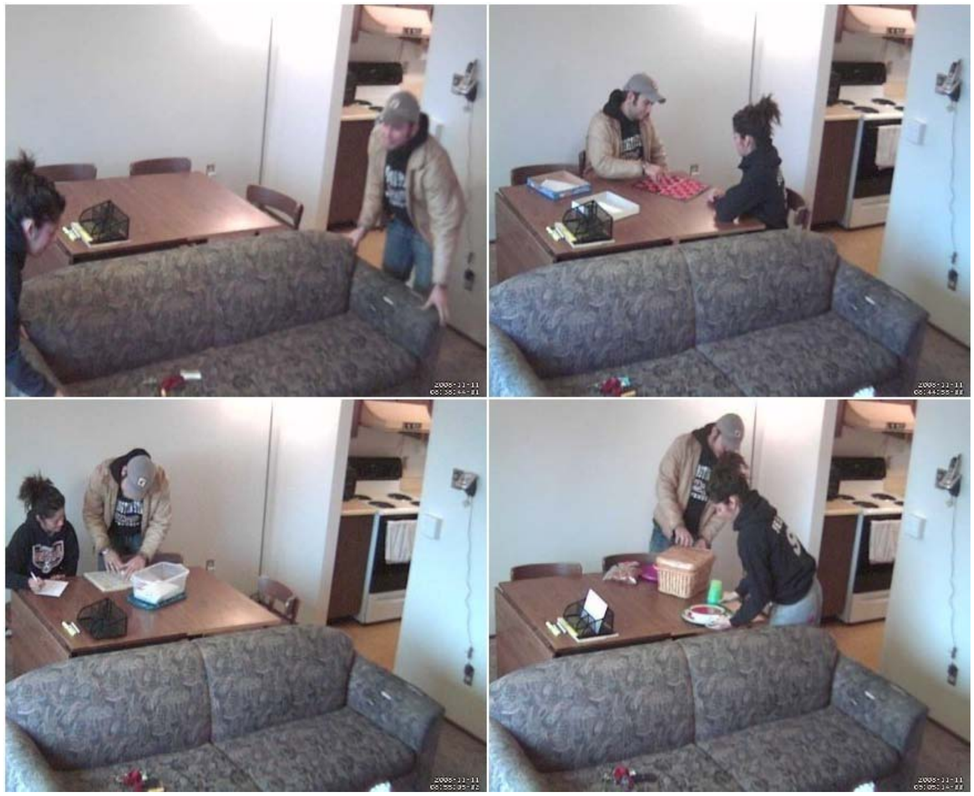
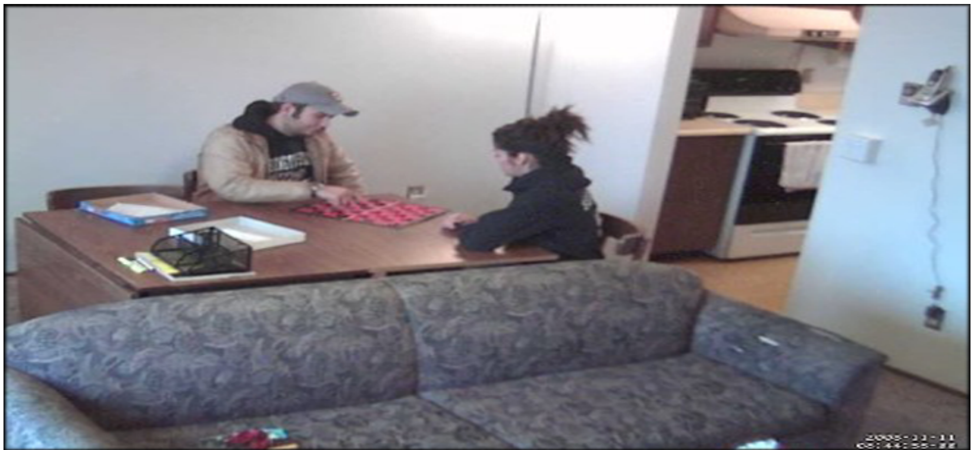


Figure 31 – The figure shows various tasks that two persons (multiple residents) are performing together. The activities are “Moving furniture” (top left), “Playing checkers” (top right), “Paying bill” (bottom left) and “Packing picnic supplies” (bottom right).



Sensor ID/ Date/ Time		Sensor Reading	
2008-11-12	16:35:22.916949	M14	ON
2008-11-12	16:35:23.440829	M09	OFF
2008-11-12	16:35:34.786819	M14	OFF
2008-11-12	16:35:38.65107	M09	ON
2008-11-12	16:35:39.590029	M09	OFF
2008-11-12	16:35:51.44053	M14	ON
2008-11-12	16:35:56.130359	M14	OFF
2008-11-12	16:35:56.610409	M14	ON
2008-11-12	16:35:58.780479	M09	ON
2008-11-12	16:36:00.380599	M09	OFF
2008-11-12	16:36:02.290309	M14	OFF
2008-11-12	16:36:23.36208	M09	ON
2008-11-12	16:36:27.439209	M14	ON
2008-11-12	16:36:29.87046	M09	OFF
2008-11-12	16:36:31.26023	M14	OFF
2008-11-12	16:37:35.19139	M14	ON
2008-11-12	16:37:36.40697	M09	ON
2008-11-12	16:37:36.876889	M14	OFF

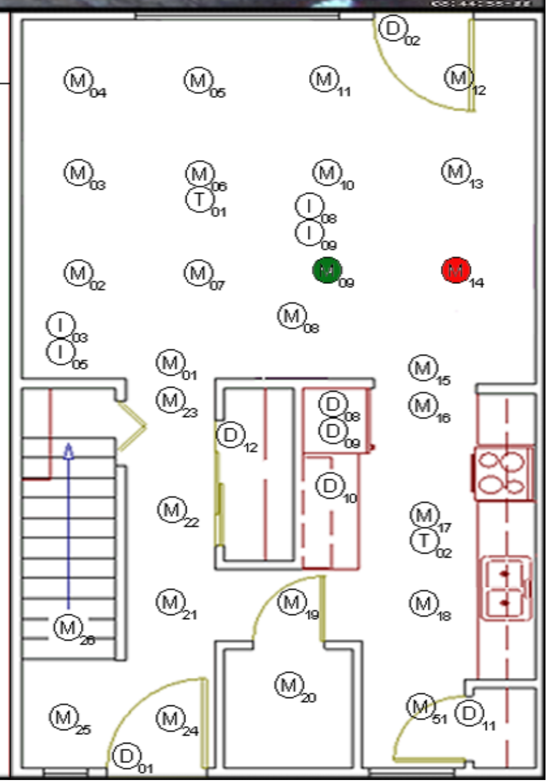


Figure 32 – Two residents are playing checkers (top). Sensor events recorded for this activity are shown in the red color for person A and green color for

person B (bottom left), the sensor layout visualization showing the state of the sensors when this activity is being performed (bottom right).

Our goal from this study is to identify the activities in a multi-resident scenario. In terms of implementation, our aim is to label every sensor event with an activity label. We used probabilistic models to accomplish this task. To recognize multiple concurrent activities, we can construct a unique model representing every activity but this approach would ignore the relationship between different activities. Hence, learning a separate Markov model for each activity will not work well for a multi-resident scenario. Hidden Markov models offer a better approach for recognizing multiple concurrent activities.

The average times taken by the Person A to perform the assigned activities are 3 minutes, 40 seconds, 2 and half minutes, 3 and half minutes, 1 and half minutes, 4 and half minutes, and 1 and half minutes, respectively. The average number of sensor events for these activities are 47, 33, 61, 38, 41, 64, and 37 sensor events, respectively. Similarly, the average times taken by Person B to perform the 8 assigned activities are 1 and half minutes, 30 seconds, 1 minute, 2 minutes, 2 minutes, 1 minute, 5 minutes, and 3 minutes, respectively. The average number of sensor events recorded for these activities are 55, 23, 18, 72, 25, 32, 65, and 38 sensor events, respectively. Some of these tasks are very short, like 'Moving furniture" performed by Person A and "Reading magazine" performed by person B, and are very hard to recognize by using any technique.

5.1 Hidden Markov Model

A hidden Markov model is a state-space probabilistic model that operates on the underlying assumption that there exist hidden states which are evolving. HMMs probabilistically infer the hidden states based on the observations up to the current time. A detailed description of hidden Markov models is given in Chapter 4, Section 4.2.

In the multi-resident implementation of a hidden Markov model, we represent the activities as hidden states. As a result, our HMM contains 15 hidden states, each of which correspond to an activity. Additionally, every sensor is treated as an observable state in the model. The prior, transition and emission probability values are calculated in the training phase which are then used to calculate the most probable activity for every sensor event. We trained our model on the multi-resident data to learn the transitions between tasks and individuals. We then perform 3-fold cross validation by training the model on 2/3 of the data in every pass and test it on the remaining 1/3. The model recognizes both the person and the activity with an average accuracy of 60.60%.

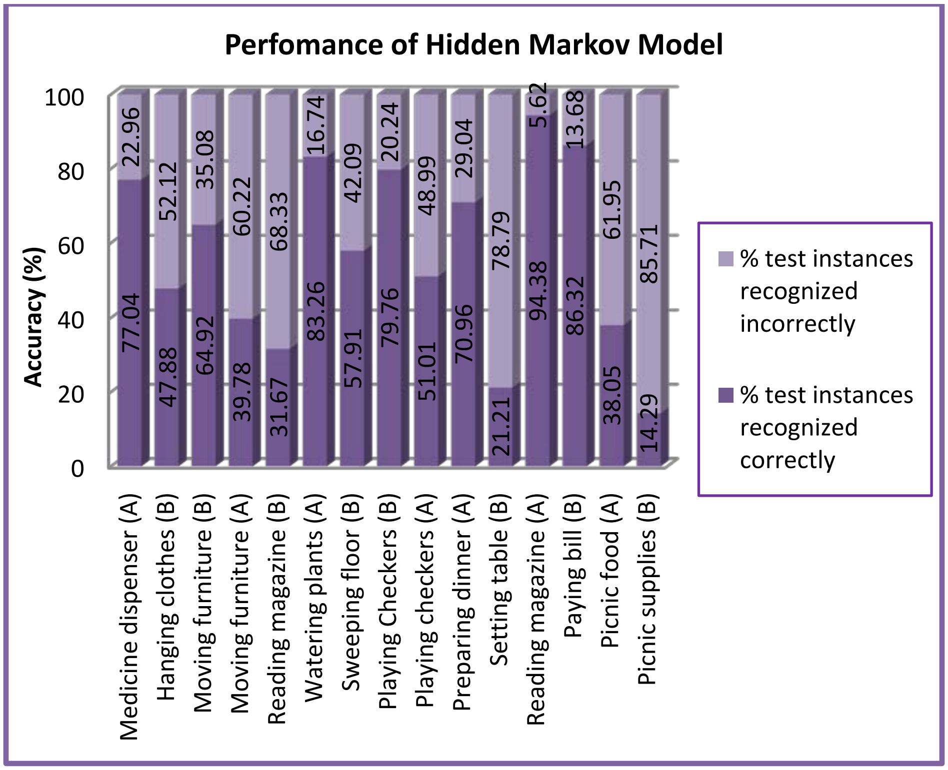


Figure 33 – Performance of a HMM in recognizing activities for multi-resident data. The label (A) or (B) represents the resident who is performing the task.

The activity "Reading magazine" performed by person A is detected with a maximum accuracy of 94%. Other activities which are predicted accurately by the model are "Filling medication dispenser" by person A, "Watering plants" by person A, "Playing checkers" by person B, and "Paying bill" by person B which show an accuracy of 77.04%, 83.26%, 79.76%, and 86.32%, respectively. Some of the activities for which the model performs poorly include “Reading magazine” by person B, “Setting table” by person B, “Packing picnic food” by person A, and “Packing picnic supplies” by person B which are identified with accuracy values of 31.67%, 21.21%, 38.05% and 14.29%, respectively.

The performance of the HMM model degrades in this case as there are two unknown parameters to be determined: the person Id and the activity Id. As an attempt to improve accuracy, we also implemented the HMM with a count-based sliding window as we did in case of the interleaved activity data. Details of the model are explained in Chapter 4 section 4.3. The HMM with a count-based window uses 2/3 of the data to find the correct window size and then uses that window size to train and test on the entire data using three fold cross validation technique. This technique results in an average accuracy of 51.93% in identifying activities for a window size of 13 sensor events. The graph in Figure 34 shows the results of this technique broken down by activity. The paired t-test when conducted for these two approaches resulted in a p-value of $p<0.008$ which means that the HMM performs better than the other approach of HMM using a count-based window with a confidence of more than 95%.

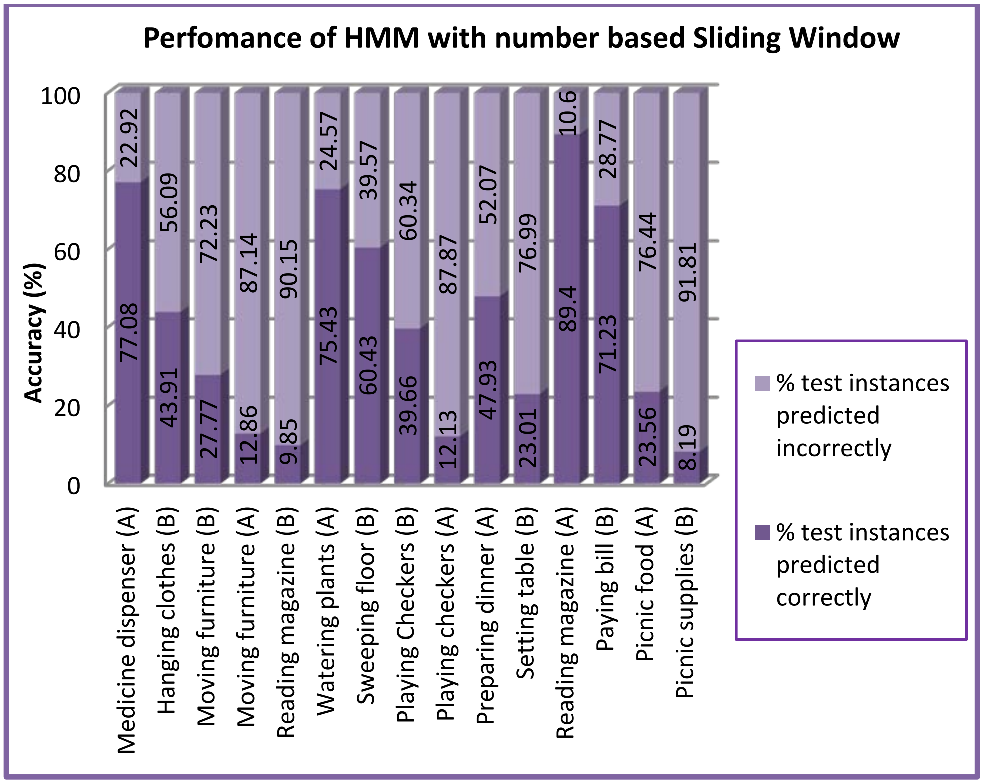


Figure 34 - Accuracy of a HMM with number based sliding window in identifying activities for all 15 activities.

The individual accuracy values broken down by activity are 77.08%, 43.91%, 27.77%, 12.86%, 9.85%, 75.43%, 60.43%, 39.66%, 12.13%, 47.93%, 23.01%, 89.40%, 71.23%, 23.56% and 8.19%, respectively. The activities "Reading magazine" by person A, "Filling medication dispenser" by person A, "Paying bill" by person B, and "Watering plants" by person A are detected with the maximum accuracy of 89.40%, 77.08%, 71.23% and 75.43% respectively. At the same time, the model shows a very poor performance in identifying activities “Moving furniture” by person A, “Reading magazine” by person B, “Playing Checkers” by person A, and “Packing picnic supplies” by person B with accuracy values of 12.86%, 9.85%, 12.13% and 8.19%, respectively. The reason for the poor performance of the model for these tasks is that the tasks are shorter in terms of number of sensor events as compared to other tasks, due to which the probability values associated with the evidences related to these hidden states are very low. The lower emission probability values for the observables of these activities eventually decreases the likelihood of these activities resulting in incorrect prediction of these activities. Also, the activities like “Reading magazine” is identified correctly by the technique but in most cases the resident performing this activity is identified as person A instead of person B which degrades the performance of the algorithm in identifying this activity. As person A has larger number of sensor events as compared to person B, even though the activity predicted by the algorithm is correct, it labels the participant as person A in most cases.

Another useful observation highlighted by these results is that for the interleaved-data HMM with a count-based window, the maximum accuracy resulted when a window size of 57 sensor events was used. On the other hand, in the case of multi-resident data, the same model shows best performance for a window size of 13 sensor events. This can be attributed to the fact that in the case of interleaved activity data, the participant was performing one task partially (for some time) and then switching to some other task. Hence, the continuous time slice consumed in performing one activity was larger. In the multi-resident experiment, there are 2 volunteers performing tasks simultaneously in the smart space due to which sensor events are being logged concurrently for both the residents. Due to this nature of the experiment, the transition from one activity (or one person to another) is made relatively much faster in the case of multi-resident data as compared to the interleaved activity data.

5.2 One HMM Per Resident

In the above implementation of a HMM, only one Hidden Markov Model is generated for both the residents. This model thus learns transitions from one resident to another and also between activities performed by different residents. However, the two residents are mostly performing different and unrelated tasks except for a few tasks like "Moving furniture" and "Playing checkers" in which they come together to collectively accomplish the tasks. Hence, the model does not need to learn transitions between activities performed by different residents. To enforce this, we implemented two separate Hidden Markov Models, one for each resident.

In this implementation, we generated one HMM for each resident; as we have 2 residents in our study, we generated 2 HMMs. One HMM consisted of all activities performed (as hidden states) and all sensors used (as observable states) by one individual. The model also had transition probabilities associated with transitions made by the individual from one activity to another. As we are discussing only activity recognition (and not person identification) in this research work, we assume that we know the person Id of the person performing the task and use this information while recognizing activities.

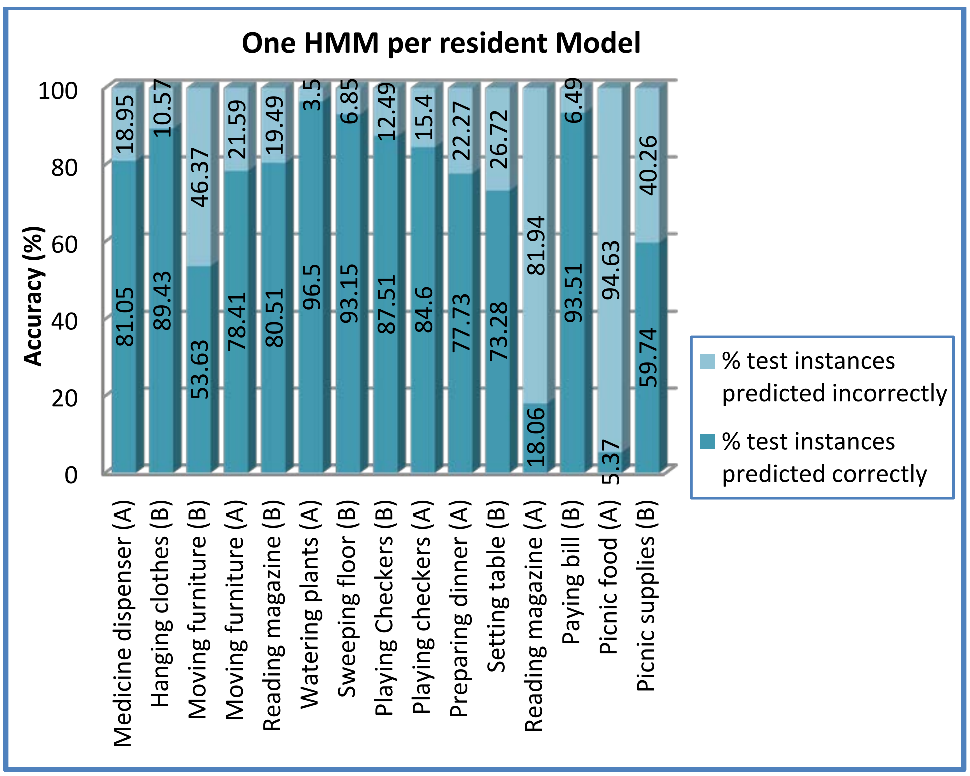


Figure 35 – Performance of the One HMM per resident Model in recognizing activities.

This technique of using two hidden Markov models for the two individuals gives an average accuracy of 73.15% in recognizing activities. The model shows an overall increase in accuracy over the previous approaches by 15%-20%. The accuracy of the approach broken down by activity is 81.05%, 89.43%, 53.63%, 78.41%, 80.51%, 96.50%, 93.15%, 87.51%, 84.6%, 77.73%, 73.28%, 18.06%, 93.51%, 5.37% and 59.74%, respectively. Activities "Watering plants" by person A, "Sweeping floor" by person B and "Paying bill" by person B are recognized with the highest accuracies of 96.50%, 93.15% and 93.51%, respectively. In contrast, some activities like "Reading magazine" by person A and "Packing picnic food" by person A are identified with very low accuracies of 18.06% and 5.37% respectively. These activities are hard to recognize because either they have fewer sensor events associated with them as compared to other activities, or they very highly resemble some other activity. Due to the lower accuracy in predicting these activities, the overall performance of the model is also degraded. The statistical significance for these two algorithms i.e. the plain HMM and the one HMM per resident model came out to be $p<0.066$ which means that the one HMM per resident model outperforms the plain HMM model, though not significantly better.

In order to compare the performance of the HMM in recognizing activities in the multi-resident scenario with the interleaved activity data, we reduce the number of activities from 15 (in multi-resident case) to 8 (as in case of interleaved activities). These 8 activities are the ones which contain the largest numbers of sensor events. These 8 activities include:

Person A – "Filling medication dispenser", "Watering plants", "Preparing dinner" and "Reading magazine"

Person B – "Hanging up clothes", "Sweeping floor", "Paying bill" and "Gathering and packing picnic supplies"

Using 3-fold cross validation to perform training and testing the model on the reduced data set containing the above mentioned 8 activities shows an average accuracy of 82.85% in recognizing the activities. The bar graph in Figure 36 shows the results of the experiment.

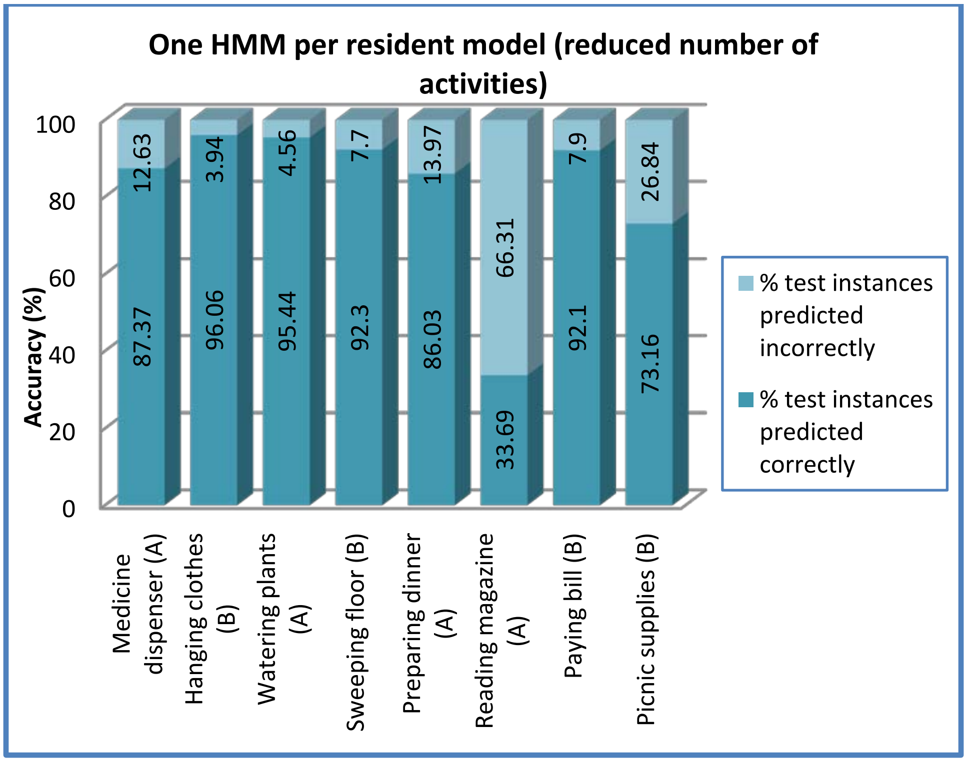


Figure 36 – Performance of one HMM per activity model for a reduced number of activities.

The accuracy values for different activities are 87.37%, 96.06%, 95.44%, 92.30%, 86.03%, 33.69%, 92.10%, and 73.16%, respectively. All activities other than "Reading magazine" by person A and "Gathering and packing picnic supplies" by person B are predicted precisely. These two activities are recognized with accuracies of 33.69% and 73.16%, respectively. The activity "Reading magazine" was performed differently by different volunteers. Also, this activity could be recognized only by the motion sensors that were recorded when this activity was performed, because there were no item (or any other) sensors associated with this activity. Volunteers sat at different places, hence involving different sets of motion sensors while performing this activity. Additionally, the motion sensors used in this activity were also a part of many other activities like "Watering plants", "Hanging up clothes", "Moving furniture" and "Paying bill". This resulted in lower evidence probability values for this activity, thereby reducing the accuracy of its prediction.

The interleaved activity experiment showed the highest accuracy of 71.01% in recognizing activities using hidden Markov model for a data set containing 8 activities, which is comparable to the performance of the one HMM per resident model for the multi-resident data having 8 activities which shows an accuracy of 82.85%. The relatively poor performance of the HMM for interleaved activity data can be attributed to the fact that the volunteers were not given any particular order in which to interweave the tasks. In the case of the multi-resident study, the participants followed a strict order in which they performed activities. This ordering consistency made it easier for the model to learn the transitions between activities.

CONCLUSIONS AND FUTURE WORK

In this work, we focused on the problem of recognizing human activity in everyday routines via supervised learning algorithms using readings from ubiquitous sensors. We investigated several possible ways of applying a probabilistic model to learn activities when they performed not only in a sequential fashion, but also in complex scenarios like when different activities are interleaved together or are performed concurrently by multiple residents. The activities that we used in our work include activities important from the perspective of daily living and medical applications such as preparing meals, washing hands, eating meals, taking medicine, and cleaning up. These activities were recognized in an accuracy range of 15% to 96% depending upon the scenario and requirements.

In addition to demonstrating that these activities can be recognized by sensors in physical environments using Markov and hidden Markov models, we also show variants of these models that help in improving the recognition accuracy. The experiments that we conducted for this study were based on an artificial script which was executed by the recruited volunteers for the purpose of data collection. As a result, this data lacked critical information like what time of day and which day of the week (weekday or weekend) the activity is performed, and which activities were performed before or after this activity.

There are observable patterns in the behavior of residents when they perform activities in real-life. For many people, using the bathroom in the morning is more likely to be associated with "Taking bath" activity than in the afternoon/ evening time. Similarly, using the kitchen sink after "cooking" has a higher probability of being a part of "Cleaning up" activity. Activity patterns for individuals also significantly differ over weekdays and weekends. This additional information can greatly help in creating a context for the current activity and in improving the 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 activities. 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.

APPENDIX A

Experimenter Instructional Protocol: Smart Environment

Sequential Task Script: Five Activities

(Button toggle off) Come to the dining room table, sit down and use the phone book to look up the number for "Safeway Food and Drugs" in Pullman. Locate the phone number for the grocery section of the Pullman "Safeway Food and Drugs" store. Dial that number and listen to the recording of the recipe that you will need for a later cooking activity. Use the pen and notepad located on the dining room table to record the pertinent information for the recipe. *(Button toggle on)*

(Button toggle off) Move into the kitchen and wash your hands using the hand-soap and paper towels provided as you are going to begin cooking. *(Button toggle on)*

(Button toggle off) Remove the materials located in the kitchen cupboard, and utensils located on the kitchen counter. Follow the recipe you recorded earlier to cook the oatmeal. After finishing, put the oatmeal in the bowl and turn off the stove. *(Button toggle on)*

(Button toggle off) Pour yourself a glass of water from the facet and remove the pill bottle from the 2nd shelf in kitchen cupboard. Bring the medication, glass of water, bowl of oatmeal and a spoon to the table and sit down. Eat the oatmeal or pretend to eat it for awhile. Take 3 pills out of the container and set them on the table when finished with the oatmeal. *(Button toggle on)*

(Button toggle off) Put the medication back in the pill bottle, gather your dishes and the pill bottle and move into the kitchen. Once in the kitchen, wash the dishes and then place them in the drying rack. Also place the pill bottle and other materials back in the cupboard. *(Button toggle on)*

APPENDIX B

Experimenter Instructional Protocol: Smart Environment

Interleaved Task Script: Eight Activities

(Button toggle off) For your first `task, I am going to have you fill a medication dispenser. Please do not initiate the task until I have completed the instructions and said begin. You will find the medication dispenser, the bottles of medication and the directions for filling the medication dispenser in the kitchen cupboard labeled "A". Once you have located the items, please remove the items from the cupboard and use the space on the kitchen counter to fill the dispenser. Once you follow the directions and fill the medication dispenser, please return all items to cupboard "A". You may begin. *(Button toggle on)*

(Button toggle off) Now, when I say begin, I would like you to move into the living room and select the DVD labeled "Good Morning America" from the pile of DVDs located on the shelf below the TV. Once you have found the "Good Morning America" DVD, please follow the instructions posted near the TV, which will allow you to watch on the TV the 5-minute news clip contained on the DVD. After you have watched the DVD, please turn off the TV and return the DVD to the pile of DVDs on the shelf. You may begin. *(Button toggle on)* Wait for the participant to finish.

(Button toggle off) For this next task, I would like you to lightly water the apartment plants. There are 3 plants; two plants are located on the kitchen windowsill and the other plant is located on the living room table. The watering can is located on a hook in the kitchen closet, which is located near the kitchen window and labeled "supply closet". When you finish with this task, please empty any extra water from the watering can into the sink and return the watering can to the hook in the kitchen "supply closet". You may begin. *(Button toggle on)* Wait for the participant to finish.

(Button toggle off) In a minute, the phone is going to ring. Please answer the phone. After you finish answering the phone, you can hang up the phone. *(Button toggle on)* Wait for the participant to finish.

(Button toggle off) For this next task, I would like you to imagine that you need to send a Birthday card along with a birthday check to a close relative. The supplies for completion of this task are located on the dining room table. Please choose a card and write a birthday wish inside the card. In addition, write a check in a suitable amount for a birthday gift. Place the card and the check in the envelope and address the envelope appropriately. After you have finished, please leave the envelope on the dining room table. Later when I say "We are now done with the first set of activities", I want you to retrieve

the letter from the dining room table and bring it to the front door. *(Button toggle on)* Wait for the participant to finish.

(Button toggle off) For the cooking task, I would like you to pretend that you are preparing a cup of noodle soup and getting a glass of water for a friend. A glass, a measuring cup, the cup of noodle soup, and utensils are located in the cupboard labeled "A". Please fill the measuring cup with water and microwave for 3 minutes. Then follow the remaining directions on the cup of noodle soup to prepare the soup. In addition, please fill the glass with water using the pitcher of water located on the top shelf of the refrigerator. Bring all items to the dining room table for your friend. *(Button toggle on)*

(Button toggle off) For this next task, I would like you to sweep the kitchen floor and dust the dining room and the living room. All supplies that you will need are located in the kitchen closet labeled "supply closet". When you have finished this task, please return the supplies that you used to sweep the floor and dust to the kitchen "supply closet". You may begin. *(Button toggle on)* Wait for the participant to finish.

(Button toggle off) In just a minute, I am going to have you select an outfit from the clothes closet to be worn by a male friend. Please pretend that your male

friend is going on an important job interview. Please choose an appropriate interview outfit from the closet labeled "clothes closet", which is located in the entrance hallway. After you have chosen the interview outfit for your friend, please lay the clothes out on the living room couch. You may begin. *(Button toggle on)*

Interweaving Task:

I would now like you to complete all eight of these task again. This time, however, I would like you to consider how you might most efficiently complete these daily tasks in your everyday environment. You can complete the task in whatever order you wish. Most importantly, you can multi-task and interweave tasks in a way that feels natural to you. This card contains the list of the eight activities that we want you to complete again. I will leave this list of the activities on the dining room table in case you need to refer to it. Please begin now and let me know when you have finished with all eight tasks. Remember, we want you to multi-task and interweave the tasks in order to complete the tasks in a way that feels natural and most efficiently. *(Button toggle on)*

After the participant has successfully completed all of the tasks (*Button toggle off)*.

APPENDIX C

Experimenter Instructional Protocol: Smart Environment

Multi-resident Task Script: 2 Person Protocol

For the first task, Person A will fill 2 medication dispensers, while Person B will be doing a couple of different tasks in the living room. Please do not initiate the tasks until I have completed the instructions and said begin. Person A, You will find the medication dispensers, the bottles of medication and the directions for filling the medication dispensers in the kitchen cupboard labeled "A". The directions for filling the dispensers are taped to the inside of the cupboard door. Once you have located the items, please remove the items from the cupboard and use the space on the kitchen counter to fill the dispensers. Once you have followed the directions and filled the medication dispensers, please return all items to cupboard "A", including the filled medication dispensers. Person B, you will begin by hanging up the clothes that are laid out on the couch in the living room. The closet, labeled "clothes closet", is located in the hallway. After you have finished hanging up all the clothes, your next task will be to move the couch to the other side of the living room. You should also move the coffee table accordingly. However, for this task you will request help from Person A. Person A, please stop filling the medication dispensers and come to the living room to assist Person B when the request for help is made. After you have both successfully moved the furniture, Person A please return to filling the medication dispensers. Person B, you may then sit on the couch and read the magazine located on the coffee table until Person A has finished filling the medication dispensers. Okay, you may now begin. *(Button toggle off)*

(Button toggle on) Great. For the next task, I would like Person A to begin by lightly watering the plants while Person B will be sweeping the kitchen floor. The watering can is located in the hallway closet. The broom and the dust pan are in the kitchen closet labeled "supply closet", located next to the window sill. Person A, the plants are located on the coffee table and the side table in the living room. After you have watered the plants, please empty any extra water from the watering can down the sink, and return the watering can to the hallway closet. After you have returned the watering can, please retrieve the "checkers" game from the shelf in the hallway closet. Person B after you have swept the kitchen floor and used the dust pan please return the materials to the supply closet. You can play the "checkers" game at the dining room table. After the game is complete or 5 minutes have elapsed, please put the game back in the closet, and move on to the third task. Person A, you will be setting out ingredients for dinner on the kitchen counter, while Person B will be setting the dining room table. The recipe for the meal is located on the refrigerator door. The ingredients for the meal are

located in the kitchen cupboard labeled "A". The dishes for the table are located in the kitchen cupboard labeled "B". Okay, you may now begin. *(Button toggle off)*

(Button toggle on) For this next task, I would like Person A to read a magazine in the living room, while Person B will be paying an electric bill. Person B, for your first task you will need to retrieve a check, a pen, and an envelope from the cupboard underneath the television. The electric company you will be making the payment to is Avista Utilities. You will need to use the telephone book to look up the number for Avista Utilities so that you can confirm the amount you need to write the check out for, as well as the address for Avista Utilities. For this task, I would like Person B to simulate someone having difficulties locating the number for Avista Utilities in the phone book and using the phone. Person B, I would like you to request that Person A help you locate and dial the phone number for Avista Utilities. Person A, after you have helped Person B locate and dial the phone number, you can return to the couch and continue reading the magazine. Person B, after you have dialed the number and listened to the recording, which provides you with your bill balance and the address for Avista Utilities, please fill out a check to pay the bill, put the check in the envelope and address the envelope accordingly. After you have addressed and sealed the envelope, you

can place it in the outgoing mail slot of the organizer. Okay, you may now begin. *(Button toggle off)*

(Button toggle on) For the last task, I would like you to get ready for a picnic. I would like Person A to gather the food for the picnic from the kitchen cupboard labeled "A". Please find 5 items that would be appropriate for a picnic. Meanwhile, I would like Person B to retrieve the frisbee and the picnic basket from the shelf in the hallway closet. After retrieving the frisbee and the picnic basket, please place these items on the dining room table. Person B, you will then need to retrieve the dishes from the cupboard labeled "B" and pack them in the picnic basket. However, I would like you to simulate someone getting confused and having difficulty locating the dishes by looking in the wrong cupboards and drawers, as well as looking in the supply closet and the microwave. After it becomes clear that Person B is having difficulty locating the dishes, I would like Person A to help Person B by physically showing them the cupboard that the dishes are located in, (i.e., cupboard "B"). After Person A has physically shown Person B where the dishes are located, then Person B should pack the dishes into the picnic basket. Person A, you should pack the 5 items of food into the picnic basket. After the picnic basket is packed, please bring it to the front door. Okay, you may now begin. *(Button toggle off)*

BIBLIOGRAPHY

1. W. He, M. Sengupta, V. Velkof and K. DeBarros: Current Population Reports. *U.S. Department of Health and Human services, 2006*

2. http://www.alzheimers.org.uk/society/p_demography.htmlUnited Kingdom Alzheimer Society, *Prevalence and Incidence of Dementia. 2001, Alzheimer Society (UK)*

3. C. Hawes, C. D. Phillips, M. Rose, S. Holan, and M. Sherman: A National Survey of Assisted Living Facilities. *The Gerontological Society of America 2003*

4. C. Fagerström, H. Persson, G. Holst, I. R. Hallberg: Determinants of feeling hindered by health problems in daily living at 60 years and above. *Scandinavian Journal of Caring Sciences, Volume 22, Number 3, September 2008 , pp. 410-421(12)*

5. S. Iwarsson, V. Horstmann, U. Sonn: Assessment of dependence in daily activities combined with a self-rating of difficulty. *Journal of Rehabilitation Medicine, Volume 41, Number 3, February 2009 , pp. 150-156(7)*

6. S. Katz, "Assessing Self-Maintenance: Activities of Daily Living, Mobility, and Instrumental Activities of Daily Living", *Journal of the American Geriatrics Society, vol. 31, no. 12, pp. 721 - 726, 1983*

7. A. Mihailidis, B. Carmichael, J. Boger, and G. Fernie: An Intelligent Environment to Support Aging-in-Place, Safety, and Independence of Older Adults with Dementia. *The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, 2003*

8. L. Liao, D. Fox, and H. Kautz: Extracting Places and activities from GPS Traces Using Hierarchical Conditional Random Fields. *International Journal of Robotics Research, 2007*

9. L. Liao, D. Fox, and H. Kautz: Location-based Activity Recognition. *International Joint Conference on Artificial Intelligence (IJCAI), 2005*

10. X. Sun, C. Chen, B. S. Manjunath: Probabilistic motion parameter models for human activity recognition. *IEEE International Conference on Pattern Recognition (ICPR02), 1, 2002*

11. J. Ben-Arie, Z. Wang, P. Pandit and S. Rajaram: Human activity recognition using multidimensional indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions 2002, Volume 23, Pages 1091- 1104*

12. P. C. Ribeiro and J. Santos-Victor: Human Activity Recognition from Video: modeling, feature selection and classification architecture. *International Workshop on Human Activity Recognition and Modeling, Oxford, UK, 2005*

13. N. Vaswani, A. R. Chowdhury, R. Chellappa: Activity recognition using the dynamics of the configuration of interacting objects. *Computer Vision and Pattern Recognition, Proceedings. 2003 IEEE Computer Society Conference*

14. N. Robertson and I. Reid: A general method for human activity recognition in video. *Computer Vision and Image Understanding 2006, Issues 2-3, Pages 232-248*

15. S. Hongeng, R. Nevatia and F. Bremond: Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding 2004, Issue 2, Pages 129-162*

16. D. J. Patterson, D. Fox, H. Kautz: Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. *Ninth IEEE International Symposium on Wearable Computers, 2005*

17. T. Gu, Z. Wu , L. Wang , X. Ta and J. Lu: An Emerging Patterns based Approach to Sequential, Interleaved and Concurrent Activity Recognition. In Proc. of the *7th Annual IEEE International Conference on Pervasive Computing and Communications, 2009*

18. A. Subramanya, A. Raj, J. Bilmes and D. Fox: Recognizing Activities and Spatial Context UsingWearable Sensors. *In Proc. of 22nd Conference on Uncertainty in Artificial Intelligence, 2006*

19. M. Stikic, T. Huynhy, K. V. Laerhoveny and B. Schiele: ADL recognition based on the combination of RFID and accelerometer sensing. *Pervasive Computing Technologies for Healthcare, 2008*

20. N. Ravi, N. Dandekar, P. Mysore and M. L. Littman: Activity Recognition from Accelerometer Data. *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference, 2005*

21. J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg: A scalable approach to activity recognition based on object use. *Proceedings of Int'l conference on computer vision, 2007*

22. D. H. Hu and Q. Yang: CIGAR: concurrent and interleaving goal and activity recognition. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008), Chicago, Illinois, USA, 2008*

23. T. Wu, C. Lian, and J. Y. Hsu, Joint recognition of multiple concurrent activities using factorial conditional random fields, In *Proceedings of AAAI Workshop on Plan, Activity, and Intent Recognition, California, July 2007*

24. D. L. Vail, M. M. Veloso, and J. D. Lafferty, Conditional random fields for activity recognition. *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2007*

25. S. Gong and T. Xiang : Recognition of group activities using dynamic probabilistic networks. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference*

26. D. H. Wilson, C. Atkeson: Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. *The Third International Conference on Pervasive Computing, 2005*

27. N. Oliver, E. Horvitz and A. Garg: Layered representations for human activity recognition. *Multimodal Interfaces, Proceedings. Fourth IEEE International Conference 2002, Pages 3-8*

28. D. H. Wilson, C. Atkeson, Simultaneous tracking & activity recognition using many anonymous, binary sensors, in: *The Third International Conference on Pervasive Computing, 2005, pp. 62–79*

29. J. Carter and M. Rosen. Unobtrusive sensing of activities of daily living: A preliminary report. *In roceedings of the 1st Joint BMES/EMBS Conference, page 678, 1999*

30. H. B. Jimison, M. Pavel, and J. Pavel: Adaptive interfaces for home health. In *Proceedings of the International Workshop on Ubiquitous Computing for Pervasive Healthcare, 2003*

31. S. Moncrieff: Multi-modal emotive computing in a smart house environment. *Journal of Pervasive and Mobile Computing, special issue on Design and Use of Smart Environments, 2007*

32. T. S. Barger, D. E. Brown, and M. Alwan: Health status monitoring through analysis of behavioral patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 35(1):22–27, 2005*

33. D. Cook, G. M. Youngblood, and G. Jain: Algorithms for smart spaces. *Technology for Aging, Disability and Independence*: *Computer and Engineering for Design and Applications. Wiley, 2006*

34. S. Luhr: Recognition of emergent human behaviour in a smart home: A data mining approach. *Journal of Pervasive and Mobile Computing, special issue on Design and Use of Smart Environments, 2007*

35. G. M. Youngblood and D. Cook: Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 37(4):1-12, 2007*

36. S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen: The gator tech smart house: A programmable pervasive space. *IEEE Computer, 38(3):50–60, 2005*

37. F. Doctor, H. Hagras and V. Callaghan: A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(1):55-56, 2005

38. G. D. Abowd and E. D. Mynatt: Designing for the human experience in smart environments. In Diane J. Cook and Sajal K. Das, editors, *Smart Environments: Technology, Protocols, and Applications*, pages 153-174, Wiley, 2004

39. M. Mozer. Lessons from an adaptive home. In Diane J. Cook and Sajal K. Das, editors, *Smart Environments: Technology, Protocols, and Applications*, pages 273-298, Wiley, 2004

40. H. Kautz, D. Fox, O. Etzioni, G. Borriello, and L. Arnstein: An overview of the assisted cognition project. *In Proceedings of the AAAI Worskhop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care, pages 60–65, 2002*

41. A. Mihailidis, B. Carmichael, and G. Fernie: The efficacy of an intelligent cognitive orthosis to facilitate handwashing by persons with moderate-to-severe dementia. *Neuropsychological Rehabilitation, 14(1/2):135–171, 2004*

42. 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*

43. J. M. Wiener and R. Hanley: Measuring the Activities of Daily Living: A Guide to National Surveys. *U.S. Department of Health and Human Services, 1989*

44. http://dhs.wisconsin.gov/aging/dementia/ADL.pdf Activities of Daily Living (ADL's) and Instrumental Activities of Daily Living (IADL's). *Wisconsin Department of Health Services, 2004*

45. http://dhs.wisconsin.gov/aging/dementia/stages.pdf Stages of Alzheimer's Disease, *Wisconsin Department of Health Services, 2004*

46. M. L. Puterman: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, Inc. New York, 1994

47. A. Papoulis: Bayes' Theorem in Statistics and Bayes' Theorem in Statistics (Reexamined). §3-5 and 4-4 in *Probability, Random Variables, and Stochastic Processes, 2nd ed.* New York: McGraw-Hill, pp. 38-39, 78-81, and 112-114, 1984

48. D. Wyatt, M. Philipose and T. Choudhury: Unsupervised activity recognition using automatically mined common sense. *Proceedings of the National Conference of Articial Intelligence, 2005*

49. B. Logan, J. Healey, M. Philipose, E. Munguia-Tapia and S. Intille: A Long-Term Evaluation of Sensing Modalities for Activity Recognition. *UbiComp 2007: Ubiquitous Computing, Volume 4717 Pages 483-500, 2007*

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

51. C. R. Wren and E. Munguia-Tapia: Toward scalable activity recognition for sensor networks. *Proceedings of the Workshop on Location and Context-Awareness, 2006*

52. J. Pineau, M. Montemerlo, M. Pollack, N. Roy and S. Thrun: Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems, 42(3-4), 2003*

53. L. E. Hebert, P. A. Scherr, J. L. Bienia, D. Denis, A. Evans: Alzheimer's disease in the US population: *Prevalence estimates using the 2000 census. Archives of Neurology, 60:1119-1122, 2000*

54. E. F. LoPresti, A. Mihailidis, N. Kirsch: Assistive technology for cognitive rehabilitation: State of the art. *Neuropsychological Rehabilitation, 14(1/2):539, 2004*

55. M. Perkowitz, M. Philipose, K. Fishkin, D. J. Patterson: Mining models of human activities from the web. *Proceedings of the 13th international conference on World Wide Web, 2004*

56. D. Cook, G. M. Youngblood, E. O. Heierman, K. Gopalratnam, S.Rao, A. Litvin, and F. Khawaja: MavHome: An Agent-Based Smart Home. *Pervasive Computing and Communications, 2003*

57. T.V. Duong, H. H. Bui, D. Phung, S. Venkatesh: Activity recognition and abnormality detection with the switching hidden semi-Markov model. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference, 2005*