ACTIVITIES OF DAILY LIVING DETECTION USING MARKOV MODELS

by

Love P. Kalra

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
December 2011

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "ACTIVITIES OF DAILY LIVING DETECTION USING MARKOV MODELS" by Love P. Kalra in partial fulfillment of the requirements for the degree of Master of Computer Science.

Dated: December 8, 2011

Supervisor: ____________________

Readers: ____________________

____

DALHOUSIE UNIVERSITY

DATE: December 8, 2011

AUTHOR: Love P. Kalra

TITLE: ACTIVITIES OF DAILY LIVING DETECTION USING MARKOV MODELS

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MCSc CONVOCATION: May YEAR: 2012

Signature of Author

To my family,

and friends

Table of Contents

List of Tables

List of Figures

Abstract

The healthcare systems are experiencing heavy workload and high cost caused by ageing population. The assisted monitoring systems for the elderly persons, and persons with chronic diseases, promises great potential to provide them with care and comfort at the privacy of their own homes and as a result help reduce healthcare costs. This requires a monitoring system capable of detecting daily human activities in living spaces. In this work we discuss different challenges to design such a system, present an activity data visualization tool designed to study human activities in a living space and propose a two stage, supervised statistical model for detecting the activities of daily living (ADL) from non-visual sensor data streams. A novel data segmentation is proposed for accurate prediction at the first stage. We present a novel error correction structure for the second stage to boost the accuracy by correcting the misclassification from the first stage.

List of Abbreviations and Symbols Used

ADL	Activities of Daily Living
HHMM	Hierarchical Hidden Markov Model
HMM	Hidden Markov Model
MLROC	Maximum Likelihood Rate of Change
MMM	Multi-Markov Model
MMM-FS	Multi-Markov Model - First Stage
MMM-SS	Multi-Markov Model - Second Stage
MNBC	Multi-Naive Bayes Classifier
ROCP	Rate of Change Percentage

Chapter 1

Introduction

The increasing cost of the healthcare system with sudden increase in ageing population has lead to the proposition of an idea of autonomous monitoring systems for the elderly. A system that can provide the elderly with care and supervision at the comfort of their own homes. In addition, increasing interest of home automation and home security industry for autonomous intrusion monitoring and better energy management in a house has opened great potential for research in ubiquitous computing. Automatic detection of daily human activities can greatly benefit the in-home elder care and home automation systems but creates several technical challenges and ethical responsibilities to perform the monitoring effectively without compromising a person's privacy. A typical activity detection system is shown in Figure 1.1. The automatic activity detection requires a house or a living space to be equipped with a sensor network and a system capable of making decisions according to the occupants activities by learning their regular behaviour. A filter module can be designed to isolate the irregular behavior of activities that may be further analysed in a analysis module and based on different applications different actions could be taken by the actuation module. The actuation module can be an activity logger, that can be used to log the activity patterns to study different users or it can be a monitoring system that sounds alert in case of emergency or security breach. The focus of this research work is to design an accurate activity detection engine.

The inherent presence of complex human behaviour in daily activities creates a great challenge for the effective modeling of the activities. The activities can be ambiguous and highly variable in nature from one person to another; moreover different activities or even different instances of the same activity may vary widely in duration of performing an activity or the usage of object while performing an activity. For example, the meal preparation activities like preparing breakfast, lunch and dinner can be performed very quickly when minimum cooking is involved and may

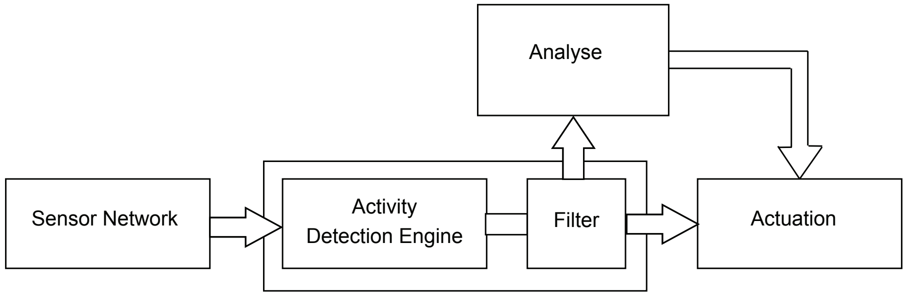


Figure 1.1: A typical activity detection system.

take hours when cooking a big meal. Similarly, some times a person would use a cooker to prepare a meal and sometimes just use a knife to slice vegetables to make a sandwich. Another typical characteristic of human behaviour in every day activities occurs when people perform more than one activity at the same time, commonly referred to as *concurrent* activities and sometimes with interleaving periods known as *interleaving* activities. One example of *concurrent* activities is, a scenario where a person is watching TV and is interrupted by a phone call, assuming the phone is right next to the couch where the person is sitting, he picks up the phone and start talking. From the sensor's point of view the two activities of watching TV and talking on the phone are being executed simultaneously, although it can be argued that the person is doing only one activity actively at a time and the other fades out in the background. Once the person finishes the phone conversation he gets back to actively watching the TV again. An example of *interleaving* activity is the following: a person starts doing *laundry*, he opens up the laundry cabinet and takes out the detergent. In this process he triggers the sensors attached to the laundry cabinet and laundry detergent jug, indicating the start of the laundry activity. He then suddenly gets interrupted by a phone call, once finished on the phone he returns back to the laundry task. The concept of concurrent and interleaving activities have been discussed repeatedly in the literature [11, 13] and it has been argued that the sequential activity models like Hidden Markov Model (HMM) and Conditional Random Field (CRF) Model are unable to detect the concurrent and interleaving activities. However, it can be counter argued that one way to capture the concurrent activities in a sequential activity model framework is by simply grouping these activities into one, so following the same example of *watching TV* and *phone conversation* we can say that the activity sequence is *watching TV*, *watching TV-phone* and *watching TV*, where *watching TV-phone* being an activity where they occur simultaneously. Representation of the interleaving activities require *long-term temporal dependencies* [26, 40] between the activities, that may be achieved by using layered HMM architecture [9] by modeling the activities at different granularities in different layers. Other variants of HMM such as Hidden Semi-Markov models [8], Hierarchical Hidden Markov models [3] and Decomposed Hidden Markov model [40] have been proposed to model the *long-term temporal dependencies* in different domains. To adapt the generic HMM to model different characteristics of the process, these different variants of HMM make the learning and inference highly complicated and may not be suitable for online applications, where inference time is critical. Within the CRF framework, the Skip Chain CRF model (SCCM) is proposed in [11] to model the interleaving activities.

To capture the complex human activities we require a system capable of identifying different activity patterns performed by the occupant in a living space, with an ability to disambiguate them in the presence of spurious activations of the sensors and activity labelling errors. The model should be able to capture the *long-term dependencies* to allow the modeling of the interleaving activities for proper detection.

To acquire the intuitive understanding of the complexity of different behaviours making up an activity, in this work we designed an activity visualization tool. This visualization tool is designed to visualize the data from different activity datasets that are collected from sensor equipped houses by different research groups in this domain.

Activity Modeling Everyday human activities are associated with two feature sets, temporal and spatial features. Studying the daily behaviours in depth gives the insight of how these features can be used to effectively infer the activities. We can divide the temporal features further into two sets, absolute and relative [38]. The absolute temporal feature gives the approximate time of the day the activity is usually conducted. Although absolute time feature is important, it can only be relevant to certain set of activities like *sleeping*, *going to work* or *eating a meal*, which are more likely to happen every day in a fixed time slot. Other activities may have irregular daily pattern that may change every day, like *eating a snack* or *preparing a beverage*, *reading* or *entertainment*. The relative temporal feature on the other hand, gives information on which activity is more likely to follow another activity. For example, it is more likely that the person will dress after taking a shower rather than eating his meal. Absolute and relative temporal features if used in conjunction can strengthen the belief in the occurrence of an activity in an error prone sensory environment. In addition, if we consider the activities at their atomic level, the relative temporal feature can be more useful. Activities at their elemental level constitute several steps that may follow a certain order. For example, activity *preparing a beverage* consists of steps such as fetching a glass and opening a refrigerator. The relative temporal feature at the atomic level of activities can help distinguish them by identifying the subtle differences between closely related activities where common sensors may trigger.

Another important characteristic of the daily activities that can be helpful is location or a spatial feature. Certain activities are very location specific, for example preparing a meal can only be done in kitchen, toileting and showering activities can only occur in a bathroom. The spatial feature can be very useful as it can be used to increase the accuracy of activity prediction for certain activities. It may also improve the efficiency of the model by considering the sensor data only from the activity specific locations.

Modeling of human activities starts by quantifying different characteristics of the activities. This is done by capturing human actions over time using electronic sensors. Given the contextual and temporal nature of human activities in a typical home setting, several different approaches have been proposed to quantify the human activities using visual and non-visual sensors [28, 21, 27, 18]. The video or image based activity recognition approaches, focus on human posture and locations to record the activities and non-visual sensors rely on the human-object interaction that is captured by the sensors that are built into the surrounding objects. To alleviate the privacy concerns to some extent, in this work we use the non-visual sensors to capture the activities. In human-object interaction approach that uses non-visual sensors, the triggering of sensors individually or in a sequence is interpreted as an execution of an activity. For example, in a house RFID sensors can be attached to different objects in kitchen such as cups, saucers and stove knobs. The use of these objects is registered by a reader attached to a wrist or other parts of a human body. The switch sensors can be built into different cabinets, closets and faucets in different rooms of the house, such as bathroom, kitchen and bedrooms. The human interaction with the sensor equipped objects such as retrieving a glass and reaching to the kitchen faucet might imply the person is thirsty and is getting a glass of water. Turning on the shower faucet indicates the person is about to take a shower. In this work our focus is on non-visual sensor perception of activities. We use a human-object interaction dataset which is discussed in detail in chapter 4.1.

Traditional approaches with HMM and the variants of HMM, where all the activities are modeled as states, has been the most popular approach [35, 4] to model ADLs. Although, modeling the activities with HMM in this manner has been shown to be feasible but lacks the flexibility to make use of the temporal features embedded at different levels. HMM with activities as states fails to capture the temporal information within the activities, also known as intra-activity information. The intra-activity information is very important for distinguishing between the activities that trigger the same set of sensors but have different triggering patterns. The thorough inspection of each activity leads to the conclusion that the nature of different activities can vary widely and a "single model fits all "approach [35] may not be the best route towards the development of a robust and accurate model for the detection of the daily activities. In this work we chose the layered-HMM architecture because of its several different advantages. Firstly, as mentioned above it allows the process to be modeled at different granularities that allow *long term dependencies* between the states which are difficult to model with the single layer HMM. Secondly, without adding the complexities of the additional parameters for the different variants of HMMs discussed above we can achieve the long term dependencies for the activities that are performed for longer periods in addition to the short term dependencies for the activities that are performed for short periods. The activities of daily living have complex time dependence and have the hierarchical structure [23] at different time granularities, that can be modeled effectively using the layered-HMM architecture. To cover the complete spectrum of different characteristics of each activity we chose to model each activity individually. To accommodate the low order temporal feature of the activities of daily living (ADL) we propose a two stage approach to model the

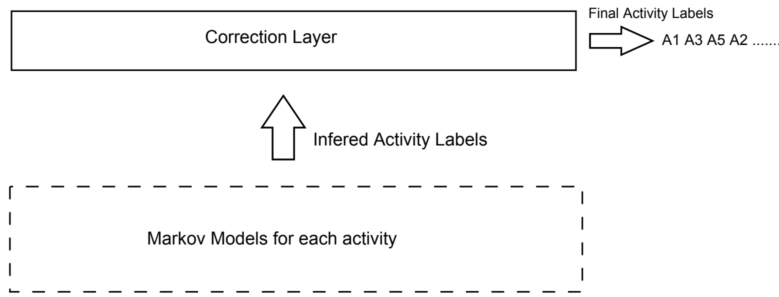


Figure 1.2: General architecture of the two stage model.

ADLs.

The general architecture of the proposed two stage model is shown in Fig. 1.2. The first stage consists of an individual Markov chain for each activity. Although this configuration helps the model to capture the intra-activity information, it creates some other challenges. First, we require a method to segment the data by recognising the points in time where an activity ends and new one begins. Second, when modeling the activities independently, we lose the inter-activity transitions.

In individual activity model configuration, to detect an activity a segment of sensor data stream corresponding to that activity is fed to each activity model. The activity model that gives out the highest likelihood value is selected as the most probable activity. The accurate segmentation of the data is a complex problem. When modeling each activity individually, the effective decoding of the activities requires the data segments to consist only instances from one activity. There is a rather simple approach taken by [12] where they segment the data with a fixed length window. This approach does not consider the start and end times of the activities and may generate segments with more than one activity. Tapia et al. [33], on the other hand, use a sliding window technique with a fixed length window equal to the average length for each activity. Although this method provides better segmentation it may be problematic for detecting instances of activities with a length different from the mean and it also requires repeated scans of the data stream for each activity. In Section 3.1 we propose a novel approach for the segmentation problem.

The inter-activity information that is not modeled in the first stage is captured at the second stage. The second stage consists of a HMM where each activity is modeled as a state, while the observation distribution is based on the first-stage output. The use of a second layer in a layered-HMM architecture is not unique to this work but the way to structure the second stage as an ‘error correction’ stage to boost the accuracy of the first stage is unique. Modelling the activities with an ‘error correction’ stage allows the model to capture the activities more closely at sensor level while keeping it immune to the effect of spurious sensor activations, that becomes more prominent when modelling the activities with low level sensor information. The second stage aims at correcting possible misclassification originated in the first stage. It performs error correction by learning the misclassification from the first stage. This stage also considers the relative temporal relationship between the activities which can improve the confidence of the model when inferring the activities.

Chapter 2

Literature Review

Over the past two decades two industries, elder care and home automation, has been seen as the driving force for the research in context aware computing. The potential of context aware ubiquitous computing to provide better living at homes and increased productivity in office spaces has drawn lot of attention in this area. In assisted living complexes, context-aware systems monitor the state of the elderly occupants, freeing the nursing staff from the task of constantly supervising them, thus giving them more time to care about those who actually need their support most. The in-home elder care and home automation requires a house to be equipped with ubiquitous sensor networks that is capable of making decisions according to the occupant's activities by learning their behavior. The inherent complexity of human behavior poses a great challenge for recognition [33]. For example, people often perform several activities simultaneously and everyday activities are subject to periodic variation. Different activities occur at different time scales. The activities can be performed with interleaving intervals, such as laundry, a person can start doing laundry and may start cooking a meal and then come back to finish laundry. This requires a system capable of identifying different activities that are being performed by the occupant in a living space with an ability to disambiguate them for proper detection.

Detection and recognition of activities is not a trivial task. Several approaches are being carried out to design a monitoring and detection system with ubiquitous sensor embedded living laboratories [5, 16, 2]. The goal of such systems is to be sensitive enough to disambiguate the abnormal activities that may be an indication of distress of an occupant in a house and at the same time robust enough to detect the activities in a error prone sensor environment. This needs a thorough study of each activity being performed by an individual to design a system that can learn the behavior of a person to correctly identify the normal activities from abnormal activities.

The emerging market of home automation promises some very attractive options for better living. Meyers and Rakotonirainy listed very interesting smart homes scenarios in [19]. Some of them are as below:

- Phones only ring in room where the addressee is present, preventing other people being disturbed by useless ringing.

- The music being played in a room adapts automatically to the people within and the pictures in the frames on the desk change depending on which person is working there.

- In-house context-aware communication systems allow family members to speak to each other as if they were in the same room, even when they are in different rooms.

- Elderly people will be supported in their daily life by context-aware homes, allowing them to age in their own home or familiar environment.

To study ADL several living laboratories have been designed such as MavHome [5], a multi-disciplinary research project at the University of Texas at Arlington focused on the creation of an intelligent home environment; AwareHome [16], aiming to create a living laboratory for research in ubiquitous computing for everyday activities; and HomeLab [2], established by Philips as a testing ground for a better tomorrow with an emphasis on advanced interaction technology. The PlaceLab project [15], a joint initiative between Massachusetts Institute of Technology(MIT) and TIAX, a collaborative product and technology development firm, aimed to provide a living laboratory to systematically study human behaviors, the routine activities and interactions of everyday life.

Existing approaches on activity recognition can be broadly classified based on types of sensors used for observing the activities.

1. Non-Visual Sensors based recognition

2. Video & Audio based recognition.

2.1 Non-Visual sensor based recognition

Given the contextual and temporal nature of human activities in a typical home setting several different approaches has been proposed for the detection of activities using the non-visual sensors. Depending on the objective of monitoring, activity recognition has been bifurcated into two branches: single and multiple occupant activity detection. The single occupant detection approaches are geared more towards the independent living of the elderly and multiple occupant approaches target other smart home applications, where multiple occupant environment is more likely.

Yang in [37] elaborated on the current research trends in the activity detection domain. He discussed how different machine learning techniques making their way in the detection of activities, such as transfer learning that is used in [24, 41] for localization of human in a building environment using wifi radio signals, inferring the cross domain activities using Transferred HMM and Skip Chain Conditional Random Field (SCCRF) models for detecting concurrent and interleaving activities. SCCRF is proposed in [11] to model the interleaving activities and *Goal Graphs* to model the concurrent activities. Although SCCRF is capable of modeling the interleaved activities, it is not clear whether they provide any significant advantage with the added complexity that is created by number of skip chain between the interleaved activities [17]. Several approaches propose a method to capture the concurrent and interleaving activities [13, 11, 25]. In [11], they demonstrated how often a person performs the concurrent and interleaving activities in a typical day.

A context aware ADL system proposed in [4] uses a variant of HMM called Adaptive Learning HMM that incorporates a self adapting mechanism to adapt to the new environment without needing to re-train the model. Another interesting approach is seen in [27] that exploits the user-object interaction to infer the activities. It uses dynamic Bayesian networks to model the activities and then Monte Carlo approximation to solve for the most likely activity. They tried to build a very generic dynamic Bayesian network that can easily adapt to different environments. This approach inherently allows the contextual information to increase the accuracy of activity detection. An interesting work was done in [18], where they equip the house with several different types of sensors including RFID switches, motion sensors, on-body accelerometer, current and flow sensors. Different aspects of activity detection were then evaluated, such as effectiveness of each sensor to detect an activity independently and in combination with other sensors, evaluating different machine learning techniques, such as Naive Bayes, Support Vector Machines (SVM) and C4.5 decision tree models. It was discovered that motion-based sensors outperformed every other sensor for most of the activities and SVM model showed the best results for activity detection.

The MavHome project in [30] proposed a predictive framework for the location-aware resource optimization in smart homes. The predictive framework uses spatio-temporal context of the activities to provide services and managing resources in a house by predicting next action of the inhabitant. Sanchez et al. in [32] proposed and tested two configurations of a 2-level layered HMM model, serial layer and parallel layer HMM models, on the multiple occupant dataset, consisting of activities performed by doctors and nurses in a typical hospital. This approach distinguishes between different activities performed by different staff members by associating the contextual information such as the usage of the artifacts and the tasks that are relevant for different staff members (nurses, physicians and interns). All the data was created manually. They modeled the first layer as low level activities abstraction layer, they used the artifact contextual information in the form of artifact usage matrix at the observation of the first layer. The output from the first stage is combined with people interaction information in the form of weight matrix and is then fed to the second layer for higher level activities in a serial configuration. In parallel configuration they modeled the artifact and people interaction in the first layer with two separate HMMs and then combine the output to feed the second layer. They reported that parallel layered HMM performed better than other configuration. Perdikis et al. in [26] also used a layered HMM model for the detection of human actions on a desk in an office environment, using on-body magnetic position sensors. They reported that the inherent advantages of using layered HMM are their capability of modeling actions of relatively long durations and reducing model complexity. Although our approach follows the same suite by modeling the activities at different levels, we use the high level activity model in a unique way, by using it as an error correction layer for the first stage and at the same time using it to capture the inter-activity transitions between the activities. Most of the work done in activity detection for multiple occupants uses an explicit ID in the form of RFIDs transmitters attached to each user to distinguish the activities performed by different users [36]. However the work presented in [10] uses Emerging Patterns (a type of knowledge pattern that describe significant changes between classes of data to model the activities of different users). A novel two phase approach for detecting abnormal human activities to provide support to the occupant of a house or other premises in case of emergency is presented in [39]. They first use one class SVM to filter out the activities with highest probabilities of being normal and then used kernel non-linear model (KNLR) to derive abnormal activity model.

2.2 Video & audio based recognition

The video-based activity recognition approaches, focus on human postures and locations to infer the activities, rather than object-human interaction. Video-based recognition can be classified into two categories [28]: offline human body modeling, where a known human body model is used and sequences of postures are learned through a probabilistic models for the recognition process and online human body modeling, where a typical human body model is not assumed a priori but variations in pose configuration, body shape, camera viewpoint and appearance are used to model the human body and infer the activities. Mori et al. in [21] used a time series of 3-D spatial human postures as input to infer the daily life actions of a human, such as walking, standing, sitting and lying. They proposed a hierarchical structure of actions to capture the fine details of human postures using continuous HMM. To capture long term complex activities involving interaction of multiple people, Zhang et al. [40] proposed a variant of HMM, which is a combination of the coupled HMM and hierarchical HMM and is called decomposed HMM that decomposes the standard HMM hierarchically with coupling between the different layers in hope to capture the interacting activities multiple processes at the same time. They used video clips of activities as input to their model and dealt with a problem of detecting the interacting activities, where two people may interact with each other in the same frame. Their approach showed only marginal improvements over HMM, coupled HMM and hierarchical HMM models with additional complexity in learning and testing phase. Audio data could potentially be a powerful source for anomaly detection. Doukas et al. in [6] used on-body accelerometers and microphones for the detection of patient activities. They showed how sound data can be effectively used for patient fall detection. They used short time Fourier transform and spectrogram analysis to extract features from sound data and then used SVM to classify the activities. Doukas et al. in [7] used both video and audio for patient activity detection and emergency recognition in cases like elder fall. Their system uses microphone array to localize and classify the abnormal sounds and then use over head video camera data with different classifiers to detect an activity or patient's situation.

Chapter 3

Methodology

Human actions over time are quantitatively captured using electronic sensors. To model the activities more accurately and make better use of the information available in the form of relative and absolute time features, we model the activities with low level sensor information at the first stage. The higher level temporal features between the activities are introduced in the second stage.

Model architecture of the proposed two-stage model is shown in Fig. 3.1. We use Markov chains to model the activities. Markov chains are suitable for the modeling of time varying discrete stochastic processes [22]. In addition, they support the inclusion of temporal and spatial characteristics of activities discussed above, either implicitly or explicitly. In this work we have adopted the Markov chain framework as described in Hasan [12].

3.1 Multi-Markov Model - First Stage (MMM-FS)

The first stage consists of a collection of Markov chains, each representing an activity. A Markov chain is a discrete random process that complies with the Markov Property and is represented as a state-space, linking each state with other in a chain like manner. The Markov property defines that when the next state of a system depends only on the current state and not on the sequence of events that preceded it. A random process with Markov property is called Markov process [22]. A Markov process is a system that randomly changes state at each time step. A series of sensor activations within an activity can be considered as a discrete-time random process and can be effectively modeled using a Markov chain.

The activities can be modeled as Markov chains at different information levels. To model the higher level temporal information, that may define the ordering of the human activities in a day as a Markov chain, the series of activities performed in a day are considered as a single process and different activities as different states of that

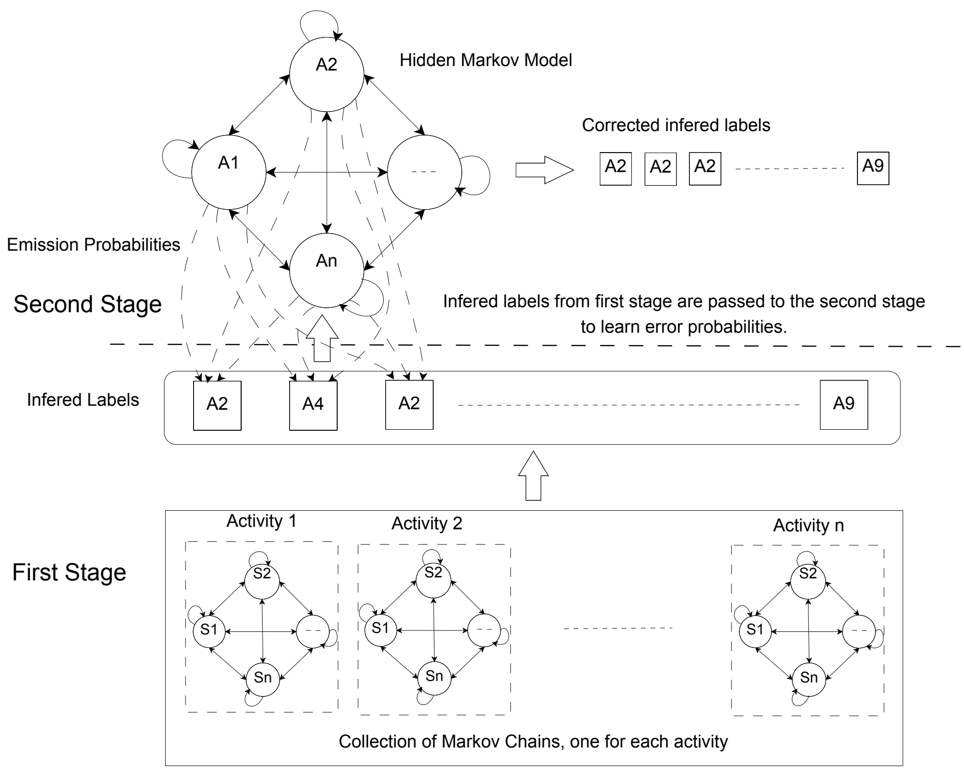


Figure 3.1: Model architecture. The first stage is a collection of activity Markov chains and the second stage is a corrective HMM that gives the final detected activities.

system that changes randomly at each time step. Although modeling the activities at higher temporal information level simplifies the modeling process it alone does not allow the complete use of the temporal features that are embedded at lower level within the activities. Each activity is composed of a series of elemental steps indicated by the sensor activations. These steps are more likely to be repeated every time the activity is performed by a person and are more dependent on the person's habit. One example of that would be every time a person prepares a tea he may fetch a spoon from a drawer to add sugar in the cup either before pouring the boiling water or after. To model the activities more accurately and make better use of the information available in the form of relative and absolute time features, we model the activities at low level sensor information in first stage. The higher level temporal features between the activities are introduced in the second stage.

An activity Markov chain consists of a set of states, $S = s_1, s_2, ..., s_n$, where each state is an active sensor that are activated during the activity. The activity process starts in one of the states depending upon the initial state distribution and steps from one active sensor state to another. Moving from the state s_i to the state s_j at the next time step with probability a_{ij} that depends only on the current state of the system (Markov property). Transition probabilities a_{ij} are modeled during the learning phase. The system could remain in the same state at next time step happens with probability of a_{ii}. The activity Markov chain is defined by three parameters:

$$MM = (S, \Pi, A) \tag{3.1}$$

where: S is the set of states, Π is the initial state distribution and A is the state transition probability matrix. Development of an activity model has two phases: learning and testing.

During the learning phase, each activity is modeled by learning the initial state Π and state transition A distribution matrices from the training samples of the corresponding activity using the following equations.

$$a_{ij} = T_{ij}/T_i \tag{3.2}$$

$$\pi_i = N_i/N \tag{3.3}$$

where: T_{ij} is the number of i to j state transitions, T_i is the number of transitions from state i, N_i is the number of activations of state i and N is the number of states.

When modeling the activities at lower level, we observe that there could be more than one sensor that is active at one time slice. This could be due to three possible reasons: some times more than one sensor can be associated with one sub-activity, more than one sub-activity is performed within one sampling period, or due to the presence of spurious sensor readings. When dealing with non-visual sensors the errors in activity representation by the sensors can be introduced very easily, one example of that is if a person forgets to shut the closet after opening it, the sensor associated with that closet door keep triggering while the person is moved on to perform another activity. This representation of the data allows natural filtering of the sensors with respect to the activity being performed, as most of the sensor that are active during an activity are more likely to be associated with that activity. Inspired from the work in [12] we use the concurrently active sensors to train the activity model by considering only active sensors as the states for that model. We dynamically add new states as they appear in successive time periods. This method is very useful in limiting the complexity in cases where there are hundreds of sensors deployed in a house. So, instead of considering all the possible sensors as states, we learn the activity models only with the sensors that are relevant to modeled activity i.e. those sensors that were active during different samples of the modeled activity. All the activities are trained individually with their corresponding training samples. While training an activity model we learn the transitions only between the active sensor states. These transitions capture the sequences of sensor activations. In testing phase, when modeling the activities individually, the typical problem of data sequence segmentation comes up. A sensor data stream is a long sequence of sensor activations during multiple activities and there is no direct way to distinguish between the sensor data corresponding to different activities during testing.

In this work we propose a novel variable window length segmentation technique called rate of change termination segmentation (MLROC). It allows the segmentation of the sensor data sequences more accurately than the sliding window segmentation approach, by monitoring the change in the likelihood value of the most likely activity model, also called forward variable at each time step of a test sensor data sequence. It is described in detail in the next section. In testing or decoding phase, a test sequence is fed to each of the activity model and the maximum likelihood of each model given the prior and state transition matrix is evaluated using the Viterbi algorithm [29]. The Viterbi algorithm reduces the problem of finding the most likely state sequence into finding a shortest path from the start state to the end state in a trellis of all the possible paths given the model parameters.

$$P_a(S_i^t) = \begin{cases} P_a(S_i)P_a(O_{oi}) & \text{if } t = 1 \\ P_a(O_{oi})max[P_a(S_i^t|S_i^{t-1})P_a(S_a^{t-1})] & \text{otherwise} \end{cases} \tag{3.4}$$

where: $P_a(S_i)$ is the prior probability of sensor i for activity a, $P_a(O_{oi})$ is the probability of output O_o from sensor i during activity a and $P_a(S_i^t|S_i^{t-1})$ is the state transition probability for activity a. The activity model with the highest maximum likelihood value of all the models is chosen as the activity for the given test sequence of the sensor data.

3.1.1 Maximum-Likelihood Rate of Change Termination Segmentation Technique (MLROC)

The Maximum-Likelihood rate of change (MLROC) segmentation technique proposed in this work is inspired by the observation that the likelihood value of the most likely activity changes substantially when the transition from one activity to another occurs. This is due to the fact that different sensor triggering patterns occur when a new activity starts and activities that have similar sensor signature may not occur in succession. Based on this observation we increment the length of the window by one time unit at each step while monitoring the rate of change percentage in the likelihood value using Eq. 3.5 where $\hat{\Theta}$ is the maximum likelihood value. When the rate of change percentage exceeds a predefined threshold value, the window is terminated and the inferred activity for that segment is used as the output. From the point of termination a new open window is started in the next time step and the process repeats until the end of the test day. It is to be noted that, unlike the sliding window approach for segmenting the data where the data is segmented corresponding to the window of a fixed length that is decided based on the average lengths of the activities is moved over the data sequence with a predefined overlapping, the use of MLROC does not require repeated scans of the test day with fixed size windows of each activity and hence is a more efficient segmentation method. The rate of change percentage (ROCP) is calculated using the Eq. 3.5 where: $\hat{\Theta}$ is the Maximum likelihood.

$$ROC = |(\hat{\Theta}(t) - \hat{\Theta}(t-1))/\hat{\Theta}(t-1)| \times 100 \tag{3.5}$$

The threshold value relies on different sensor activations during different activities, in cases where sensor errors (triggering of unrelated sensors during an activity) are very dominant in the data streams this value will be smaller as there will be relatively smaller changes at activity transitions as the unrelated sensors will be active during most of the activities. For the selection of the rate of change threshold value in cases where the original labels or ground truth is available, an alignment accuracy of the segmented data with the original labels can be used during the training phase. The alignment accuracy is measured by calculating the distance from the start and end point of the identified segments with the start and end points of the original labels. To quantify the best segmentation we designed a ad-hoc distance measure that was used to evaluate the best threshold value. The distance measure is explained in detail in Sec. 3.4. This evaluation is done over a validation set and is then used for the test set. In cases where the ground truth is not available, the threshold can be selected by analysing the changes in the rate of change percentage of the data streams during the training phase. Figure 3.2 shows an example of how the rate of change percentage changes significantly when the transition from one activity to another occurs. The x-axis shows a portion of a sequence of activities from dataset DS-2, introduced in Sec. 4.1, the y-axis represents the rate of change percentage value.

3.2 Multi-Markov Model - Second Stage (MMM-SS)

The first stage does not model the relationships between the activities executed over the day because each activity is modeled individually. The second stage of the model makes use of the relative temporal features at the activity level, into the activity detection process. Furthermore, it boosts the accuracy of ADL recognition by acting as an error-correction layer on top of the first stage. We chose HMM for the second stage, where the corrected activity process is unobservable and the detected activity process is the output from the first stage. The error-correction process using HMM works as follows:

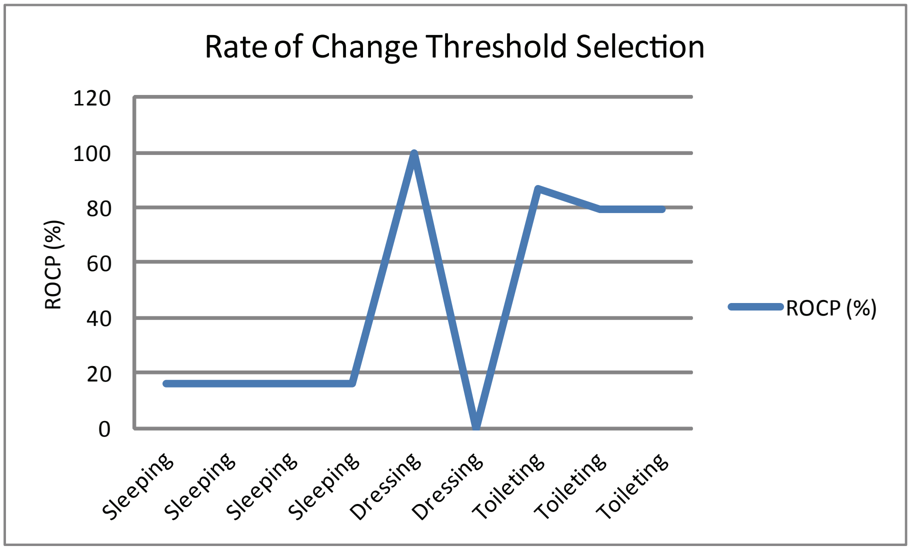


Figure 3.2: Rate of change threshold selection. The graph peaks represent the transition from one activity to another.

1. In the training phase, the corrected activity process is modeled by learning the activity transitions from the training dataset and the error probabilities from the first stage output.

2. In the testing phase, the Viterbi algorithm is used to find the corrected activity sequence using the transition for each activity and the error probabilities from the first stage model.

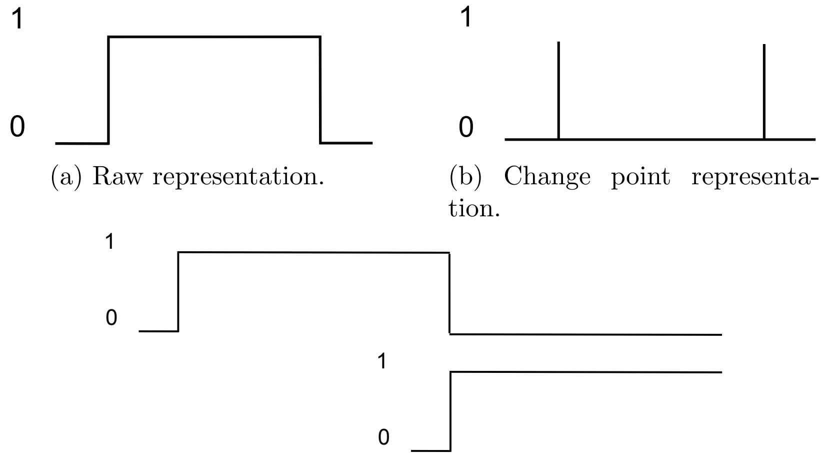
An error-correction HMM is defined by a five-tuple:

$$HMM = (S, O, \Pi, A, E) \tag{3.6}$$

where: S is the set of hidden states (activities), O is the set of observation symbols (activities learned from first stage), Π is the initial state distribution, A is the state transition probability distribution and E is the probability of activity i classified as activity j.

3.3 Data Representation

The goal of this research is to detect the activities without interrupting the natural behaviour of a person that pertains to the induction of false sensor readings into datasets. In both datasets, that are used in this work, it was observed that certain sensors continue to fire even after the activity was over. This could be due to the reasons, such as annotation errors (since the data is annotated manually), this was confirmed with manual inspection of the data, unintentional triggering of sensors by the subject or false triggering of sensors due to sensor calibration faults. For example activities that involved door of rooms or cabinets tend to be left open by the subjects, causing the sensors to fire continuously and give false impression of the activity being executed. Another example from dataset 1 Sec.4.1 the person starts preparing lunch and during this activity he triggers sensors associated with 'Electric cooker' and 'Dishwasher'. It was observed that these sensors remain active when the person moved on to the next activity which in this case was *toileting*. Although the person moved on to the toileting activity, the *preparing lunch* activity was being executed in the background. This might be one of the examples where people perform concurrent activities and for this dataset only the active activity was registered as ground truth. Therefore, to deal with the false sensor readings, annotation errors and concurrent activities the *raw* representation of the data where the sensor value when active is recorded as '1' and when inactive it is recorded as '0'. Two novel data representations called *change point* and *last sensor* representation were proposed in [35]. To reduce the sensor errors (unrelated sensors triggering during an activity) introduced in the data due to the natural behaviour of the subject or malfunctioning of the door or drawer mechanism, the *change point* representation registers the sensor activations only for the time instances where sensor changes its value. This resolves the problem of misrepresentation of the activity in cases where the doors or drawers are left open even after the activity is completed. Although this representation increases accuracy [35] it fails to capture the temporal element between the atomic activities. The *last sensor* representation, as the name suggests, registers only the last sensor that changed value and continues to register it as active until another sensor changes its value. This representation in addition to re-introducing the temporal element in the data, minimises the sensor errors issues in *raw* representation and is more suitable for models like Markov chains that capture sequential data. Kasteren [35] reported that they achieved highest accuracy when they combined the *change point* and *last sensor* representations. This may be due to the fact that *change point* representation removes the activity misrepresentation errors of the sensors from



(a) Raw representation. (b) Change point representation.

(c) Last Sensor representation.

Figure 3.3: Data representation.

the data and *last sensor* representation adds back the temporal relationship between the atomic activities. Different data representations are shown in Fig. 3.3. Although the combination strategy of the *change point* and *last sensor* representation is not clearly defined in [35], we combined the two by simple logical OR operation. In our experiments we found no difference in results between the last sensor representation and combination of *change point* and *last sensor* representation. We used the *last sensor* representation for all the experiments presented in this work.

3.4 Rate of change threshold selection

In order to achieve the best segmentation it is important to select an appropriate threshold value for the termination of the window. This is done by quantifying the alignment accuracy of each day in the validation set using the distance measure that is calculated as shown in algorithm 1. The smaller the distance, the better is the alignment accuracy. The accuracy of the alignment of segmented labels against the true labels is measured for different candidate threshold values. The candidate threshold values are selected based on the amount of activity misrepresentation errors present in the dataset. More the errors in the data, the larger is the variation between two successive instances within an activity (unrelated sensors changing values) and hence higher threshold values are required to identify the transition between activities. The threshold value with the least distance is considered to give the best segmentation for the given dataset and is used for the test set.

Due to the limited size of the datasets we use the training set as validation set for the threshold selection. We segment all days in the validation set for each candidate threshold value. For the dataset used in this work we chose the candidate threshold values between 80 % and 99.9 % with the increments of 5 %. 80 % was chosen as the starting point based on our experimentation with both the datasets. For every candidate threshold value, the alignment accuracy is calculated for each day in the validation set and then averaged over all the days, the threshold value with least average distance is selected for the test set.

Algorithm 1 Alignment Distance

Input: Vectors ST_O, ET_O, ST_S and ET_S, the start and end times of original and segmented data

Output: Alignment distance AD.

clear AD
for every day d **do**
  clear ST, clear ET
  $ET = min(length(ET_O), length(ET_S))$
  $ST = min(length(ST_O), length(ST_S))$
  $k = length(ET)$
  for every k **do**
    if $ET_O(k) == ET_S(k)$ **then**
      $dist = 0$
    end if
    if $ET_O(k) > ET_S(k)$ **then**
      $dist = sum(ET_O(k) - ET_S(ET_O > ST(k) \&\& ET_O < ET(k)))$
    end if
    if $ET_O(k) < ET_S(k)$ **then**
      $dist = sum(ET_O(k) - ET_S(ET_O > ST(k) \&\& ET_O < ET(k)))$
    end if
    $AD(d) = AD(d) + dist$
  end for
end for

Chapter 4

Results and Evaluation

4.1 Datasets

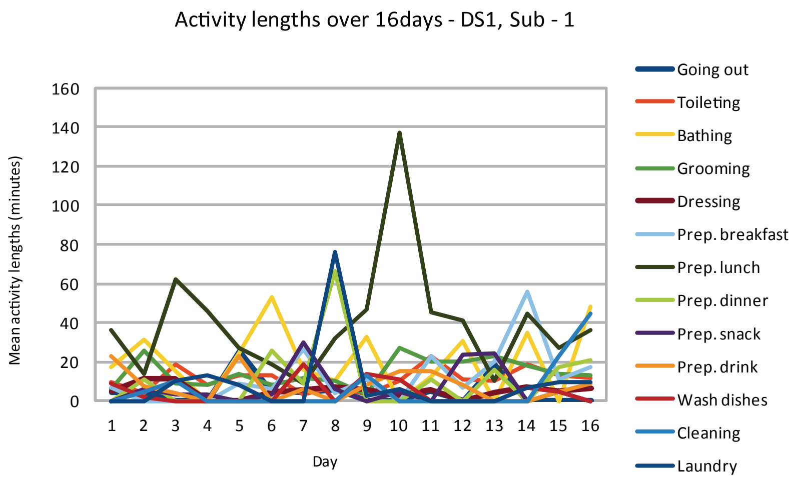
In this work we used two different datasets. Both datasets are collected from a real house set up with single occupancy. The subject is asked to perform a certain set of activities during the day. Care has been taken so that the activities are not highly orchestrated to avoid any unintended bias in the dataset. For modeling and evaluation purposes both datasets provide activity labels, which are collected manually by the subjects while performing an activity.

4.1.1 Dataset - 1 (DS-1)

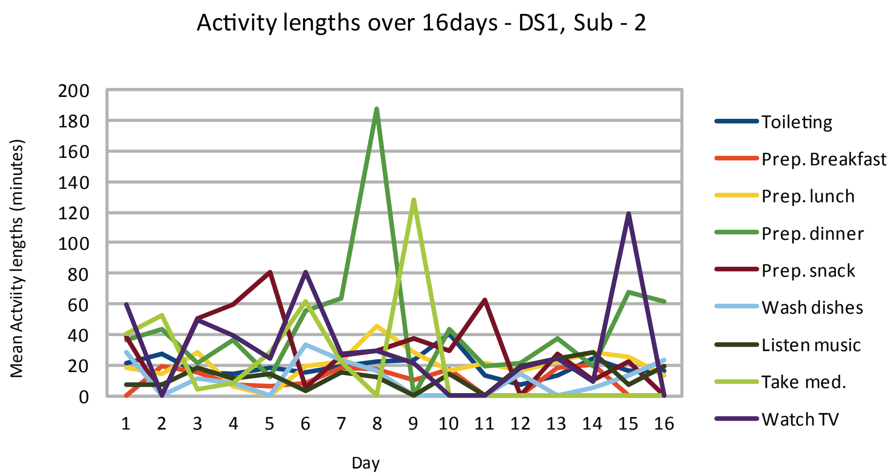
The first dataset [33] uses wireless sensor nodes. These sensor nodes can be equipped with different sensor types. This dataset consists of only reed switches that are connected to 77 and 84 collection boards. The sensors are installed in everyday objects such as drawers, refrigerator, containers. The dataset consists of the data from two different houses occupied by two different subjects. The number of instance for each activity in the dataset are shown in Table 4.1. Fig. 4.1 shows the lengths of different activities over 16 days for both subject 1 and 2. We use this dataset to compare our approach with the Multi-Naive Bayes classifier (MNBC) proposed in [33]. It can be observed from Fig. 4.1 that the lengths of each activity varies widely among different days and may not necessarily present on each day.

4.1.2 Dataset - 2 (DS-2)

The second dataset [34], consists of 21 binary sensors installed in a two storey house for 18 days. The sensors are installed in everyday objects in a house, such as reed switches in the cabinets and doors, mercury contact sensor for the movement of contacts such as drawers, float switch for the toilet flush system. The house is occupied by a 57



(a) DS-1 Subject 1



(b) DS-1 Subject 2

Figure 4.1: Activity lengths over 16 days for DS-1

year old male, who was asked to perform fifteen fixed every day activities. The activities were hand annotated by the subject in a diary. The list of activities with their corresponding number of samples can be seen in Table 4.2 and the lengths of different activities over 19 days can be seen in Fig. 4.2.

Both the datasets are provided in matlab format. There are two main files, *sensor structure* that provides the sensor activation time with location and Sensor IDs and *activity structure* that provides the start and end time of an activity with activity labels. The data for all the days is concatenated in two files. To use the dataset for

S.No.	Activities	Subject 1		Subject 2	
		Samples	Std. Dev. over 16 days	Samples	Std. Dev. over 16 days
1	Going out to work	12	0.83	-	-
2	Toileting	85	0.90	40	0.87
3	Bathing	18	0.78	-	-
4	Grooming	37	0.88	-	-
5	Dressing	24	0.94	-	-
6	Preparing Breakfast	14	0.60	18	0.81
7	Preparing Lunch	17	0.24	20	0
8	Preparing Dinner	8	0.50	14	0.63
9	Preparing a Snack	14	0.66	16	0.48
10	Preparing a Beverage	15	0.75	-	-
11	Washing Dishes	7	0.50	21	1.00
12	Cleaning	8	0.87	-	-
13	Doing Laundry	19	1.59	-	-
14	Listening to Music	-	-	18	0.56
15	Take Medication	-	-	14	1.47
16	Watching TV	-	-	15	1.37

Table 4.1: Activity samples in DS-1 and the standard deviation of the occurrence of the activities over 16 Days for subject 1 and 2.

training our models we divided the data in days at 6 AM point of the day for DS-1 to and 3 AM point of the day for DS-2 to capture the repeated patterns of daily activities as suggested in [35]. Both datasets are recorded with assumption that there is always one person at a time in the house and the all the activities are performed sequentially, however with close examination of the sensor activations in the datatsets it was found that at several occasions concurrent activities were performed but only the activities that the person was performing actively were registered as ground truth.

S.No	Activities	No. of samples	Std. Dev. over 19 Days	Percentage of time
1	Going out to work	47	1.00	45.7%
2	Toileting	89	1.27	1.0%
3	Shower	14	0.67	0.8%
4	Brushing teeth	26	0.92	0.4%
5	Go to bed	19	0.64	29.2%
6	Prepare breakfast	18	-	0.6%
7	Prepare dinner	11	-	1.1%
8	Get drink	10	0.49	0.1%
Other Activities (21.1%)				
9	Eating	27	1.04	-
10	Shave	7	0.48	-
11	Get dressed	23	0.52	-
12	Take medication	5	0.44	-
13	Prepare Lunch	8	-	-
14	Get snack	9	-	-
15	Relax	30	-	-

Table 4.2: Activity samples in DS-2 and the standard deviation of the occurrence of the activities over 16 Days.

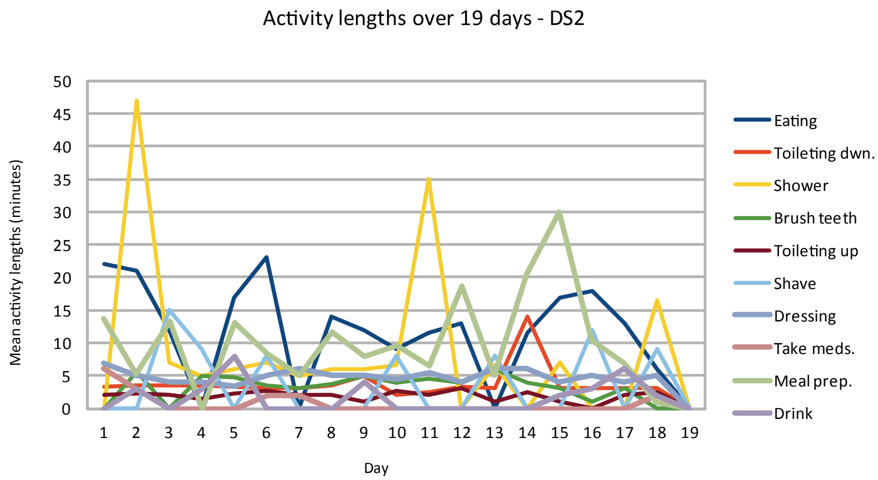


Figure 4.2: Activity lengths over 19 days

4.2 Activity Visualization Tool

In activity detection problem, understanding the nature of the activities can help to model them as closely as possible. To understand the activities we built a visualization tool that can be used for different datasets to give direct visualization of the activities as being performed in real time. This can help the experts to get a greater insight of the activities and help them to choose the appropriate modeling techniques to model the activities. To get the complete sense of activity episodes in a house, this tool requires a floor plan layout image and the physical coordinates of each sensor installation. If the physical coordinates are not available with the dataset, a unique identifier to indicate the room-wise location of the sensor can be used to display the rough estimate of activities being performed.

The design objective of an activity visualization tool was to provide the user with multiple views and enough controls to study and analyse the sensor data to understand the dynamics of the sensor activation with respect to the activities performed by the occupant of a house. A screen shot of the visualization tool is shown in Fig. 4.3. This tool is not designed for monitoring the patients in a living space. Its sole purpose is to aid an expert to get a better understanding of the activity data while designing the activity detection engine for the an activity detection system. One such example is the use of this tool by the expert to analyse the activities with respect to different

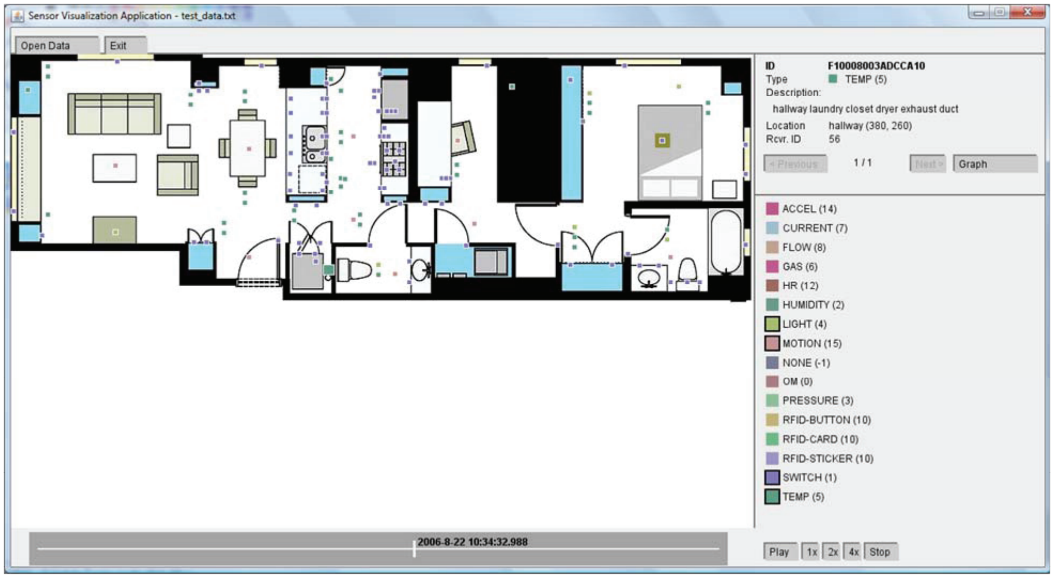


Figure 4.3: Sensor Visualization Window

sets of sensors giving a better understanding of the effectiveness of different sensors to capture the natural human activities in a living space.

For the tool to function properly, it requires a floor plan image of the house where the activities are needed to be analysed, a file containing the physical coordinates of the sensor installation scaled to the dimensions of the image and the data files with columns sensor ID, times of activations (time stamps), sensor description and values. For purpose of demonstration we used the PLCouple1 [1, 14] dataset created by PlaceLab research team. This dataset provides the floor plan image of the house and the physical coordinates of the sensor installation in the house. The *SensorVisualization* window is divided into two main sections. The *sensor view* on the left and *controls panel* on the right. The *sensor view* shows the floor plan of the house with the sensor locations in the house. The house is installed with different types of sensors, each sensor type is shown as a different color square. The activation of binary sensors such as switches or motion sensors is indicated by glowing rectangles around the sensor points as a visual cue, shown in Fig. 4.4. The data from the sensors that have continuous values over time, such as temperature, humidity, etc can be viewed on the *graph view* with respect to time for better understanding of the variation in values over time. The graph view is shown in Fig. 4.5. A user can click on each sensor point on the sensor view to read the sensor ID, type, description and its exact location on the top right corner of the *SensorVisualization* window also shown in Fig. 4.4. When the user clicks on a sensor a search action is triggered that retrieves all the fields corresponding to that sensor ID and displays them in the information box in the main window.

The sensor location file for the dataset used to demonstrate this visualization tool had many sensors that were missing the coordinate information due to the nature of the sensors such as RFID-tags. These RFID tags can be placed on any movable object, so in order to handle such situations we allow to display them with their approximate locations on the floor plan with respect to their location description. In cases where the exact coordinates of each sensor are not available we display a stack of sensors in each room. Each sensor can be accessed using the next and previous controls on the top right corner of the *control panel*. Another feature that is incorporated in the sensor view is the ability to display the sensors by type. This allows the user to select only a subset of sensors allowing them to study the effectiveness of each sensor for their intended task. Furthermore, it can provide the ability to analyse the layout of the house in terms of mobility and accessibility for the occupant in each room. A user can select the sensor types from the list displayed on the *control panel*.

Figure 4.4: Sensor Information Display.

The activity of each sensor value can be viewed on the *graph view* window, shown in Fig 4.5. The *graph view* can be easily accessed by first selecting the sensor whose value needs to viewed and then clicking on the graph button in the *control panel*. In the *graph view* window the time slider (black vertical line) indicates the sensor values on a chart with respect to the current time. For the detailed analysis of the sensor data the user can zoom in on the graph by simply dragging a selection box over the area of interest. The *graph view* window can be zoomed in 4 X the default scale to allow viewing of the finest detail in sensor values at the scale of millisecond (depending on the dataset) on the time axis. The zoom function is implemented such that the graph is expanded only across time axis (x-axis), as it allows the user to visualize the data in reference to time and prevent the view to jump with respect to the sensor value. The graph can be zoomed out in steps by right clicking anywhere on the graph screen. The graph window also shows the time duration and data values (Height) at the top right corner and updates them when zoomed in to reflect what time slot is being observed.

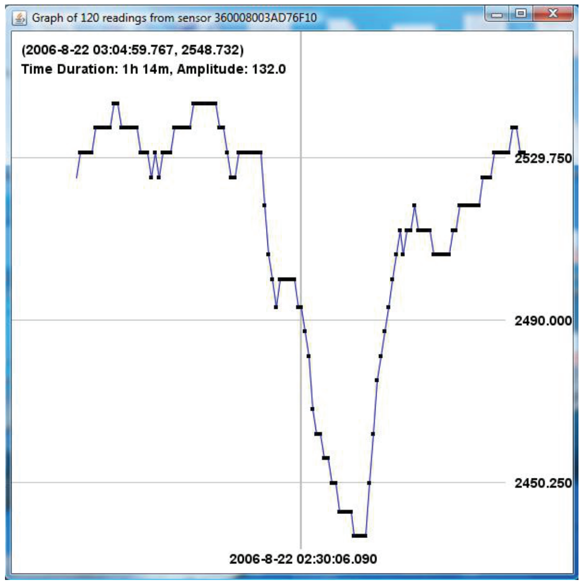


Figure 4.5: Graph view displaying the temperature sensor values over time scale. The x-axis represents time and y-axis represents amplitude.

4.3 Theoretical and Experimental Analysis

To demonstrate the performance of the proposed methodology two different sets of experiments were performed. The first experiment shows the performance of the proposed MLROC segmentation technique against a general fixed window segmentation technique, while the second shows the performance of our two-stage approach for activity detection. Furthermore, to demonstrate the effectiveness of our results we present a test of statistical significance for our approach presented in this thesis with a one stage traditional HMM.

Due to the small size of the datasets, we test our two-stage approach for activity detection on each day in a leave-one-day-out fashion. We evaluated the activities by two measures: *time slice accuracy* and *class accuracy*, these evaluation methods are commonly used in the domain of ADL detection. We present the time slice and class accuracy results for our approach in comparison with the approach presented in [33] and with traditional single stage HMM. The average class accuracy and average time slice accuracy are defined as follows:

Average Class Accuracy: For each activity, the percentage of the activity time slices identified correctly within an activity averaged over all the days in the dataset. The average class accuracy is calculated using the Eq. 4.1 where d is number of days, tp is true positive labels and fn is false negative label.

$$\text{Average class accuracy} = \frac{1}{d}\sum_{d}\frac{tp}{tp+fn} \tag{4.1}$$

Average Time Slice Accuracy: The percentage of time slices, of the correctly identified activities averaged over all the days in the dataset. The average time slice accuracy is calculated using the Eq. 4.2 where d is number of days, tp is true positive labels.

$$\text{Average Time Slice accuracy} = \frac{1}{d}\sum_{d}\frac{tp}{\text{total number of time slices}} \tag{4.2}$$

4.4 Results

4.4.1 MLROC variable window segmentation technique vs fixed window segmentation technique

This section presents the alignment accuracy results for different fixed window lengths and candidate threshold values. It can be observed from Table 4.3 that MLROC segmentation technique outperforms the fixed window technique for all candidate threshold values.

4.4.2 MNBC vs MMMC vs HMM (Leave-one-day-out cross validation)

There are thirteen and nine activities in DS-1 for subject 1 and 2 respectively. However, for subject-1 only 8 activities and for subject-2 only 7 activities are shown for

Fixed window Lengths	**2**	**7**	**10**	**15**
Alignment Distance	1590.4	520.13	390	256.1
MLROC segmentation	**80**	**90**	**95**	**99**
Alignment Distance	148.4	148.17	101.53	133

Table 4.3: Alignment distance comparison between MLROC and fixed window segmentation techniques

the purpose of comparison. These activities are selected based on the Katz ADL index [31], that is a standard measure used in healthcare to assess the cognitive and physical abilities of an elderly person.

Activities	No. of Samples	MNBC (%)	MMM-FS ROCP = 90(%)	MMM-SS ROCP = 90(%)	HMM (%)
Preparing Lunch	17(7.4%)	29	33.03	42.67	53.76
Toileting	85(37.1%)	31	44.32	42.69	72.3
Preparing Breakfast	14(6.1%)	6	37.05	23.58	48.24
Bathing	18(7.8%)	29	43.02	50.47	27.36
Dressing	24(10.4%)	3	65.43	66.79	27.67
Grooming	37(16.1%)	26	46.47	54.29	31.5
Preparing a beverage	15(6.5%)	13	21.28	23.84	16.84
Doing Laundry	19(8.2%)	7	63.37	63.15	21.05
Ave. time slice acc.	-	-	41.82	45.75	43.68

(a) Subject-1.

Activities	No. of Samples	MNBC (%)	MMM-FS ROCP = 99(%)	MMM-SS ROCP = 99(%)	HMM (%)
Prep. Dinner	14(9.5%)	30	37.42	44.1	6.23
Prep. Lunch	20(13.6%)	22	34.32	29.08	16.62
Listening to music	18(12.3%)	9	54.3	55.67	28.85
Toileting	40(27.3%)	23	32.13	36.84	70.45
Prep. Breakfast	18(12.3%)	24	13.56	16.93	56.71
Washing Dishes	21(14.3%)	11	34.36	31.32	29.92
Watching TV	15(10.2%)	16	43.39	43.15	23.22
Ave. time slice acc.	-	-	37.17	37.12	33.77

(b) Subject-2.

Table 4.4: Activity identification accuracy for DS-1. No. of samples is the total number of each activity in the dataset presented with the proportion of each activity that makes up the complete dataset. The difference in accuracy of the activities in bold letters were found to be statistically significant for MMM when compared with HMM.

For DS-1 dataset, the results in Tables 4.4 shows the class and time slice accuracy for our two-stage approach MMM-FS (first stage) and MMM-SS (Second Stage) in comparison with HMM and Multi-Naive Bayes classifier (MNBC) as discussed in [33]. The activity results are the averages of the class accuracies of each activity. Both class accuracies and time slice accuracies are averaged over all the days in the dataset. It can be observed from the results that in almost all the activities our model performs better than the MNBC. When comparing to HMM (Tables 4.4) the meal preparation activities, specifically *breakfast* and *lunch* activities are better identified with HMM for subject-1, and our approach outperforms in the remaining activities. These observations can be confirmed from Table 4.5. This indicates that even after introducing the temporal feature in MMM at the second stage, HMM was still more capable to use it effectively to distinguish between the two activities. Tables 4.5 and 4.6 present the confusion matrices for subject 1 and 2 in DS-1. The rows represent the true labels and columns represent the predicted labels by the model. Some interesting observations were made from these confusion matrices. In the case of closely related activities like *Bathing*, *Grooming* and *Dressing* which are more likely to happen in physical locations with close vicinity that may trigger a similar set of sensors our model performs much better when compared with HMM. This can be observed in Table 4.5. However from Table 4.5(b), we observed that for the *Lunch Preparation* activity our model was confused with *Breakfast Preparation* and *Beverage Preparation* activities at several instance when compared with HMM. Another observation was made that reveals the spurious activations of the sensors where the models were being confused between the *Lunch Preparation* activities with the *Toileting* activities. This was more prominent with HMM than the output from the two stages of our model. On a closer examination of the data it was found that a sensor that is labelled as 'Trash compactor' is activated in both the activities, so its activation is learned by both activity models during training phase. The time slices (one minute sample during an activity) where the sensor associated with 'Trash compactor' is activated alone are predicted as *Toileting* activity, even though it was found from the true labels that the person was performing the *Lunch Preparation* activity in that time slice. This was also found to be the case for the several instance of *Beverage Preparation* activities in the dataset. For subject-2, during the activity *watching TV* several unrelated sensors trigger. It seems that although the TV is 'on' the person is not actively watching the TV but doing other activities that includes toileting at several instances. For *Listening music* we observed the same behaviour but this behaviour of unrelated sensor triggering is expected because while listening music people generally perform other activities at the same time.

To find how significant are the improvements of our proposed approach compared to the model presented in [33] and single stage traditional HMM we performed a test of statistical significance. Due to the presence of a highly variable distribution of activity length and occurrences among the different days, it was not possible to prove that the differences in mean classification accuracy between MMM and HMM are statistically significant. However, we were able to perform a test of statistical significance (Analysis of Variance and Dunnett multiple comparison tests) per activity over all the days in the dataset. We first performed Analysis of Variance test over the average class accuracies of each activity for both the models to find whether they are significantly different. After finding the MMM and HMM model accuracies significantly different the Dunnett multiple comparison test was executed to find which model performed better. We found certain activities, namely *grooming*, *dressing* and *doing laundry* in subject 1 and *preparing dinner*, *Watching TV* in subject 2 showed significantly better accuracy than HMM with at most 0.05 probability error. Effects of this variability in the datasets is also observable in cases where the accuracy degrades from first stage to second stage. This behaviour is due to the fact that not all the activities are present in all of the days of the datasets and the second stage, being a single model representation for all the activities, in some cases incorrectly labels the activities with the activities that are not present in the test day.

For MNBC we were not able to perform the same multiple comparison test because of the unavailability of the results over different days, but we perform a Paired t-test on the means of the classification accuracy of the activities over all the different days in subject 1 and 2 as shown in Table 4.4. We found that differences in the mean of each activity between MNBC and MMM (first and second stage), and MNBC and HMM are statistically significant with at most a 5 percent probability error.

For DS-2, Table 4.7 shows the average class and time slice accuracy for Multi-Markov model and the traditional HMM. DS-2 is a cleaner dataset with lower number of sensor errors comparing to DS-1, resulting in higher overall accuracies for each activity. There are a total 15 activities in DS-2, which we collapsed to 12 by merging the meal preparation activities that are often confused with one another, such as *preparing breakfast*, *lunch* and *dinner*. For DS-2 we were able to compare our model only against HMM as results for MNBC were not available on this dataset. We did not find any significant improvements with our approach when compared with traditional

HMM. This may be due to the reason that the DS-2 is crafted to incorporate high level relational temporal information, making it more prominent in the dataset, hence the advantages with our approach are not very visible in the results.

	Toileting	Bathing	Grooming	Dressing	Breakfast Prep.	Lunch Prep.	Beverage Prep.	Laundry
Toileting	596	12	58	10	8	29	77	6
Bathing	280	123	33	4	4	4	10	4
Grooming	304	0	138	25	6	1	12	0
Dressing	84	9	6	36	0	3	10	6
Breakfast Prep.	10	0	36	9	56	66	52	1
Lunch Prep.	202	7	4	10	53	337	55	10
Beverage Prep.	96	0	6	0	10	11	41	0
Laundry	151	5	8	8	2	22	7	42

(a) Confusion Matrix Subject-1 HMM

	Toileting	Bathing	Grooming	Dressing	Breakfast Prep.	Lunch Prep.	Beverage Prep.	Laundry
Toileting	363	28	123	53	31	84	49	65
Bathing	107	206	61	23	12	32	11	10
Grooming	87	35	222	54	30	29	16	13
Dressing	11	7	19	97	4	10	3	3
Breakfast Prep.	12	15	37	17	61	67	20	1
Lunch Prep.	60	31	12	10	188	187	142	48
Beverage Prep.	21	5	25	6	15	59	25	8
Laundry	21	17	5	2	8	11	6	175

(b) Confusion Matrix Subject-1 MMM-FS

	Toileting	Bathing	Grooming	Dressing	Breakfast Prep.	Lunch Prep.	Beverage Prep.	Laundry
Toileting	361	26	137	43	9	62	91	67
Bathing	97	241	42	24	1	13	37	7
Grooming	66	40	251	48	25	25	22	9
Dressing	8	12	25	95	13	0	1	0
Breakfast Prep.	11	13	42	22	38	103	0	1
Lunch Prep.	57	23	7	10	87	236	222	36
Beverage Prep.	24	2	30	10	14	40	35	9
Laundry	21	9	7	7	1	13	5	182

(c) Confusion Matrix Subject-1 MMM-SS

Table 4.5: Confusion matrices for DS-1, Subject-1. Each number in the matrices represents the number of time slices.

	Toileting	Breakfast Prep.	Lunch Prep.	Dinner Prep.	Dish Wash	Listen Music	Watch TV
Toilteting	200	69	90	91	63	105	21
Breakfast Prep.	27	74	179	60	83	33	10
Lunch Prep.	65	117	277	132	96	69	87
Dinner Prep.	32	51	117	191	35	46	34
Dish Wash	25	13	57	40	110	37	46
Listen Music	71	58	63	50	86	372	18
Watch TV	131	63	167	68	60	105	570

(a) Confusion Matrix Subject-2 MMM-FS

	Toileting	Breakfast Prep.	Lunch Prep.	Dinner Prep.	Dish Wash	Listen Music	Watch TV
Toilteting	236	36	79	119	50	98	21
Breakfast Prep.	47	87	150	74	88	15	5
Lunch Prep.	112	102	211	171	101	57	89
Dinner Prep.	85	0	98	221	25	45	32
Dish Wash	22	9	54	48	101	52	42
Listen Music	106	35	53	48	81	377	18
Watch TV	156	32	152	66	62	100	596

(b) Confusion Matrix Subject-2 MMM-SS

Table 4.6: Confusion matrices for DS-1, Subject-2. Each number in the matrices represents the number of time slices.

Activities	HMM	MMM-FS ROCP = 90	MMM-SS ROCP = 90
Leaving house	84.75	84.68	84.64
Eating	72.07	86.32	83.76
Toilet downstairs	56.03	53.51	50.64
Shower	69.90	81.71	78.17
Brush teeth	46.48	0.00	36.70
Toilet upstairs	40.27	2.61	8.60
Shave	12.70	43.06	33.56
Sleeping	92.79	98.74	98.70
Dressing	68.42	43.38	52.28
Take medication	40.00	60.00	60.00
Meal Prep.	70.21	57.76	70.01
Drink	12.50	11.98	12.50
Time slice Accuracy	87.14	89.06	89.38

Table 4.7: Activity identification accuracy for DS-2

Chapter 5

Conclusion and Future Work

ADL detection is a very challenging problem. The high variability in the nature of activities, sensor errors and annotation inaccuracies, all contribute to the complexity of modeling the ADL effectively. To assist the experts of the domain to study the different aspects of ADL we designed and presented an activity visualization tool. This visualization tool can be used to assess the different qualities of the datasets and hence it aids the expert in getting an intuitive understanding of the properties of the dataset and in tuning the algorithms appropriately.

To overcome the modeling complexity of the activities due to their highly variable nature, we proposed a two-stage approach to decompose the problem into two levels. The first stage models the activities at their atomic level and the second stage attempts to correct the misclassified information from the first stage by learning the inter-activity transitions from the data and the error distribution from the first stage output.

We demonstrated that our approach gives a higher activity identification accuracy than Multi Naive Bayes Classifier in almost all test cases. We observe that for certain activities, namely *grooming*, *dressing*, *doing laundry*, *preparing dinner* and *watching TV* our proposed two-stage approach performed significantly better than traditional HMM. The experiments also suggest that, although decomposing the activities helps to improve the detection accuracies in most cases, it can also degrade detection accuracy for certain activities. Finally, we proposed a novel segmentation approach that automatically provides better segmentation of the data and hence allows better classification. We emphasize that the correct segmentation of the data is a crucial step in our method to achieve high accuracy.

Our two-stage approach with Markov models outperforms MNBC and traditional HMM in most of the cases and it is on par with HMM in the remaining cases. It would be worth exploring if modeling the duration of an activity along with other temporal features may help improve the activity detection accuracy. The activity duration may provide another level of classification, by providing a distinguishing factor between the long and short duration activities. One way to model the activity duration is by considering the activities as semi-Markov processes [20]. The semi-Markov model allows the duration of an activity (sojourn time of the process) to be learned explicitly during the training phase of the model, while still capturing the other temporal features in a typical Markov chain. The effectiveness of the Hidden Semi-Markov models (HSMM) compared to HMM has been shown in [34]. We believe using HSMM in a two layer architecture will further enhance the detection accuracy of the model. Furthermore, we would like to explore the idea to learn the threshold value for the proposed segmentation approach, during the learning phase of the model.

Bibliography

[1] Placelab couple 1 dataset, http://architecture.mit.edu/house-n/data/placelab/placelab.htm.

[2] *Ubiquitous Computing: Smart Devices, Environment and Interactions*. Wiley, 2009.

[3] Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Hierarchical hidden markov models with general state hierarchy. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 324–329. AAAI Press, 2004.

[4] B.-C. Cheng, Y.-A. Tsai, G.-T. Liao, and E.-S. Byeon. HMM machine learning and inference for activities of daily living recognition. *The Journal of Supercomputing*, 54(1):29–42, 2009.

[5] D.J. Cook, M. Youngblood, III Heierman, E.O., K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: an agent-based smart home. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 521 – 524, march 2003.

[6] C. Doukas and I. Maglogiannis. Advanced patient or elder fall detection based on movement and sound data. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 103 – 107, 30 2008-feb. 1 2008.

[7] Charalampos Doukas, Ilias Maglogiannis, Aggelos Rouskas, and Aristodimos Pneumatikakis. Emergency incidents detection in assisted living environments utilizing sound and visual perceptual components. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '09, pages 14:1–14:7, New York, NY, USA, 2009. ACM.

[8] T V Duong, H H Bui, D Q Phung, and S Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, 1(c):838–845, 2005.

[9] M. J. F. Gales and S. J. Young. The theory of segmental hidden markov models. Technical report, 1993.

[10] Tao Gu, Zhanqing Wu, Liang Wang, Xianping Tao, and Jian Lu. Mining emerging patterns for recognizing activities of multiple users in pervasive computing. In *Mobile and Ubiquitous Systems: Networking Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International*, pages 1 –2, july 2009.

[11] Derek Hao Hu, Sinno Jialin Pan, Vincent Wenchen Zheng, Nathan Nan Liu, and Qiang Yang. Real world activity recognition with multiple goals. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 30–39, New York, NY, USA, 2008. ACM.

[12] M.K. Hasan, H.A. Rubaiyeat, L. Yong-Koo, and L. Sungyoung. A reconfigurable HMM for activity recognition. In *10th International Conference on Advanced Communication Technology*, pages 843–846, 2008.

[13] Xin Hong, Chris D. Nugent, Steven Devlin, Maurice D. Mulvenna, Jonathan G. Wallace, and Suzanne Martin. Dynamic multi-component based activity detection and recognition within smart homes. In *Proceedings of the 2011 international workshop on Situation activity & goal awareness*, SAGAware '11, pages 77–82, New York, NY, USA, 2011. ACM.

[14] S S Intille, Kent Larson, E Munguia Tapia, Jennifer S Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. *Pervasive Computing*, 3968:349–365, 2006.

[15] Stephen S Intille, Kent Larson, J S Beaudin, J Nawyn, E Munguia Tapia, and P Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1941, 2005.

[16] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, pages 191–198, London, UK, 1999. Springer-Verlag.

[17] Eunju Kim, Sumi Helal, and Diane Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9:48–53, January 2010.

[18] B. Logan, J. Healey, M. Philipose, E.M. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *9th International Conference on Ubiquitous computing*, pages 483–500. Springer-Verlag, 2007.

[19] Sven Meyer and Andry Rakotonirainy. A survey of research on context-aware homes. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, ACSW Frontiers '03, pages 159–168, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[20] Charles J. Mode and Gary T. Pickens. Computational methods for renewal theory and semi-markov processes with illustrative examples. *The American Statistician*, 42(2):pp. 143–152, 1988.

[21] T. Mori, Y. Segawa, M. Shimosaka, and T. Sato. Hierarchical recognition of daily human actions based on continuous hidden markov models. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 779 – 784, may 2004.

[22] J.R. Norris. *Markov Chains*. Cambridge Universtiy Press, 1998.

[23] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, ICMI '02, pages 3–, Washington, DC, USA, 2002. IEEE Computer Society.

[24] Sinno Jialin Pan, James T. Kwok, Qiang Yang, and Jeffrey Junfeng Pan. Adaptive localization in a dynamic wifi environment through multi-view learning. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1108–1113. AAAI Press, 2007.

[25] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, pages 44–51, Washington, DC, USA, 2005. IEEE Computer Society.

[26] S. Perdikis, T. Dimitrios, and M.G. Strintzis. Recognition of humans activities using layered hidden markov models. In *Cognitive Information Processing Workshop*, pages 114–119, 2008.

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

[28] Ronald Poppe. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst.*, 108:4–18, October 2007.

[29] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pages 257–286, 1989.

[30] A. Roy, S.K. Das Bhaumik, A. Bhattacharya, K. Basu, D.J. Cook, and S.K. Das. Location aware resource management in smart homes. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 481 – 488, march 2003.

[31] Katz S. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. *J. Am. Geriatiric Soc.*, 1983.

[32] D. Sanchez, M. Tentori, and J. Favela. Hidden markov models for activity recognition in ambient intelligence environments. In *Current Trends in Computer Science, 2007. ENC 2007. Eighth Mexican International Conference on*, pages 33 –40, sept. 2007.

[33] E.M. Tapia, S.S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *2th International Conference on Pervasive Computing*, pages 158–175. Springer, 2004.

[34] T. van Kasteren, G. Englebienne, and B. Kröse. Transferring knowledge of activity recognition across sensor networks. In *8th International Conference on Pervasive Computing*, pages 283–300. Springer, 2010.

[35] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *10th International Conference on Ubiquitous Computing*, pages 1–9. ACM, 2008.

[36] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, January 1992.

[37] Qiang Yang. Activity recognition: linking low-level sensors to high-level intelligence. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 20–25, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[38] Juan Ye, Adrian K. Clear, Lorcan Coyle, and Simon Dobson. On using temporal features to create more accurate human-activity classifiers. In *Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science*, AICS'09, pages 273–282, Berlin, Heidelberg, 2010. Springer-Verlag.

[39] J. Yin, Q. Yang, and J.J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082 –1090, aug. 2008.

[40] Weidong Zhang, Feng Chen, Wenli Xu, and Zisheng Cao. Decomposition in hidden markov models for activity recognition. In *Proceedings of the 2007 international conference on Multimedia content analysis and mining*, MCAM'07, pages 232–241, Berlin, Heidelberg, 2007. Springer-Verlag.

[41] Vincent Wenchen Zheng, Sinno Jialin Pan, Qiang Yang, and Jeffrey Junfeng Pan. Transferring multi-device localization models using latent multi-task learning. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*, pages 1427–1432. AAAI Press, 2008.