HUMAN ACTIVITY RECOGNITION WITH WIRELESS SENSOR NETWORKS USING MACHINE LEARNING

by

Hande Alemdar

B.S., Computer Engineering, Boğaziçi University, 2004

M.S., Computer Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2015

HUMAN ACTIVITY RECOGNITION WITH WIRELESS SENSOR NETWORKS USING MACHINE LEARNING

APPROVED BY:

Prof. Cem Ersoy (Thesis Supervisor)	.....
Assist. Prof. Bert Arnrich	.....
Prof. Ayşın Baytan Ertüzün	.....
Assoc. Prof. Ali Taylan Cemgil	.....
Assoc. Prof. Hazım Kemal Ekenel	.....

DATE OF APPROVAL: 13.02.2015

ACKNOWLEDGEMENTS

I would like to express my deepest gratitudes to my thesis supervisor, life mentor and role model Cem Ersoy. I always admire his incredible power for leading high quality research while maintaining a perfect level for recreation. This thesis would be impossible without his generosity in sharing his wisdom, vision, food, drink, and pretty much everything else.

I would like to thank my core jury members, Ayşın Baytan Ertüzün and Ali Taylan Cemgil for taking this journey with me over the years, providing the most constructive comments. I also would like to thank Bert Arnrich, Hazm Kemal Ekenel and Albert Ali Salah for their invaluable comments and suggestions.

I would like to express my deepest gratitudes to my colleagues who have been generous with their help, inspiration, and encouragement. Especially, I would like to thank Tim van Kasteren for all the "why"s, "how"s and "when"s, Halil Ertan for embracing the torture in the form of research, Özlem Durmaz İncel for always being there to make sure everything is on track, Can Tunca for making me feel like a mentor and being a stress ball lately. I owe special gratitude to Atay Özgövde for being so considerate, Aykut Yiğitel for the French practices and flight anxiety discussions, Orhan Ermiş for the coffees, Serhan Daniş for excellent architectural drawing skills, Bilgin Koşucu for all the weird edible stuff and being a great travel buddy, Gökhan Remzi Yavuz for all the collaboration we could achieve between his two consecutive appearances.

I would like to thank all the past and present members of the CMPE family with whom I exchanged even a little bit of smile accompanied by some coffee and a little chat. They are the main reason I linger around so long. I would like to single out Lale Akarun, Ufuk Çağlayan, Tuna Tuğcu, Suzan Üsküdarlı, Pınar Yolum Birbil. I look up to them with great respect and admire their inexhaustible energies.

I would like to thank my family for making this happen with their unconditional love and support throughout my life. Lastly, I would like to take the opportunity for an official acknowledgement for my better half Serdar. Thank you for all “write your thesis”advices. I did, but, none of these would matter without you by my side.

This thesis has been partially supported by Scientific and Technical Research Council of Turkey (TÜBİTAK) under the grant number 108E207, by Bogazici University Research Fund (BAP) under the grant numbers 6370, 8684, 5146, 6056, 5344 and by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

ABSTRACT

HUMAN ACTIVITY RECOGNITION WITH WIRELESS SENSOR NETWORKS USING MACHINE LEARNING

Recognizing human behavior in an automated manner is essential in many ambient intelligence applications such as smart homes, health monitoring applications and emergency services. In order to make such long term health monitoring systems sustainable, we need smart environments in which the human activities are recognized automatically. In order to infer the human behavior, we can use machine learning methods on the data collected from the smart environments but those methods require annotated datasets to be trained on. Recording and annotating such datasets are costly since they require time and human effort. Moreover, the complex nature of human activities makes it difficult to accurately model them. While hierarchical models can be a remedy for more accurate representation, finding suitable complexity levels is not a trivial task. Finally, when we deploy automatic human behavior monitoring systems on a world-wide scale, we need to fine tune the model behavior for each new house to accurately reflect the residents' behavior for that specific house. Rather than annotating a dataset consisting of several weeks of data, an algorithm can be used to decide for which point in time it would be most informative to obtain annotation in order to minimize the need for annotation and maximize the usefulness of annotation. This thesis addresses the above mentioned issues by (i) collecting publicly available benchmark datasets, (ii) proposing a methodology for incorporating a hierarchy into the model that is tailored for various activities individually, (iii) improving the ways of evaluating different approaches and models considering the domain specific needs, (iv) handling multi-resident environments in an unobtrusive manner and, (v) using active and semi-supervised learning techniques in order to reduce the annotation effort in large scale deployments.

ÖZET

KABLOSUZ ALGILAYICI AĞLAR İLE MAKİNE ÖĞRENMESİ KULLANARAK İNSAN AKTİVİTESİ ANLAMA

Otomatik insan davranışı tanıma, akıllı evler, sağlık izleme uygulamaları ve acil durum servisleri gibi birçok çevresel zeka uygulaması için önemlidir. Sağlık izleme sistemlerini sürdürülebilir yapmak için insan aktivitelerinin otomatik olarak algılandığı akıllı ortamlara ihtiyaç vardır. İnsan davranışlarını anlamak için, akıllı ortamlardan toplanan veriler üzerinde makine öğrenmesi yöntemlerini kullanabiliriz ancak bu yöntemler işaretlenmiş eğitim kümelerine ihtiyaç duyarlar. Bu kümeleri oluşturmak insan çabası gerektirdiğinden pahalıdır. Ayrıca, insan faaliyetlerinin karmaşık yapısı, onları doğru bir şekilde modellemeyi zorlaştırır. Hiyerarşik modeller daha doğru temsil için bir çare olabilir, ancak uygun karmaşıklık düzeylerini bulmak kolay değildir. Son olarak, otomatik insan davranışı izleme sistemlerini dünya ölçeğinde uygulanabilir kılmak için model davranışını her farklı evin sakinlerinin davranışlarını yansıtacak şekilde ayarlamak gereklidir. Her ev için haftalarca eğitim kümesi toplamaktansa, zaman içinde sadece en çok bilgi içeren noktalar için etiket toplayarak, etiketleme eforunu azaltırken öğrenme yönteminin yararlılığını artıracak bir mekanizma geliştirilebilir. Bu tezde, (i) tüm araştırmacılara açık, karşılaştırma amacıyla kullanılabilir veri kümeleri oluşturarak, (ii) makine öğrenmesi modelinde, her aktiviteye özel olarak hiyerarşi seviyesi belirlemek için bir yöntem önererek, (iii) farklı yaklaşımların ve modellerin değerlendirilmesini, alanın özel ihtiyaçlarını gözetecek şekilde geliştirerek, (iv) evde birden fazla kişinin yaşadığı durumları kullanıcılara ek yük getirmeyecek şekilde ele alan yöntemler önererek, (v) geniş ölçekli kurulumlarda etiketleme eforunu azaltmak için aktif ve yarı-denetimli öğrenme teknikleri kullanarak, yukarıda bahsedilen konuları hedef alan çalışmalar yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	State transition distribution
$\mathcal{A}$	Set of activities
B	Observation probability distribution
C	Correctly classified occurrences
D	Deletion errors
$\mathcal{D}$	Dataset
E	The number of chains in FHMM
f_t	Finished state variable at time t
F	Fragmented errors
F'	Fragmenting errors
FM	Fragmented and merged errors
FM'	Fragmenting and merging errors
I'	Insertion errors
K_a	The number of sequences for activity a
$\mathcal{L}$	Labeled dataset
m	The number of parameters in the model
M	Merged errors
M'	Merging errors
n	Sequence length for an occurrence sequence
N	The number of sensors
N_p	The number of particles
o	Single occurrence of an activity in the dataset
o_n	First layer output of TWNN
O_a	Start overfill errors
O_ω	End overfill errors
$\mathcal{O}$	All occurrences of an activity in the dataset
Q	The number of states
R_m	Measurement model noise variance for particle filter

R_p	Process noise variance for particle filter
s_h	Output of the hidden unit h
sc^*	Minimum model score
T	Length of the sequence
$\mathcal{T}$	Training dataset
$\mathcal{U}$	Unlabeled dataset
U_a	Start underfill errors
U_ω	End underfill errors
w_i	Weight of the particle i
W	Half window size for TWNN
$\mathbf{W}$	Mean matrix for FHMM model
x_t	Observation vector at time t
x_t^i	The value of i^{th} sensor value at time t
y_t	State at time t
z_t	Action state at time t
α	The probability of movement in random walk
δ_{ij}	Kronecker delta function
$\delta(.)$	Dirac delta measure
Δt	Discretization interval
θ	The set of model parameters for HHMM
μ_{ij}	Bernoulli parameter for the i^{th} sensor for the j^{th} state
$\boldsymbol{\nu_n}$	Second layer weights for the output n
π	Initial state probability distribution
Σ	Covariance matrix for FHMM model
ϕ_f	Binomial distribution for the finished state variable f
χ	The set of model parameters for HMM
ψ	The set of model parameters for FHMM
$\boldsymbol{\omega_h}$	First layer weights for the hidden unit h
Ω	Total feature size for TWNN

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAL	Ambient Assisted Living
ANN	Artificial Neural Network
ARAS	Activity Recognition with Ambient Sensing
AIC	Akaike's Information Criteria
BFS	Breadth First Search
BIC	Bayesian Information Criteria
CASAS	Center for Advanced Studies in Adaptive Systems
CVL	Cross Validated Likelihood
DBN	Dynamic Bayesian Network
DT	Decision Tree
EAD	Event Analysis Diagram
ECG	Electrocardiography
EEG	Electroencephalography
EM	Expectation Maximization
FHMM	Factorial Hidden Markov Model
FN	False Negative
FP	False Positive
FSR	Force Sensitive Resistor
GPS	Global Positioning System
HHMM	Hierarchical Hidden Markov Model
HMM	Hidden Markov Model
ICL	Integrated Complete Likelihood
iHMM	Infinite Hidden Markov Model
iid	Independent and Identically Distributed
JPDA	Joint Probabilistic Data Association
KNN	K Nearest Neighbors

MC	Monte Carlo
MHT	Multiple Hypothesis Tracking
MLP	Multi Layer Perceptron
NBC	Naive Bayes Classifier
PAN	Personal Area Network
pdf	Probability Density Function
PF	Particle Filter
PIR	Passive Infra Red
PML	Penalized Marginal Likelihood
RFID	Radio Frequency Identification
SMC	Sequential Monte Carlo
SIR	Sequential Importance Resampling
SIS	Sequential Importance Sampling
SVM	Support Vector Machine
TDNN	Time Delayed Neural Network
TN	True Negative
TP	True Positive
TWNN	Time Windowed Neural Network
WSN	Wireless Sensor Network

1. INTRODUCTION

Recognizing human behavior in an automated manner is essential in many ambient intelligence applications such as smart homes, health monitoring and assistance applications, emergency services, and transportation assistance services [1]. It is foreseen that in the near future, smart environments that interact with the people according to their specialized needs will be become an inseparable part of daily life. Since the global increase in the ratio of the elderly population is already prominent, the *aging in place* gained utmost importance. It is possible to relieve the economic effects of global aging by enabling the elderly to stay active and healthy for longer years in their own homes where living independently is more natural and comfortable [2]. When a caretaker or a relative lives with an elderly or disabled person, health state changes can easily be detected since they are indicated by the changes in the activities of daily life, for example changes in the eating or sleeping behavior. Unfortunately, the growing population of the elderly make it prohibitive to assign a human caretaker for all homes with elderly residents. The need for self managing health in partnership with health care providers is inevitable. For that reason, ambient assisted living (AAL) systems which enable relatives and health personnel to monitor everyday behavior of the elderly living alone are needed. In order to make such long term health monitoring systems sustainable, we need smart environments in which the human activities, hence the human behavior are recognized automatically [3, 4].

During the past decade, the advances in the sensor technology and wireless communication networks in terms of capacity increase, cost efficiency and power efficiency made it possible to use sensors for human activity recognition purposes. These miniaturized sensors are soon to be deployed in large scale and produce vast amount of data. As the data supply increases, the demand for techniques to process such a huge amount of data in order to extract useful information in a reasonable amount of time also increases. In order to meet this demand, we need data-driven methods that are easily applicable to novel settings. In order to infer the human behavior, we can use machine learning methods on the data collected from the smart environments but those

methods require annotated datasets to be trained on. Recording and annotating such datasets are costly since they require time and human effort. Although the annotated datasets are essential, they are hardly useful when recorded in laboratory settings following predefined scenarios since they do not reflect the natural human behavior. Besides, the evaluation of several inference methods in order to find the optimal performance in terms of behavior recognition for healthcare purposes requires metrics and methodologies beyond the ones that are available for general use.

Moreover, the complex nature of human activities makes it difficult to accurately model them. While hierarchical models can be a remedy for more accurate representation, finding suitable complexity levels is not a trivial task. The diversity in human activities in terms of duration, interactions with the environment and the differences in the order of the actions that makes up the activity make the problem even more complicated. For example, an activity like *preparing breakfast* might consist of several actions such as 'turning on the coffee maker', 'turning on the toaster' and 'getting cheese out of the fridge'. The order of these actions may change for different occasions of the same activity or some of the actions may completely disappear. Conversely, *sleeping* activity may not contain so many actions although it typically lasts for several hours. In order to correctly model the human activities, the correct complexity and hierarchy levels should be determined.

Finally, when we deploy automatic human behavior monitoring systems on a world-wide scale for healthcare purposes, we need to fine tune the model behavior for each new house to accurately reflect the residents' behavior for that specific house. In order to accomplish that, annotated data from that house is needed. Rather than annotating a dataset consisting of several weeks of data, an algorithm can be used to decide for which point in time it would be most informative to obtain annotation. The system can prompt the resident and ask which activity is currently being performed. This would minimize the need for annotation and maximize the usefulness of annotation.

This thesis addresses the above mentioned issues by (i) collecting publicly avail-

able benchmark datasets, (ii) proposing a methodology for incorporating a hierarchy into the model that is tailored for various activities individually, (iii) improving the ways of evaluating different approaches and models considering the domain specific needs, (iv) handling multi-resident environments in an unobtrusive manner and, (v) using active and semi-supervised learning techniques in order to reduce the annotation effort in large scale deployments.

1.1. Research Overview and Contributions

In this thesis, we concentrated on human behavior identification in smart environments using interaction based sensing. We addressed the multiple person human activity recognition problem as opposed to the current state-of-the-art which mostly concentrates on the single resident case. Considering the domain specific needs of the human behavior monitoring for health assessment purposes, we used machine learning in order to model and recognize activities of daily living in an accurate and efficient manner. Moreover, we addressed the scalability issues arise when we deploy these systems on a world-wide scale. We summarize the contributions of this thesis as follows:

- *Activity Recognition with Ambient Sensing (ARAS) datasets with multiple residents:* ARAS human activity recognition datasets are collected from two different real houses. Each house was equipped with 20 interaction-based binary sensors of different types that communicate wirelessly using a low power ZigBee communication protocol, which enables the sensors to have longer battery lifetimes. A full month of information which contains both the sensor data and the activity labels for both residents was gathered from each house, each with two residents, resulting in a total of two months data. The datasets are made public so that the community can develop and benchmark novel methods' performances under realistic conditions. [5, 6].
- *Hierarchical hidden Markov model (HHMM) with a variable number of states per activity:* Human behavior contains rich hierarchical structure and previous work has shown that modeling this structure can benefit the recognition of human activities from sensor data. However, the added complexity that a hierarchy brings

can make the construction of an accurately fitting hierarchical model challenging, while the additional layers of representation can require additional annotation efforts for supervised learning methods. Our proposed model uses a semi-supervised learning approach to automatically cluster the inherent structure of activities into actions so that we can remain agnostic about the interpretation of the actions that the learning method allocates. The only design consideration is the number of states used to represent the actions that make up each activity. For this purpose, we propose using three different model selection mechanisms: Akaike's information criteria (AIC), and Bayesian information criteria (BIC) and cross-validated likelihood (CVL) [7].

- *Behavioral performance monitoring:* The performance of newly developed inference methods for more accurate behavior recognition are evaluated using metrics widely used in machine learning domain such as accuracy, precision, recall and F-measure. Although these metrics are solid, they may fail to reveal the actual performance in terms of behavior understanding. Human behavior is characterized by frequency and duration of several activities. In order to evaluate the performance in terms of behavior recognition, we need to define metrics that are suitable for the specific needs of human behavior. We propose a two level evaluation mechanism in order to reveal the actual performance at the application layer [8].
- *Handling multiple resident activity recognition:* We focus on making smart houses smart enough to provide long term health monitoring for not only people who live alone but also with a spouse or a flat mate. In that respect, we need to recognize behavior individually in multi-resident environments without assuming any person identification which generally requires the use of wearable technology that can be obtrusive. We propose two different methods for handling the multiple resident case. First, we use nonlinear Bayesian tracking for decomposing the observation space into two, secondly we directly model the overlaid observations together with multiple chains of activity sequences using a factorial hidden Markov model (FHMM) model [9].
- *Active learning with uncertainty sampling:* Human behavior recognition meth-

ods in smart interactive environments depend on both the environment and the people, therefore the models and the parameters are subject to change across different environments and different people. In order to deploy these systems on a large scale, we need to relearn the parameters for each setting. Moreover, even for the same setting, they are subject to change over the course of the time. This change can stem from a variety of reasons such as the changes in the behavior of the people, changes in the environment or changes in the sensor behaviors. Learning the parameters for every different setting from scratch is not feasible since it requires large amount of annotated data which is hard to obtain. Instead, we can use active learning to select the most informative data points for annotation. By requesting annotation only for the most informative data points, we reduce the amount of training data needed and minimize the annotation effort. [10, 11].

1.2. Thesis Outline

Chapter 2 presents a review of the state-of-the-art human activity recognition systems. Contributions of the thesis are presented in Chapters 3-7.

Chapter 3 presents the ARAS datasets, sensor and activity selection strategy, details of the data collection phase together with design criteria and lessons learnt from real world deployments. We also provide benchmark results and insights on the activity recognition performance on the datasets.

Chapter 4 describes the hierarchical model that allows having different model sizes for different activities and three different model selection strategies together with experimental evaluation results on multiple publicly available data sets.

In Chapter 5, we present a new evaluation mechanism for evaluation of two different classifiers' performance on multiple datasets.

The methods for handling the multiple residents in smart environments without assuming any explicit identification are explained in Chapter 6. This chapter presents

our two different approach to the problem together with an extensive experimental evaluation on ARAS datasets.

Chapter 7 presents an active learning scheme based on uncertainty sampling in order to reduce the annotation efforts in new settings. We evaluate the performance of three different uncertainty measures using real world deployment scenarios with ARAS datasets.

Although each chapter has a separate conclusion section, in Chapter 8, we present an overall conclusion and discussion of the contributions of the thesis.

2. STATE OF THE ART ON SENSOR-BASED HUMAN ACTIVITY RECOGNITION

Given the importance and promise of automatic human activity recognition, there has already been a significant research effort on the subject during the last decade. In this chapter, we provide an overview to the recent approaches to human activity recognition problem. More comprehensive literature survey on sensor-based human activity recognition can be found in [12]. In [1], we provide a more detailed review on the wireless sensor networks that are used for healthcare purposes specifically.

In terms of sensor-based human activity recognition, there are two main tracks in terms of sensor deployment strategies. The first approach is deploying the sensors in the environment, making them *ambient* and mostly stationary. In the second track, the sensors are carried by the humans and they are *mobile* or nomadic. The studies using ambient sensing are further categorized into three subcategories. To begin with, we observe a computer vision based human activity recognition domination in the ambient track. Computer vision based systems have a longer history in human activity recognition mainly because of the security and surveillance applications in public space. The use of video cameras in private environments such as smart homes for healthcare purposes raises privacy concerns and therefore it not likely to be widely accepted by the inhabitants. Instead, the use of miniaturized sensors that can measure the conditions of the environment and the interactions of the inhabitants with it. This second branch of ambient sensing has started in early millennium and expanded quickly due to the advancements in the sensor and communication technology and having less privacy related problems. There is also an increasing trend in acoustic sensing of the activities in smart environments since the sound contains rich information about the environment and the activities performed. Besides, speech is a natural way of interacting and communication. Understanding the speech and ambient sound is beneficial for many healthcare applications especially in the remote monitoring cases.

On the mobile sensing track, the penetration of smart phones that have abundance of functionality together with sensing capability made it possible to use them for human activity recognition purposes as well. Although the mobility is one of the main advantages of smart phone based sensing that enables us to expand to outdoor environments as well as indoors, it brings added complexity in activity recognition which in turn leads to challenges in processing of the data on a battery operated limited capacity device. More recently, the unprecedented growth of the wearable devices gave rise to a whole new track of well-being applications that require automatic recognition of activities. This quantified-self paradigm, that has originally started with simple accelerometer-based sensing of the activity levels, has expanded quickly to include many other physiological signs such as heart rate, blood pressure, and oxygen saturation levels. In order to meet the demand, the research efforts on wearable sensing have also exponentially increased in the last couple of years. An overview of the classification of sensing based human activity recognition literature is given in Figure 2.1. Although the general groups are prominent, there are several studies at the intersections which use combinations of the technologies.

In the following sections, we provide an overview of the state-of-the-art for each main branch of activity recognition research separately. Due to the large collection of studies in each group, we focus only on the recent trends.

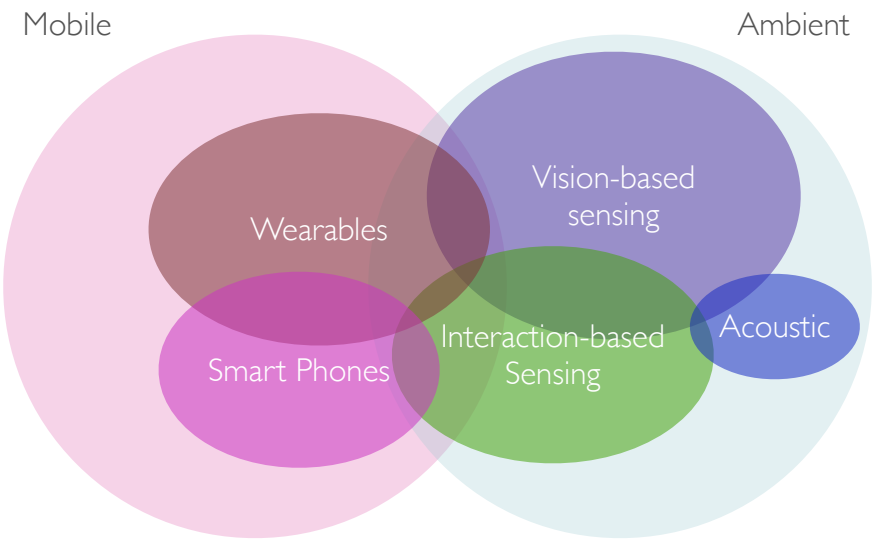


Figure 2.1. Classification of sensing based human activity recognition studies.

2.1. Ambient Sensing

2.1.1. Vision-based Human Activity Recognition

Computer vision based human activity recognition systems have used a wide variety of camera setups using a single camera, multiple cameras, stereo vision, infrared or thermal cameras and a wide variety of methods ranging from single layered space-time based approaches to multi-layered description based approaches. There are already a number of extensive literature surveys on vision-based human activity recognition [13–17] and there are a number of publicly available datasets that the computer vision community used for benchmarking purposes. In [18], a detailed review of the available datasets is given.

In the recent years, the research on human activity recognition using vision-based sensing has moved from two dimensional (2D) towards three dimensional (3D) with the emergence of cameras providing the depth information. Especially, with the introduction of the Microsoft Kinect sensor [19], the single and direct 3D imaging devices have become widespread and commercially available at low costs. The reduced costs and ergonomic form factors of depth video sensors have made human activity recognition realizable for elderly monitoring applications in homes. In [20], a depth-based life logging system is designed to recognize the daily activities of elderly people. Initially, a depth imaging sensor is used to capture depth silhouettes. Based on these silhouettes, human skeletons with joint information are produced which are further used for activity recognition and generating their life logs. The life-logging system is divided into two phases. During the training phase, the researchers collected a dataset using a depth camera, extracted features and trained a hidden Markov model (HMM) for each activity separately. In the second phase, the recognition engine classified the learned activities and produced life logs. The system was evaluated using life logging features against principal component and independent component features and achieved satisfactory recognition rates on the smart indoor activity datasets.

Using a single camera makes such systems more easily deployable avoiding the

difficulties inherent to classical stereo and multi-view approaches such as the correspondence problem, careful camera placement and calibration. On the other hand, full-volume 3D data contracted by multiple views contains more detailed information as opposed the simple frontal surfaces of humans and other objects provided by depth cameras. Additionally, these sensors are usually limited to a range up to about seven meters, and the estimated data can become distorted by scattered light from reflective surfaces. In [21], the authors perform a qualitative comparison of several approaches using two different datasets. According to their results, the methods using 3D representations of the data turn out to outperform their 2D counterparts. The main strength of multi-view setups is the high quality full-volume 3D data, which can be provided from 3D reconstruction by shape-from-silhouettes and refinements techniques. It also helps to uncover occluded action regions from different views in the global 3D data, and allows for extraction of informative features in a more rich 3D space, than the one captured from a single view. However, although the reviewed approaches show promising results for multi-view human pose estimation and action recognition, 3D reconstructed data from multi-view camera systems has some shortcomings. First of all, the quality of the silhouettes is crucial for the outcome of applying shape-from-silhouettes. Hence, shadows, holes and other errors due to inaccurate foreground segmentation will affect the final quality of the reconstructed 3D data. Second, the number of views and the image resolution will influent the level of details which can be achieved, and self-occlusion is a known problem when reconstructing 3D data from multi-view image data, resulting in merging body parts. Finally, 3D data can only be reconstructed in a limited space where multiple camera views overlap.

Khan *et al.* [22] present a video based system for detecting abnormal activities for the elderly home care applications. The proposed method is validated by a dataset consisting of six abnormal activities: forward fall, backward fall, chest pain, faint, vomit, and headache. The purpose of their research is recognizing abnormal activities from normal daily life activities in order to identify the potentially emergent situations. The system uses a single camera to recognize activities from two view points by using a transform technique followed by kernel discriminant analysis. Their system demonstrates an average recognition rate of 95.8% for the six abnormal activities.

In [23], the authors use the joint angles from a 3D model of a human body as opposed to conventional approaches in which the joint angles are computed from inverse kinematic analysis of the optical marker positions captured with multiple cameras. Their approach estimates the body joint angles directly from time-series activity images acquired with a single stereo camera by co-registering a 3D body model to the stereo information. The estimated joint-angle features are then mapped into codewords to generate discrete symbols for HMM of each activity. With these symbols, each activity is trained through the HMM, and later, all the trained HMMs are used for activity recognition. The performance of the joint-anglebased method were compared to that of a conventional binary and depth silhouette-based methods, producing better results in the recognition rate, especially for the activities that are not discernible with the conventional approaches.

Several researchers addressed the replicability of the research results on datasets that are recorded in more uncontrolled conditions. Amiri *et al.* [24] introduced a new dataset for human actions in a smart home environment which is specifically collected for helping us to analyze human actions in a home environment. Using this dataset, they studied the performance of some existing human action recognition algorithms, which had shown excellent performance on other simple datasets. Their experimental results show that the complexity and variations of this dataset make action recognition more challenging that it proved to be when using simple datasets. The low performance of the tested human action recognition algorithms on this dataset suggests revisiting the action recognition problem for smart home applications. Cheng *et al.* [25] propose a home activity summary system by highlighting two challenging problems in a real world application. First, the amount of data for different activity categories is extremely unbalanced, which severely degrades the classifying performance. Second, peoples activities are usually accompanied by other people such as a walking nurse nearby. It is impractical to predefine and label all the possible activities of all the potential visitors. With a technique called subspace naive-Bayesian mutual information maximization, they divide the feature space into a number of subspaces and allows the kernel and normalization parameters to vary between different subspaces. They also propose a feature filtering technique to reduce the effects of the interest points

that belong to other people. To evaluate the proposed activity summary system, they recorded a senior home activity recognition dataset and performed activity recognition for eight different categories.

Rather than recognizing the activities per se, describing the scenes in video sequences in sentences is another recent research area. For this purpose, Romdhane *et al.* [26] introduced a probabilistic framework for handling the uncertainty in a description-based activity recognition approach. This approach allows the flexible modeling of composite events with complex temporal constraints that are natural in human activities. They use probability theory to provide a consistent framework for dealing with uncertain knowledge for the recognition of complex events. They validate the event recognition accuracy of the proposed algorithm on real-world videos.

2.1.2. Acoustic Human Activity Recognition

The ambient sound can give an idea about the activities performed in an environment. There are several studies trying to identify the activities using acoustic information. To begin with, AuditHIS system which performs real-time sound analysis from eight microphone channels in a smart home is presented in [27]. The evaluation of AuditHIS in different settings showed that audio modality is very promising to acquire information that are not available through other classical sensors. Audio processing also has the potential of providing a natural way of interactions between people and the smart environment. First results reported by the study are promising, giving a 72% correct classification rate on the data gathered from volunteers in a real health smart home environment. The corpus dataset from this study is also made public recently [28]. In the dataset, there exists four scenario based interactions from 12 different users. The activities involved preparing and having a meal, sleeping, initiating and having a talk with a relative, and listening to the radio.

Stork *et al.* [29] propose an online method called non-Markovian ensemble voting in order to classify multiple human activities in a bathroom and kitchen context. Their algorithm does not need a silence detection or audio stream segmentation. Moreover,

the method can deal with activities that are extended over undefined periods in time. The method is based on learned soundbooks of activity classes and the recognition was achieved by scoring the votes from short-duration audio frames that are cast in a consistent way with respect to the learned model. According to the results of the experiments in real environments, the method can recognize 22 different sounds that correspond to a number of human activities with a recognition rate of 85% in a continuous activity recognition setting.

In [30], a multi-modal human activity recognition system that utilizes both the video and audio signals is presented. The audio corpus collected by the authors contains five spoken commands and 12 non-speech acoustic events for different types of humans activities. They also defined a set of alarming speech and audio events ("Help", "Problem", "Cry", "Cough", "Fall", "Key/object drop"), which can be a signal on a critical situation. The recognizer of speech and non-speech audio events is based on HMMs modeling and calculates Mel-frequency cepstral coefficients from multi-channel audio signals. According to the results, the lowest accuracy was observed for the non-speech audio event "Fall" with 60% recognition rate. About 30% of the occurrences are confused with the "Steps". The overall recognition accuracy of speech and acoustic events were 96.5% and 93.8%, respectively.

Hollosi *et al.* [31] propose a method for detecting coughs with a binary output. Then, the labels are fed to an event modeling scheme to determine information about the reoccurrence, the strength and the duration of the event within a given time interval. They also developed a rule-based emergency classification model for long-term monitoring and the surveillance of the progression of an event over a longer period of time. If a potentially dangerous event is identified, a message is generated to inform medical personnel.

In [32], a mobile system is presented for using outdoors as well as indoors. The system utilizes the environmental background sound which is considered as a rich information source for identifying both individual and social behaviors. Through understanding individual activities, social interaction, and group dynamics of crowds can be

deduced. The researchers use wearable devices with sound recognition capability and they attack two major challenges: limited computation resources and a strict power consumption requirement. They use a single dimensional Haar-like sound feature with HMM classification in order to achieve high recognition accuracy with low power requirement. The experimental results indicate an average recognition accuracy of 96.9 % has been achieved when testing with 22 typical environmental sounds related to personal and social activities. It outperforms other commonly used sound recognition algorithms in terms of both accuracy and power consumption. In a similar study, the authors explores semi-supervised learning options for audio-based mobile activity recognition [33]. They tested the approaches on seven users with a total data of 14 days and up to nine daily context classes. Experimental results indicate that the semi-supervised model can improve the recognition accuracy up to 21% but is still significantly outperformed by a fully supervised model on user data.

2.1.3. Interaction-Based Sensor Human Activity Recognition

The idea of using interaction-based ambient sensors for home automation in an intelligent way was first presented in the late 90s [34]. The studies that use those sensors for activity recognition purposes started in the early millennium. The GatorTech smart house was built by University of Florida for research on ambient assisted living [35]. The house contained several smart appliances equipped with sensors such as a smart refrigerator in order to monitor food usage. A similar project called AwareHome was developed by Georgia Institute of Technology [36]. They used several ceiling mounted cameras and radio frequency identification (RFID) tags for localization purposes. These projects are among the first examples of living laboratories and they aimed developing a proof of concept.

In terms of activity recognition purposes, one of the pioneering studies is the House_n project developed by Massachusetts Institute of Technology. Tapia *et al.* [37] installed reed switches and piezoelectric switches on doors, windows, cabinets, drawers, microwave ovens, refrigerators, stoves, sinks, toilets, showers, light switches, lamps, some containers and electronic appliances in two different houses in order to detect

more than 20 activities. The collected data was labeled by the subjects using software running on a personal digital assistant, was processed using a naive Bayes classifier and revealed a performance of 25% to 89%, depending on the evaluation metric used.

Several researchers used RFID for detecting the interactions with the environment through the object use. With this approach, activities are recognized based on the information provided by RFID readers which informs whether a specific tag is present or not in the environment [38–42]. RFID-based systems require residents to either wear a portable RFID reader on their bodies or wearing special RFID tags. Either way, additional burden on the inhabitants is brought besides the higher electromagnetic exposures. For this reason, low-power systems that can measure the interactions without the additional burden have become more popular.

In [43], van Kasteren *et al.* deployed a wireless sensor network (WSN) based system consisting of 14 sensors in a real house and collected data for 28 days. The data were automatically labeled by the subject using a Bluetooth headset with voice recognition software. The deployment targeted the classification of seven activities, and the data was processed using both HMM and conditional random field (CRF). They reported an accuracy of 79.4%. Kasteren datasets were expanded to include three different houses and they were among the first to take the activity recognition research from laboratory settings to real houses [44].

The Center for Advanced Studies in Adaptive Systems (CASAS) datasets were presented in [45]. 15 different activities were monitored using a smart home testbed, which was equipped with motion and temperature sensors, as well as analog sensors that monitor water and stove burner use. The system was tested in a multi-resident environment, where two students lived together. In total, CASAS contains 11 separate sensor event datasets collected from seven physical testbeds. Using this dataset, an evaluation study has been conducted to compare the performance of a naive Bayes classifer (NBC), an HMM, and a CRF model. The result of recognition accuracy using threefold cross validation over the dataset is 74.87%, 75.05%, and 72.16% for the NBC, HMM, and CRF, respectively [46].

In [47], a smart home monitoring application for assisted living was introduced. The system monitors the use of electronic appliances with current sensors, the water usage with water flow sensors and the bed usage using a force sensor for determining the sleeping pattern of the elderly. The collected data is transmitted to a central server, and if abnormal situations, such as excessive water usage, occurs the system informs the related people. A prototype of the system was deployed in a two-bedroom house with six sensors. However, no activity recognition performance results were presented in the paper. Similarly, in [48], well-being conditions of the elderly based on the usage of household appliances are monitored using ZigBee-based wireless sensors. Current sensors monitor the use of electric appliances, force sensors were attached to the bed, couch, toilet and dining chair to monitor their daily usage and contact sensors were attached to the grooming cabinet and fridge to monitor the opening and closing of the doors. Two wellness functions are defined according to the use of house appliances and their inactivity. The system was deployed in four houses with six sensors for a week and collected data in real time about the wellness of the elderly.

In another recent study [49], the use of hybrid models was proposed for increasing the accuracy of activity recognition. The authors combined the artificial neural network (ANN), specifically multi-layer perceptron (MLP) and support vector machine (SVM), with HMM and show that hybrid models achieve better recognition performance compared to MLP, SVM, decision tree (DT), k-nearest neighbors (KNN) and a rule-based classifier. They used five different datasets, including three datasets from Kasteren *et al.* [44] and two datasets collected by the authors that included 12 different activities.

Fatima *et al.* proposed a unified framework for action prediction besides activity recognition in [50]. An SVM-based kernel fusion method was utilized for activity recognition and identifying the significant sequential activities of the inhabitants to predict the future actions. CRF was used as a classifier for predicting the future actions. The performance of the kernel fusion method was compared with other kernel methods, including linear kernel, radial basis function kernel, polynomial kernel and MLP kernel, and it was shown that a 13.82% increase is achieved in the accuracy on

average for recognized activities. For action prediction, the performance of CRF was compared with HMM, and it was shown that an increase of 6.61% to 6.76% is achieved in the f-measure with CRF.

A recent literature survey of state-of-the-art AAL frameworks, systems and platforms to identify the essential aspects of AAL systems was provided in [51]. Their review revealed that only 12 projects out of many continued their projects beyond the pilot phase and deployed their solutions into the real world, either at care facilities or private homes. Their findings indicate that the scalability issues and the reusability of the knowledge obtained previously should be addressed in the following studies.

2.2. Mobile Sensing

2.2.1. Wearable Sensors

The most widely used wearable sensor modality for activity recognition is the accelerometry. Bao and Intelle were among the first to built such system using five accelerometers placed on the knee, ankle, arm, and hip in order to recognize 20 activities, including ambulation and daily activities such as scrubbing, vacuuming, watching television, and working at the computer [52]. All the collected data were labeled by the user in a home environment. They used several time and frequency-domain features with a C4.5 decision tree classifier. According to their results, ambulation activities were recognized with 95% of accuracy but other activities such as stretching, scrubbing, riding escalator and riding elevator were often confused giving an overall accuracy of 84%.

Recently, a European Union project named Opportunity was proposed with the aim to develop a new methodology for activity recognition that will remove the constraints such as static assumptions on sensor availability, placement and characteristics [53]. During the project, a dataset that use of wearable accelerometers together with video cameras was collected in a breakfast scenario. In total, there were 72 sensors of ten modalities. 12 different people performed a predefined drill of activities such as

opening/closing a drawer, cupboard or the fridge, cleaning table, moving cups, etc. However, 19 different sensors placed on the subject's body made the overall system quite obtrusive.

Many of the studies using accelerometers recognizes activities with distinctive acceleration patters only such as walking, sitting, running, standing, etc. These activities are excellent for determining the activity levels of the people as shown in many studies [54–56], but they do not convey enough information about the activities of daily living since they cannot separate *eating* from *reading* with high accuracy for example. On the other hand, wearable technology can offer a wide range of physiological sensing modalities in order to measure blood pressure, heart rate, body temperature, skin conductance, electroencephalography (EEG), electrocardiography (ECG), and respiration rate. These additional information when combined with the activities of daily living patterns offer a richer view of health status of the individuals. A more detailed review of the literature can be found in two recent surveys are given in [57, 58].

2.2.2. Smart Phones

Smart phone related human activity recognition emerged with an increasing trend during the past decade. Instead of placing additional sensors on the people, exploiting the sensors that are already embedded into the smart phone devices that we carry around all day is more practical. Although the smart phones are equipped with several sensors such as compass, Global Positioning System (GPS) sensors, microphones, camera, light, proximity, together with accelerometers and gyroscopes, the accelerometers are the most widely used sensors for activity recognition purposes.

Kose *et al.* [59] proposed a system working on Android platforms that supports online training and classification using only the accelerometer data for classification. The proposed clustered KNN method exhibited 92.27% in terms of f-measure on mobile platforms with limited resources for recognizing *running*, *walking*, *standing*, and *sitting* activities. In [60], the authors extract spectral features using dyadic wavelet transform and build a codebook using vector quantization to cluster and discretize the feature

vectors. The codebook is then used by an HMM for each activity. According to their results, the average accuracy for six locomotion activities (jogging, walking, upstairs, downstairs, sitting, standing) was 96.15%.

While many of the related studies only consider similar locomotion activities, there are some studies that considers more complex activities. In [61], the authors investigate the ability to recognize complex activities, such as cooking, cleaning, with a smart phone. According to their experiments, simple activities were be easily recognized but the performance of the prediction models on complex activities as low as 50% in terms of accuracy.

Durmaz *et al.* compiled a survey study on the activity recognition on smart phones recently [62]. According to the survey, location- and motion-associated activity recognition are the two dominating types of activity recognition using mobile phones. Besides, there are other applications for sportive activities such as bicycling, soccer, nordic walking, rowing, or for daily activities such as shopping, using a computer, sleeping, going to work, going back home, working, and having lunch, dinner, or breakfast. Also, there are applications that are using mobile phones for detecting emergency situations such as falls.

One of the major problems of activity recognition on smart phones is the orientation. Since the users can carry the devices in different positions, such as in the pocket, in the bag, or in their hands, accurate activity recognition even for the simplest activities becomes a challenge. For that reason several researchers focus on position independent and position dependent classification models [63]. In [64], the authors propose a calibration methodology combining accelerometer and GPS for handling the phone location and orientation variability. The calibration method was shown to reduce the walking speed estimation error at the individual level by 8.8% on average.

Similar to the wearable sensing, activity recognition on smart phones offers a complementary solution rather than a complete one for activities of daily living. In indoor environments such as homes and offices, the mobile phones can only give little

cues about the activities being performed. On the other hand, an indoor activity recognition system does not provide any information on the outdoors activities. Hence, a combination of both methods will provide a more complete view of the daily activity patterns of people and will be more desirable.

2.3. Publicly Available Human Activity Recognition Datasets

The research efforts on activity recognition in smart homes can be categorized into two groups. In the first group, there are studies where hundreds of sensors and sensor equipped home appliances are deployed in smart laboratory houses [35, 65–67]. Those studies generally focus on smart human interactions with the future smart environments and do not necessarily have an activity recognition for healthcare purposes focus. The second group of studies focuses on human activity recognition for health status monitoring [37, 43, 49, 68]. During the last decade, there have been a couple efforts on collecting datasets for human activity recognition in smart homes. Although, these datasets are important for the research community there are very few annotated datasets since they are harder to obtain because of the costly annotation procedure. Besides, naturalistic datasets that are collected in real houses rather than laboratory settings are even rarer.

In Table 2.1, we summarize the main attributes of the most widely used publicly available datasets together with ARAS datasets which we collected as part of this doctoral study. Most of the earlier studies consider a single resident situation. While collecting ARAS datasets, we relaxed that assumption and collected the data from

Table 2.1. List of publicly available annotated smart home datasets.

Dataset	**# of Houses**	**Residents**	**Duration**	**# of Sensors**	**# of Activities**	**Activity Occurrences**
ARAS [5]	2	Multi	60 days	20	12-14	658 - 1281
CASAS [68]	7	Multi	2-8 months	20 - 86	11	37 - 1513
Kasteren [43]	3	Single	58 days	14 - 21	10 - 16	200 - 344
Ordonez [49]	2	Single	35 days	12	10 - 11	250 - 495
House_n [37]	2	Single	14 days	77 - 84	9 - 13	176 - 278

multi-resident homes. We focused on making future houses smart enough to provide long term health monitoring for not only people who live alone but also with a spouse or a flat mate.

3. ARAS DATASETS

In this chapter, we present the architectural details of the proposed WSN-based AAL system used in ARAS dataset collection. We also explain the challenges related to the sensor selection/deployment, networking and data collection and present the respective solutions that we have devised in order to provide design criteria and guidelines for different components of multimodal WSN-based AAL systems, with the intention of also assisting future research.

3.1. Sensor Selection and Deployment

In our deployment, we used the Arduino [69] platform together with the Xbee [70] transceiver modules, which use the ZigBee protocol, to enable the sensing and wireless communication components. The Arduino platform is an open source, cost- and power-efficient hardware platform, which helped us to quickly prototype the different sensor modalities that we required for the AAL system.

We have deduced several criteria for the selection of different sensor types suitable for AAL applications. These criteria guided us through both the general and activity-specific sensor selection and deployment decisions and allowed us to overcome challenges related to the robustness and efficiency of the individual sensors and the overall system.

The foremost decision regarding the use of ambient sensors rather than wearable sensors stems from the possible concerns of the potential system users regarding obtrusiveness. Wearable sensors that are directly attached to the body or clothes are not a viable choice, since they may be uncomfortable, intrusive and even limit the bodily movements of the users. Privacy is also a significant concern that we addressed. We avoided the use of cameras, video recorders or microphones, since such devices pose a direct threat to the daily life privacy of the users. These strict guidelines ensure that the proposed AAL system is privacy-preserving and unobtrusive.

The ambient sensor devices available in our inventory include force sensitive resistors (FSRs), photocells, digital distance sensors, sonar distance sensors, contact sensors, temperature sensors, infrared receivers and humidity sensors. FSRs produce readings inversely proportional to the changing resistance according to the force applied to it. The photocells are sensitive to the change of the amount of light in the environment. Digital distance sensors measure object presence in small ranges within 10 cm. Sonar distance sensors can measure the presence of objects at higher distances, up to seven meters. Contact sensors produce readings according to the contact of their two separable components. Temperature sensors measure the environmental temperature. The humidity sensors measure the relative humidity of the environment.

The specific choice among the different types of ambient sensors is influenced by three primary criteria: targeted activity, robustness and efficiency. To assess the performance of different sensors with respect to these criteria, we have conducted experimental dry runs with the individual sensor devices under various activity scenarios before the actual deployment of the system. Moreover, we interviewed the residents about the usage of the goods and items at their homes to assess their individual interaction patterns. Such interviews enabled us to make more accurate decisions on the type and deployment location of the sensors and to better match particular activities with the sensors. The performance and convenience of the specific types of sensors with respect to the above-mentioned criteria are summarized in Table 3.1. In the following subsections, we elaborate on the details about these criteria and provide example scenarios.

3.1.1. Targeted Activity

Since the proposed AAL system's ultimate aim is to recognize the activities of the residents using the data coming from the deployed sensors, we primarily decided on the sensor types and their locations by matching them with the targeted activities. For instance, during the teeth brushing activity, we expect several actions to occur at the same time or in succession. For instance, to recognize this activity, we might deploy a contact sensor on the bathroom door, a photocell in the bathroom cupboard and a

Table 3.1. General sensor selection criteria for smart homes.

Sensor Type	**Location**	**Targeted Activity**	**Robustness**	**Efficiency**
FSR	Under bed	Lying, sleeping	High	High
	Under couch	Sitting, lying	High	High
	Under chair	Sitting	Low	Medium
Photocell	In drawer	Kitchen activities	High	High
	Cupboard/Wardrobe doors	Bathroom activities, changing clothes	High	High
Digital distance	Back of chair	Sitting	Medium	High
	Toilet seat cover	Bathroom activities	Medium	High
	Above water tap	Bathroom/kitchen activities	Medium	Medium
Sonar distance	Walls	Activity related to presence in a room	High	Medium
Contact	Regular door	Activity related to leave/entering room/house, showering	High	High
	Sliding door	Showering, changing clothes	Medium	Medium
	Drawer	Bathroom/kitchen activities	Low	Medium
Temperature	Above oven	Cooking	High	Medium
	Near stove	Cooking	Medium	Low
Infrared	Around TV	Watching TV	High	Medium
Humidity	Near shower sink	Showering	Medium	Low
Pressure Mat	On bed	Lying, sleeping	Medium	High
	On couch	Sitting, lying	High	High
	On chair	Sitting	Medium	Medium
Vibration	In drawer	Kitchen activities	Medium	Low

digital distance sensor above the water tap. The teeth brushing activity is expected to be performed as follows. Firstly, the person closes the bathroom door, which triggers the contact sensor, which is located on the side of bathroom door. It continues firing during the activity, since we expect the bathroom door to be closed during the activity. After a while, we expect the photocell located in the cupboard to be activated when the subject opens the cupboard door to get the toothbrush. As the cupboard door is

closed, we expect the photocell to stop firing. After the person has finished brushing his/her teeth, we expect the digital distance sensor above the water tap to fire for a short time when the subject is washing his/her mouth. Finally, we expect the contact sensor at the bathroom door to stop firing as the bathroom door is opened and the teeth brushing activity ends. It should be noted that the sensor selection for this scenario is done based on our intuitive belief on the succession of actions related to this targeted activity. To further increase the compatibility of the sensors to the targeted activities, we conducted short interviews with the residents to assess the ways they perform the targeted activities. For instance, they stated that they always keep the toothbrushes in the bathroom cupboard and keep the bathroom door closed when brushing their teeth. Even though such interviews do not form the basis of our sensor selection criteria, they have definitely been helpful for choosing the adequate types and deployment locations of the sensors.

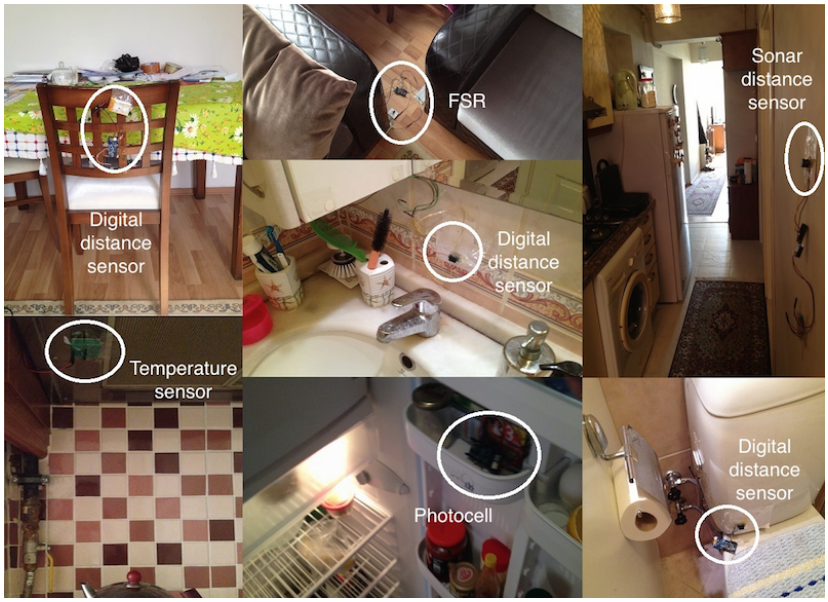
3.1.2. Robustness

The sensors should be selected so as to keep the components of the sensor devices intact and to allow them to function without malfunctioning in the event of possible activities involving the interaction of the users with the sensors. For instance, we have initially used an FSR sensor, which is placed under the leg of a chair, to detect the action of sitting on the chair. However, it was not sufficiently robust, even if we placed it in a stable position, because it was in contact with the ground, and since the chair is a mobile item, it had a high probability of breaking down or coming apart during the operation of the system. Instead, we preferred to use a digital distance sensor located at the back of the chair to detect the sitting in the chair action. The properties of the specific chair in that house also influenced this decision, since the back of the chair had an appropriate hole to accommodate such a sensor. As another example, we can give the toilet flushing action. Initially, we have used FSR and contact sensors positioned on the flush button successively. However, they were not robust enough to give consistent results. They were prone to dislocation by the physical contact of the users. Therefore, we placed a digital distance sensor to the toilet seat cover to recognize whether it is

open or not. As a learned lesson, we can state that the sensors should be selected and deployed so as to minimize the contact between the sensors and subjects, in order to increase robustness, hence enabling clean and consistent results.

3.1.3. Efficiency

No matter how robust or intuitively convenient a sensor is for a targeted activity, it cannot be considered an adequate choice unless it is efficient. Efficiency is directly related to the correctness and completeness of the readings a sensor generates in harmony with the targeted activity. As an example, we have initially used the humidity sensor to detect the activity of taking a shower. However, using this sensor, the exact duration of the showering behavior could not be inferred, since the relative humidity in the bathroom does not decrease rapidly; hence, the humidity sensor continues to give high humidity values for a long time even after the activity is completed. Therefore, we decided to use a contact sensor located on the shower cabin door to detect whether the shower cabin is closed. Another example is the choice between the photocell and the vibration sensors to detect if a drawer is opened. The vibration sensor placed inside a drawer starts firing with the motion of opening the drawer, as expected; however, even after the drawer is closed, the vibrations continue, thus making it impossible to infer when the action ends. Therefore, we preferred photocells that are far more efficient to detect such an action. The tuning of the threshold values for the initiation of sensor firings also plays an important role in adjusting and enhancing the sensor's efficiency. For instance, to detect if a person is sleeping, we were able to use an FSR sensor, since setting a threshold enabled the sensor to recognize the extra weight of the person in addition to the weight of the bed. However, the sensitivity of a specific sensor ultimately determines if such a fine threshold could be set; hence, the efficiency of a sensor also depends on its sensitivity with respect to the sensing requirements of a targeted activity. Several example sensor deployments in real houses, which are made considering the above-mentioned criteria, are shown in Figure 3.1.



(a) House A

(b) House B

Figure 3.1. Example deployments of ambient sensors considering the designated criteria.

3.2. Networking

The proposed system's networking component is composed of star-topology ZigBee networks. Depending on the coverage of the central base stations (coordinators), the network consists of one or a few personal area networks (PANs) operating in different channels. In case of multiple PANs, the coordinators (cluster heads) of individual PANs communicate with each other through a base station (e.g., access point), hence creating a star-star tree topology. We use commercialized Xbee transceivers, compatible with the Arduino modules, as the ZigBee solution. Due to the obstacles and walls affecting the signal propagation in a typical house, multiple PAN coordinators may be required to achieve complete coverage of the deployed sensor devices. The PAN

coordinators should be deployed to provide line-of-sight communication with as many sensors as possible. The communication channel selections are to be made based on their overlap with the WiFi networks in the vicinity, since ZigBee and WiFi standards utilize overlapping bandwidths.

Each sensor unit sends sensor values to the associated PAN coordinator when an event is detected. The sensors and coordinator within the same PAN utilize the same channel for transmissions, which is to be set differently from the channel's other PANs use, in order to prevent interference. The PAN coordinators are connected to a central processing unit via a serial interface. In the central unit, the data from the two subnetworks and the ground-truth labels are matched and synchronized, for which the details are given in the next section.

The sensor nodes are configured to transmit data in an event-based binary format, although the sensors being used are not binary. The sensors produce values from zero to 1024. In order to convert the sensor data to binary format, we use thresholding. During the operation of the system, a sensor is sampled ten times in a second, and the sensor value is compared with the predefined threshold value specific for that sensor. If the sensor value exceeds a predefined non-activity range, an event is detected. Upon detection, the sensor node wakes up its transmitter and starts transmitting binary data to the relevant PAN coordinator. As the sensor values fall back under the specified thresholds, data transmission stops, and the transmitters are switched to sleep mode again in order to save energy. Despite the increased battery lifetime advantage of putting transmitters into sleep modes, it has a notable drawback. Although the wake-up time for the Xbee module is as low as $13.2ms$, the reassociation of the transmitters with the PAN coordinators can take as long as $300ms$. On the other hand, given the considerable increase in the battery lifetime and the typical duration characteristics of human activities, this delay does not affect the performance of the system significantly; hence, using sleep mode is more preferable. During our one-month field study, in each of the two houses, the battery replacement frequency varied between two times to eight times, depending on how frequently a specific sensor detects an event and transmit readings.

3.3. Data Collection and Annotation

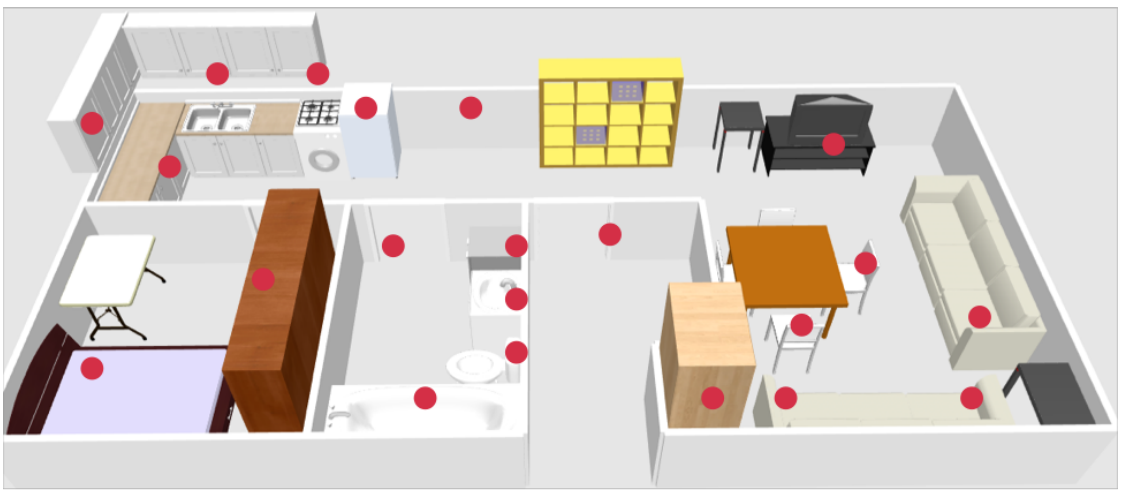
The sensor data flowing to the PAN coordinators are synchronized and time-stamped at a central component. The raw sensor data obtained in this stage has a granularity of seconds. Since privacy is of utmost concern for the proposed system, we avoid the use of video cameras for obtaining the ground truth activity labels. Previous studies use methods, like keeping a diary or using Bluetooth headsets, for annotation. Instead, we provide a software application, running on a laptop situated in the house, with a simple user interface and ask the residents to provide the ground-truth labels of the activities in which they were engaged. Our method is more accurate than manually keeping a diary and more user-friendly than wearing a headset all the times [43]. Furthermore, we did not ask them to carry any identification sensors on them to ensure unobtrusiveness. Likewise, as previously stated, sensors were placed in convenient locations to ensure the natural behavior of the residents and not to disturb their daily routines. Moreover, during the field study, the residents were not required to follow a specific scenario and were asked to continue leading their daily lives as if the AAL system did not exist. In our field study, there are from 60 to 100 labels for each day, which indicates the level of detail of the ground-truth labels made possible by the user-friendliness of the ground-truth labeling interface. The interface contained 27 different activity labels, including every day activities, like sleeping, brushing teeth, watching television (TV), toileting, preparing a meal and eating. Rare activities that are not

Table 3.2. Properties of ARAS datasets.

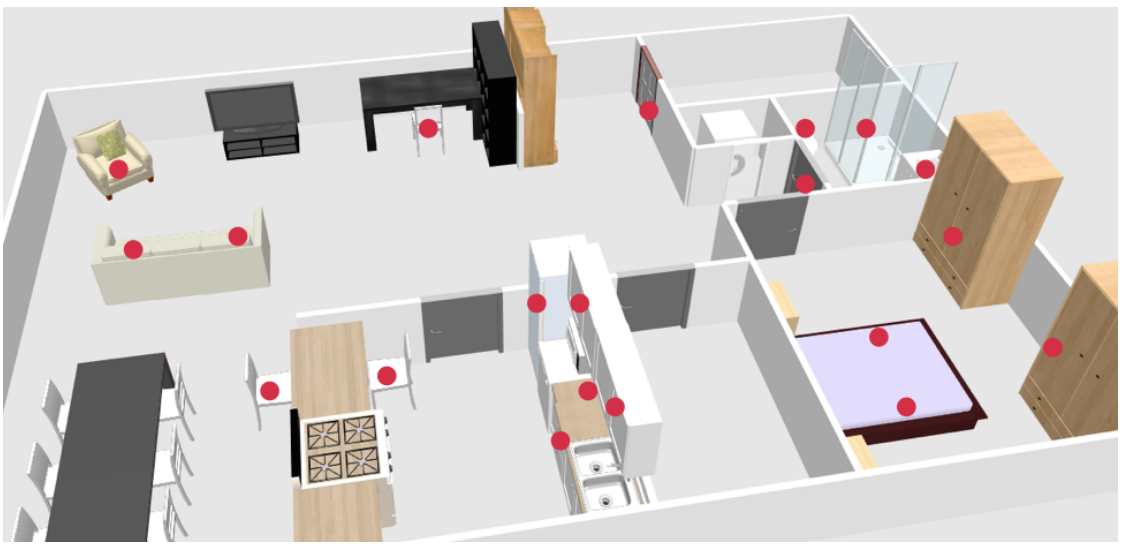
	House A	**House B**
# of PANs	2	1
# of Ambient Sensors	20 of 7 different types	20 of 6 different types
Size of the House	50 m^2	90 m^2
House Information	One bedroom, one living room, one kitchen, one bathroom	2 bedrooms, one living room, one kitchen, one bathroom
Residents	2 males both aged 25	Male-female couple, age average 34
Duration	27 - 30 full days	27 - 30 full days
# of Activities	14	12

performed every day, such as hanging out laundry, having a guest, doing cleaning and having a nap, are also captured. These rare activities might have great significance in an application inferring the health status or wellbeing of the residents. Therefore, we gathered information about such activities, unlike most of the previous studies.

We deployed the described system in two real home settings and collected fully labeled one-month-long datasets. The details about the two houses (annotated as House A and B), the deployed systems, the residents and the collected data are given in Table 3.2. The detailed layouts of Houses A and B along with the locations of the deployed sensors are presented in Figures 3.2a and 3.2b, respectively. Unlike most of



(a) House A



(b) House B

Figure 3.2. House layouts and sensor deployments in ARAS datasets.

the other similar studies that include the deployment of systems collecting daily living data regarding people, the data we collected from each house is composed of sensor readings influenced by two residents who share the same house. We think that such a setting reflects real life more closely by accounting for most of the people who live with their family, spouse and friends, and additionally, it will give the opportunity to investigate the social interaction patterns between couples. Moreover, since we have used real homes instead of controlled laboratory environments and allowed the residents to pursue their normal daily lives and perform their regular behaviors/routines, we

Table 3.3. Availability and locations of sensors in both houses.

Sensor	**Location**	**House A**	**House B**
Contact sensor on shower cabinet	Bathroom	✓	✓
Distance sensor above tap	Bathroom	✓	✓
Contact/distance sensor on door	Bathroom	✓	✓
Distance sensor on WC	Bathroom	✓	✓
Photocell in bathroom cabinet	Bathroom	✓	
Photocell in fridge	Kitchen	✓	✓
Photocell in drawer	Kitchen	✓	✓
Distance sensor on wall	Kitchen	✓	✓
Temperature sensor above oven	Kitchen	✓	
Contact sensor on right cupboard	Kitchen		✓
Contact sensor on left cupboard	Kitchen		✓
Force sensor on chair	Kitchen		✓
Force sensor on chair	Living room	✓	✓
Force sensor on chair	Living room	✓	✓
Infrared reader below TV	Living room	✓	
Force sensor on chair	Living room	✓	✓
Force sensor on armchair	Living room		✓
Force sensor on couch/bed	Living room/bedroom	✓	✓
Contact sensor/photocell in wardrobe	Living room/bedroom	✓	✓
Photocell in convertible couch	Living room/bedroom	✓	
Contact sensor/photocell in wardrobe	Bedroom	✓	✓
Force sensor on bed	Bedroom	✓	✓
Force sensor on bed	Bedroom		✓
Contact sensor on outside door	Hall	✓	✓
Distance sensor on wall	Hall	✓	

argue that the data we collected is more realistic.

Although the annotation interface contained 27 different choices for activities performed, due to the differences in the lifestyles of the residents in each house, the list of activities labelled by the residents differ. Also, some of the activities did not happen or happened only once during the one month long dataset collection phase. Overall, the number of recorded activities in House A and House B is 14 and 12, respectively. The availability of different sensors for each house with their types and detailed locations are also listed in Table 3.3.

3.4. Activity Recognition Performance Evaluation

Markov models are widely used in the literature for modeling sequential data because they are well suited for handling the temporal dependencies. Since human activities are sequential in nature, Markov models have already proven to be useful for human activity recognition purposes. In this section, we aim to provide more insight on the datasets, therefore, we provide the activity recognition performance on ARAS datasets using an HMM. In this way, first, we provide a benchmark on the activity recognition performance on ARAS datasets; second, we compare and contrast the differences among the houses and the residents both on an activity level and daily level.

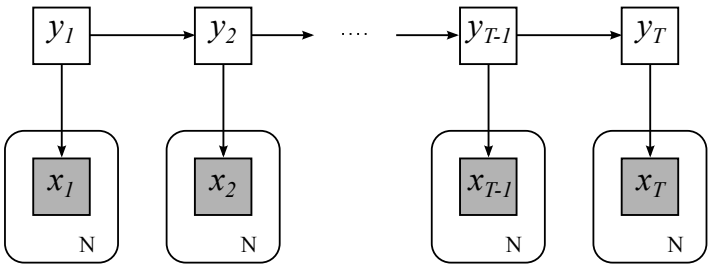


Figure 3.3. Hidden Markov model for activity recognition using N binary sensors.

3.4.1. Experimental Setup

We use the HMM depicted in Figure 3.3. The hidden state at time t, denoted as y_t, correspond to the activities performed and the observations, x_t^i correspond to i^{th} sensor's value at time t. Each sensor modeled as an independent binary feature. The total number of sensors (features) is $N = 20$ for ARAS datasets. The total number of time steps is denoted as T. HMM is a generative model that has three factors in the joint probability distribution:

$$\begin{aligned} p(y_{1:T}, x_{1:T}) &= p(y_1) \prod_{t=2}^{T} p(y_t \mid y_{t-1}) \prod_{t=1}^{T} p(x_t \mid y_t) \\ \pi &= p(y_1) \\ A &= \prod_{t=2}^{T} p(y_t \mid y_{t-1}) \\ B &= \prod_{t=1}^{T} p(x_t \mid y_t) \end{aligned} \tag{3.1}$$

The initial state distribution $p(y_1)$ is a multinomial distribution parameterized by π; the transition distribution $p(y_t \mid y_{t-1})$ is represented as a collection of Q multinomial distributions (Q is the number of different activities), parameterized by A; the observation distribution $p(x_t \mid y_t)$ is a multiplication of N independent Bernoulli distributions (N is the number of sensors), parameterized by B.

$$\begin{aligned} p(x_t \mid y_t) &= \prod_{i=1}^{N} p(x_t^i \mid y_t) \\ p(x^i \mid y = j) &\sim Ber(\mu_{ij}) \end{aligned} \tag{3.2}$$

The entire model is parameterized by a set of three parameters $\chi = \{\pi, A, B\}$. We use a fully supervised approach with the maximum likelihood method for learning the parameters and the well-known Viterbi algorithm for inference. In order to prevent zero probabilities, we use Laplace smoothing during parameter learning.

Sensor data is discretized in $\Delta t = 60sec$ intervals. Overall, there are $T = 1440$

data points for each day. For each sensor, we used the value 1 if the sensor has been fired at least once during the interval and 0 otherwise. Although the sensor data from both residents are fused at the time of data collection, we manually decomposed the observation space into two by considering the ground truth activity labels and the sensor data pattern with a set of predefined rules. For the ground truth labels used in training phase, we used the activity label that has the largest number of occurrences during that interval. We use leave-one-day-out cross validation in our experiments. We use one full day of data for testing and the remaining days for training. We cycle over days for testing and use every day once for testing. We report the average of the performance measure.

For measuring the performance, we use precision, recall, f-measure, and accuracy. For a multi-class classification problem we define the metrics averaged over the number of activity classes as follows:

$$Precision = \frac{1}{Q}\sum_{i=1}^{Q}\frac{TP_i}{TP_i + FP_i} \tag{3.3a}$$

$$Recall = \frac{1}{Q}\sum_{i=1}^{Q}\frac{TP_i}{TP_i + FN_i} \tag{3.3b}$$

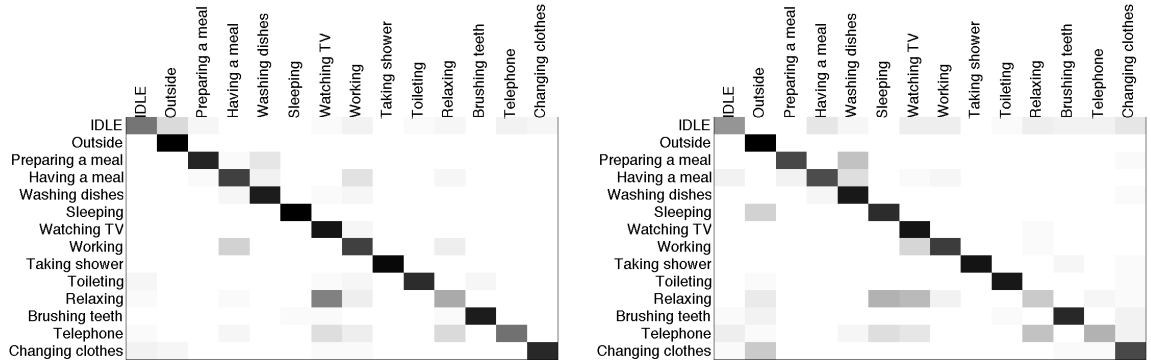
$$F - measure = 2\frac{Precision.Recall}{Precision + Recall} \tag{3.3c}$$

$$Accuracy = \frac{\sum_{i=1}^{Q} TP_i}{Total\ \#\ of\ \ DataPoints} \tag{3.3d}$$

where Q is the number of classes, TP_i is the number of true positive (TP) classifications for class i, FP_i is the number of false positive (FP) classifications for class i, and FN_i is the number of false negative (FN) classifications for class i.

3.4.2. Results

We present the results for an activity level performance and also from a daily recognition perspective. In Figure 3.4, we depict the confusion matrices for House A for both residents. There are 14 classes in House A. The average performances for

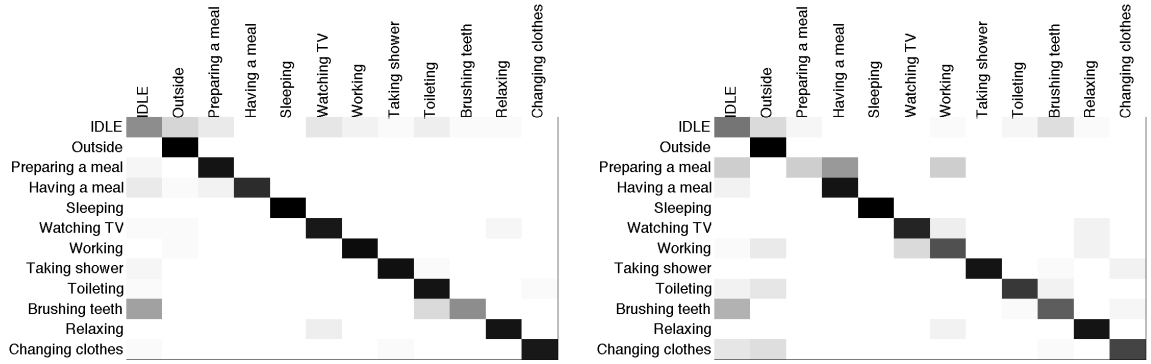


(a) Resident 1 (b) Resident 2

Figure 3.4. Confusion matrices for activity recognition using HMM in House A.

activities in terms of f-measure are 77.5% and 69.2% for Resident 1 and Resident 2 respectively. Mostly, the model confuses the activities that are performed in the living room for Resident 1. Activities that resemble each other such as *relaxing* and *watching TV* are confused mostly. Also, *talking on the phone* activity is confused with *relaxing*, *watching TV* and *working* since this activity has no specific pattern except than moving around the house and sitting down at different places between these moves. *Working* and *having a meal* activities are confused mostly because they were both performed at the table in living room. For the second resident, the same confusions are more prominent. Also for the second resident, *sleeping* activity is mostly confused with *relaxing* and *talking on the phone* and also with *being outside*. The reasons for these confusions are twofold. First, the resident choses to sit on his bed while talking on the phone and during the relaxation activities like reading and surfing the Internet and second, the force sensor attached to his bed is problematic. It frequently stopped sending data during sleeping. Finally, we observe the confusions between the kitchen activities, especially the model mixed *washing the dishes* with *preparing a meal*. This is due to the fact that these activities have similar pattern in terms of interactions with the sensors deployed, although they are semantically different.

The confusion matrices for House B are given in Figure 3.5. In total, there are 12 activities in House B. The average performances for activities in terms of f-measure are 80.4% and 77.3% for Resident 1 and Resident 2, respectively. We have similar patterns to House A in terms of *watching TV* and *relaxing* activities in House B. On

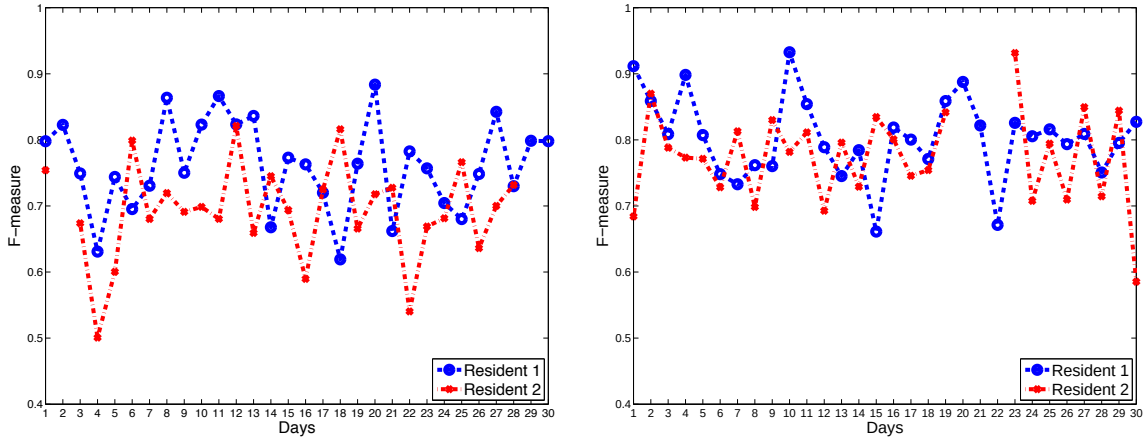


(a) Resident 1 (b) Resident 2

Figure 3.5. Confusion matrices for activity recognition using HMM in House B.

the other hand, *brushing teeth* activity in this house is more challenging in terms of recognition. This is due to the layout of the sensors in the bathroom. In House A, there exists a cupboard containing the tooth brushes. The sensor in the cupboard gave strong clues about the brushing teeth activity in House A. In House B, there existed no such cupboard so that it became difficult to infer this activity. For the second resident, there existed very few occurrences of *preparing a meal* so that the model is not trained well for this activity. It is confused with other activities such as *having a meal* and *working* and *being idle* for the second resident. Nevertheless, we prefer keeping this activity as it is for the sake of coherence between the residents and the houses.

In terms of daily recognition performances we both give results in terms of f-measure and accuracy. The daily average f-measure performances for 30 days for House A and B are given in Figure 3.6a and Figure 3.6b respectively. For both houses, the second residents were on a business trip and were absent for three days. These absent days for the residents are reflected as gaps in the graphs. The variations among the days can be attributed to the differences in the daily activity patterns, such as work days *vs.* weekends. In House A, there are prominent differences among the two residents' activity recognition performances. In the worst day, the second resident's activity recognition performance is 50% and for the maximum, we have almost 95% f-measure for Resident 1. For House B, the general performance variance is more stable when compared to House A. Also, the recognition rate is higher for House B. This stems partially from the fact that the number of recognized activities are less than House A.

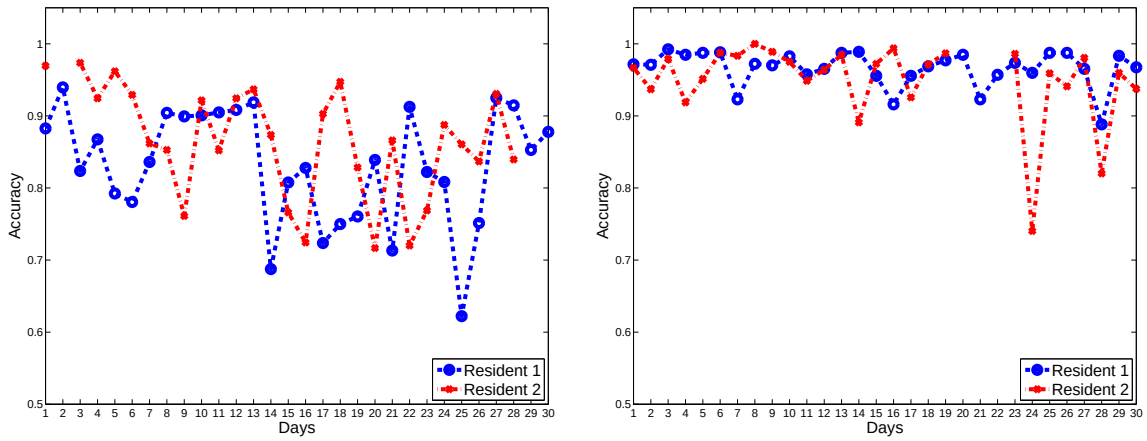


(a) House A (b) House B

Figure 3.6. Daily average activity recognition performance in terms of f-measure in ARAS datasets.

Also, the sensors were made more robust in House B and the residents lifestyles in House B were more sedentary as opposed to House A.

The average accuracy values in House A are 86.3% and 86.4% for Resident 1 and 2 respectively. In Figure 3.7, we give the time-slice level accuracies for each day in both houses. These graphs shows the percentage of correctly classified time-slices, therefore correctly classified longer duration activities have larger weights in the measure unlike the average f-measure metric which treats each activity as equally important. In the worst case, we could correctly classify the 70% of all time-slices in a day and in the best



(a) House A (b) House B

Figure 3.7. Daily average activity recognition performance in terms of accuracy in ARAS datasets.

case the performance rises up to 99%. In general, the accuracy is higher in House B. This can be attributed to the longer durations of better recognized activities in House B. For the first resident, the average accuracy is 96.6% and for the second resident it is 94.9%.

3.5. Conclusion

Despite the challenges in processing large amounts of sensor data, wireless networking and the limitations of the sensor devices, WSNs are gradually being used in activity recognition purposes in AAL applications. In this chapter, we introduced our multimodal WSN-based AAL system compatible for homes with multiple residents with the aim of recognizing the daily activities and routines of the users to detect the drifts and differences in their behavior, especially for monitoring their health and wellbeing status. In particular, we focused on the details of the system architecture and provided guidelines for the design and deployment of an effective AAL system. We presented the details of our field study to evaluate the success of the system where it was deployed in two different real home environments with multiple residents and collected data from different types of ambient sensors about different activities for 30 full days. Finally, we provided the results and insights on the activity recognition performance on ARAS datasets using an HMM. The results presented in the chapter will be used for benchmarking purposes in the following chapters.

4. HIERARCHICAL HMM WITH VARIABLE NUMBER OF STATES

4.1. Introduction

Human activities are complex and contain rich hierarchical structure and previous work has shown that modeling this structure can benefit the recognition of human activities from sensor data [71]. However, the added complexity that a hierarchy brings can make the construction of an accurately fitting hierarchical model challenging, while the additional layers of representation can require additional annotation efforts for supervised learning methods. This makes it more difficult to deploy such models in different configurations and environments, which limits their applicability. We can assume that a human activity can be broken into a set of actions that represent more atomic events of the behavioral routine. For example, an activity like cooking might consist of an action 'cutting vegetables' and an action 'frying them in a pan'. Our proposed hierarchical model learns the model parameters using a semi-supervised learning method that requires labeled data for the activities, but not for actions. The actions in the model are only used for recognition purposes, so we can remain agnostic about the interpretation of the actions that the learning method allocates. The only design consideration is the number of states used to represent the actions that make up each activity.

In this chapter, we focus on model selection for hierarchical Markov models and show that a variable number of actions per activity can further improve the recognition performance. Unlike most of the previous studies that assume a fixed number of actions for each activity [72, 73], we propose a model selection approach to determine the number of actions for each activity separately. We evaluate the model selection performance on real world datasets and show the performance increase due to the hierarchy with carefully selected models.

4.2. Related Work

Human behavior modeling using different modalities of sensing has been an active research topic for the last decade. The data were obtained from either ambient sensors deployed in the environment such as video [74, 75], audio [71, 76], and binary sensors [46,77] or wearable sensors deployed on the body such as accelerometers and gyroscopes [78, 79]. Although there are different modalities of sensing, in terms of modeling of human activities, temporal probabilistic models such as HMMs and CRFs have been shown to give better results with their ability of modeling the temporal dependencies and sequential nature of human activities.

Despite the powerful temporal modeling abilities, the flat versions of these models often fail to accurately model the complex nature of human activities with a variety of possible ways of performing the activity and with different interactions with the environment. Therefore, hierarchical models were used to obtain a more grained model for complex human activities.

The Hierarchical HMM (HHMM) is a generalization of the HMM that can have a hierarchical structure and is introduced by Fine *et al.* [80] for modeling complex multi-scale structure in sequential data. The original inference algorithm has cubic time complexity in terms of the sequence length which prevented it to be applied to domains where the sequences are long. Murphy el al. [81] showed that the HHMM can be represented as a dynamic Bayesian network (DBN) with a linear time inference complexity with respect to the sequence length. This much simpler and more efficient inference algorithm has made the hierarchical models good candidates for modeling the data in many different domains, such as natural language processing, handwriting recognition and human activity modeling.

There are several studies that use hierarchical models in human activity recognition. Kasteren *et al.* [72] proposed a two layer hierarchy where the top layer represents the human activities of daily living and the second layer are the several actions made during the course of the actual activity. The experiments on three real world smart

home datasets reveal that the use of two or three action clusters per activity gives the best performance.

Karaman *et al.* [82] use two level hierarchical model with multimodal audio and video data in order to classify human activities. The semantic activities are encoded in the top-level followed by a bottom level HHM that models an activity with a number of non-semantic states. They experimented with three, five or seven sub-states and reported that using 3 non-semantic sub-states yields better performance.

While the previous studies already showed the improvement over the flat HMM models, they use an equal and fixed number of states in the second layer of the hierarchy. Therefore, they assume the same level of complexity for every activity at the top layer. However, it is very likely that the complexity of different activities varies. For sleeping activity, one or two states may be sufficient whereas preparing a meal requires much more complicated interactions with the environment and therefore it requires more states to be accurately modeled. Therefore, the ideal number of states for each top layer activity should be decided separately.

Celeux and Durand [83] proposed using penalized cross-validated likelihood criteria to determine the number of hidden states. They compare the performance of several information criteria such as AIC, BIC, penalized marginal likelihood (PML) and integrated complete likelihood (ICL) using simulated data. According to the results, AIC, BIC and ICL were observed having similar behavior. They also state that AIC has a tendency to under-penalize the complexity of a model, ICL favors models that give rise to partitioning the data with the greatest evidence from the hidden states, and BIC performs well only if an HMM gives a representation of the observed process. PML converges very slowly to the optimal solution. Moreover, in practical situations, it seems to have a high tendency to over-penalize the complexity of HMM model when the sequence length is not very large.

4.3. Hierarchical HMM with Variable Number of States

In this section, we first describe the hierarchical model we use for behavior modeling followed by our proposed method for selecting the sub-states within an activity.

4.3.1. Hierarchical HMM

Our model for activity recognition is a two-layer hierarchical hidden Markov model as depicted in Figure 4.1. The top layer state variables y_t represent the activities and the bottom layer variables z_t represent the action clusters. Each activity consists of a sequence of action clusters and the temporal ordering of these action clusters can vary between different executions of an activity. The last action cluster of the sequence signifies the end of an activity and indicates the start of a new sequence of action clusters. This information is captured by the finished state variable f_t, which is used as a binary indicator to indicate that the bottom layer has finished its sequence.

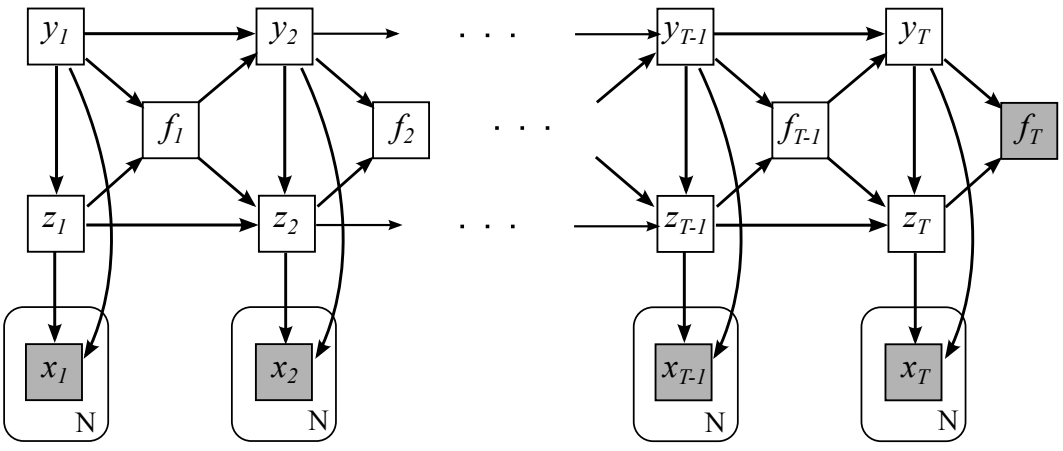


Figure 4.1. The graphical representation of a two-layer HHMM. Shaded nodes represent observable variables, the white nodes represent hidden states.

The joint probability distribution of the model factorizes as follows:

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \mathbf{f}_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(x_t \mid y_t, z_t) p(y_t \mid y_{t-1}, f_{t-1}) p(z_t \mid z_{t-1}, y_t, f_{t-1}) p(f_t \mid z_t, y_t) \tag{4.1}$$

where we have defined $p(y_1 \mid y_0, f_0) = p(y_1)$ and $p(z_1 \mid z_0, y_1, f_0) = p(z_1 \mid y_1)$ for the sake of notational simplicity.

The entire model consists of a set of parameters $\theta = \{\pi_0, \pi_{1:Q}, A_0, A_{1:Q}, B, \phi\}$. These parameters are learned in a semi-supervised way by using the expectation-maximization (EM) algorithm. The initial state parameters π and transition parameters A exist for both the top layer and bottom layer states. To distinguish between these two types of parameters, we include a 0 in the subscript to indicate that a parameter is of the top layer and an index of 1 to Q for each of the bottom layer parameters. The distributions of the bottom layer states depend on which top layer state the model is in and so there is a separate set of bottom layer state parameters for each possible top layer state, with Q being the number of top layer states. For example, if the model at one point is in the top state $y_t = k$, then the transition parameter A_k is used for the bottom layer state transitions. We now provide a detailed explanation of each of the factors that make up the joint probability and how they are parameterized.

At the first time-slice, the initial state distribution of the top layer states is represented by a multinomial distribution which is parameterized as $p(y_1 = j) = \pi_0(j)$. This top layer state generates a bottom layer state, also represented by a multinomial distribution and parameterized as $p(z_1 = j \mid y_1 = k) = \pi_k(j)$.

The factor $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f)$ represents the transition probabilities of the bottom layer state variable. These transitions allow us to incorporate the probability of a particular temporal order of action clusters with respect to a given activity. A transition into a new state z_t, depends on the previous bottom layer variable z_{t-1}, the current top layer state variable y_t and the finished state variable f_{t-1}. Two distributions make up this factor, depending on the value of the finished state variable f_{t-1}. If in the previous time-slice the bottom layer state sequence ended ($f_{t-1} = 1$), a new sequence of bottom layer states starts at this time-slice and therefore the top layer state generates a bottom layer state using the same distribution as we saw at the first time-slice, parameterized by the set of parameters $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = \pi_k(j)$. In case the bottom layer state sequence did not end ($f_{t-1} = 0$), a transition to a new bottom layer state is made using the transition matrix parameterized as $p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = A_k(i, j)$.

These two cases can be compactly formulated as:

$$p(z_t = j \mid z_{t-1} = i, y_t = k, f_{t-1} = f) = \begin{cases} A_k(i,j) & \text{if } f = 0 \\ \pi_k(j) & \text{if } f = 1 \end{cases} \tag{4.2}$$

Transitions of the top layer state variables are represented by the factor $p(y_t = j \mid y_{t-1} = i, f_{t-1} = f)$. This factor is similar to the transition distribution of an HMM, except that it also depends on the finished state variable f_{t-1}. This dependency is important because it restricts the model in transitioning to a different top layer state as long as the bottom layer state sequence has not finished. When a bottom layer state sequence did not finish, the top layer state variable continues into the next time-slice with the same state value ($y_t = y_{t-1}$). Once the bottom layer state sequence has ended, a transition of the top layer state is made according to a transition matrix parameterized as $p(y_t = j \mid y_{t-1} = i, f_{t-1} = f) = A_0(i,j)$.

These two cases can be compactly formulated as:

$$p(y_t = j \mid y_{t-1} = i, f_{t-1} = f) = \begin{cases} \delta_{ij} & \text{if } f = 0 \\ A_0(i,j) & \text{if } f = 1 \end{cases} \tag{4.3}$$

where δ_{ij} is the Kronecker delta function, giving 1 if $i = j$ and 0 otherwise.

The probability of a bottom layer state sequence finishing is represented by the factor $p(f_t = f \mid y_t = j, z_t = l)$. This factor depends on both the bottom layer state z_t and the top layer state y_t. Even though the variable f_t indicates whether z_t is a finishing state, it is important that the distribution is also conditioned on the top layer state y_t. This is because the probability of a particular action cluster being the last action cluster for that activity can differ among activities. The factor is represented using a binomial distribution, parameterized as $p(f_t = f \mid y_t = j, z_t = l) = \phi_f(j,l)$.

We use Bernoulli observation model by modeling each sensor corresponding to

one Bernoulli distribution. The conditional probability factorizes as follows:

$$\begin{aligned} p(x_t \mid y_t, z_t) &= \prod_{i=1}^{N} p(x_t^i \mid y_t, z_t) \\ p(x^i \mid y = j, z = k) &\sim Ber(\mu_{ijk}) \end{aligned} \tag{4.4}$$

where N is the number of sensors.

4.3.2. Model Selection for Sub-States

In order to estimate the number of hidden states in an HMM, there are several approaches.

- *Fully Bayesian Approach* is to treat the number of states k as a parameter and obtain a posterior distribution on k given the data and the set of models. However, even for the simplest Gaussian mixture model, this posterior cannot be obtained in closed form. Approximate methods should be used.
- *Penalized Likelihood* methods were derived as different approximations to the full Bayesian solution. These methods use a penalty term together with the data likelihood in order to prevent overfitting since it is possible to increase the likelihood by adding more parameters. The two mostly used penalized likelihood methods are BIC and AIC. They resolve the overfitting problem by introducing a penalty term for the number of parameters in the model. BIC further uses the sample size in penalty term, thus the penalty term is larger in BIC than in AIC.
- *Cross-Validated Likelihood (CVL)* judges the models on their estimated predictive performance. The data is separated into training and test sets using a cross validation scheme. Then, repeatedly, the models are estimated using the training set and evaluated the likelihood on the test set. This brings an increase in the computation by a factor of the number of cross validation folds when compared to penalized likelihood approach.

More formally, given a set of models, the model that has the minimum value of Equation 4.5 is the one to be preferred when using AIC. Similarly, when using BIC, the model that has the minimum value of Equation. 4.6 is preferred.

$$AIC = -2logp(x \mid \theta_D) + 2m \tag{4.5}$$

$$BIC = -2logp(x \mid \theta_D) + mlog(n) \tag{4.6}$$

$$CVL = -2logp(x \mid \theta_{CV}) \tag{4.7}$$

where $logp(x \mid \theta_D)$ is the data likelihood using all data, $logp(x \mid \theta_{CV})$ is the likelihood on test set using a cross validation approach with leave-one-out scheme, m is the number of free parameters and n is the length of the sequence.

We find the optimum number of sub-states with penalized likelihood methods as follows. For each activity a, we take all occurrences of that activity as different data sequences. We denote the total number of such sequences as K_a. We then experiment with different models having different number of states starting from one up to ten. For each model size, we learn the parameters on all K_a sequences using EM algorithm. Then for each sequence, we calculate the AIC and BIC scores using Equation 4.5 and Equation 4.6, respectively. Then we select the model with the minimum AIC or BIC. For the cross validated likelihood approach, for each activity a, we use a leave-one-out scheme, i.e. fitting a model using the $K_a - 1$ sequences and compute the likelihood on the remaining test sequence using Equation 4.7. The complete procedure for determining the optimal model size for each activity using BIC, AIC and CVL is given in Figure 4.2.

input: $\mathcal{A}$ Set of Activities

$\mathcal{D}$ Dataset

for all $a \in \mathcal{A}$ **do**

$\mathcal{O} = \{$All occurrences of a in $\mathcal{D}\}$

$\theta_o \leftarrow$ Learn parameters using EM on $\mathcal{O}$

for c = 1 **to** *MaxStates* **do**

$m \leftarrow$ Number of free parameters in the model

for all $o \in \mathcal{O}$ **do**

$BIC_c = -2logp(o \mid \theta_o) + mlog(length(o))$

$AIC_c = -2logp(o \mid \theta_o) + 2m$

$\mathcal{T} = \mathcal{O} \setminus \{o\}$ //Use remaining occurences

$\theta_t \leftarrow$ Learn parameters using EM on $\mathcal{T}$

$CVL_c = -logp(o \mid \theta_t)$

end for

end for

Assign the model with minimum score

$sc^*_{BIC} = \arg\min_c \ BIC$

$sc^*_{AIC} = \arg\min_c \ AIC$

$sc^*_{CVL} = \arg\min_c \ CVL$

end for

output: sc^*_{BIC}, sc^*_{AIC}, sc^*_{CVL}

Figure 4.2. Model selection algorithm using AIC, BIC, and CVL.

4.4. Experiments

Our experiments aim to answer two questions: (i) Does allowing different levels of complexity for different activities increase the recognition performance? (ii) How can we determine the optimum model complexity, i.e., the number of sub-states for activities?

We first experiment with a flat HMM and with hierarchical HMMs having a variety of fixed number of sub-states. Then, we experiment with three different sub-state selection methods: AIC, BIC and CVL. In the remainder of this section, we present the details of our experimental setup, we describe the datasets used in the experiments and provide the details of our configuration selection methods.

4.4.1. Experimental Setup

We use ARAS datasets with a manually decomposed observation space as described in the previous chapter (see Section 3.4) in order to make a proper comparison with the flat HMM version used in the previous chapter. The data are discretized in $\Delta t = 60sec$ using raw feature representation. We use leave-one-out cross validation approach and measure the recognition performance on a time-slice level using the f-measure, which is the harmonic mean of precision and recall values. Since we use EM algorithm whose performance depend on the random initialization of the starting parameters, we repeat the experiments 20 times and present the average over those runs.

4.4.2. Model Selection for Activity Complexity Determination

In order to find a suitable number of sub-states for each activity, we use AIC, BIC, and CVL measures described in the previous section. We use all the occurrences of a given activity as a separate dataset. In order to obtain the optimum complexity level for the given activity, we start experimenting with the minimum possible model having a single cluster and try up to ten clusters. In Table 4.1, we provide the model selection procedure's results on ARAS datasets.

According to the results, we observe variance in terms of different model selection criteria. Also, for the very same activity, there are differences among the different residents and different houses. In terms of model selection criteria, the penalized likelihood methods (AIC and BIC) both have the tendency to select simpler models, confirming the findings of the previous studies. Most of the time, AIC and BIC both

select the same complexity levels. Exceptions to these selections are prominent in *working* activity. The other instances of discrepancy between AIC and BIC are at *telephone* and *sleeping* activities for the first resident *relaxing* activity for the second resident in House A and *watching TV* activity for the second resident in House B. CVL method is generous in predicting the model size. The selected model sizes by CVL are always at least as large as the penalized likelihood methods. The variations between the houses and residents for the same activity indicate the challenges of finding the correct model for human activities. In the following section, we demonstrate the importance of finding the correct model with an experimental evaluation of these model size combinations.

4.5. Results

We summarize the results of our experiments in Table 4.2. Our results demonstrate a significant increase in recognition performance in terms of f-measure when a hierarchical model is used. We also show that allowing different number of sub-states

Table 4.1. Selected sub-states configurations on ARAS datasets.

	House A						House B					
	Resident 1			Resident 2			Resident 1			Resident 2		
Activity	AIC	BIC	CVL	AIC	BIC	CVL	AIC	BIC	CVL	AIC	BIC	CVL
Idle	2	2	7	1	1	1	1	1	8	1	1	7
Outside	2	2	3	2	2	2	2	2	2	1	1	5
Preparing a meal	2	2	4	2	2	4	1	1	1	2	2	2
Having a meal	2	2	6	2	2	6	1	1	1	2	2	4
Washing dishes	1	1	4	1	1	1	-	-	-	-	-	-
Sleeping	5	2	5	4	4	4	2	2	2	2	2	2
Watching TV	3	3	6	3	3	7	3	3	5	4	2	4
Working	3	2	9	4	3	7	4	3	4	4	4	5
Taking shower	1	1	6	2	2	3	2	2	2	1	1	1
Toileting	1	1	6	1	1	3	1	1	1	1	1	5
Relaxing	2	2	8	4	2	6	2	2	5	2	2	4
Brushing teeth	2	2	2	2	2	2	2	2	2	2	2	2
Telephone	3	2	7	2	2	6	-	-	-	-	-	-
Changing clothes	1	2	3	2	2	2	2	2	2	2	2	2

for different activities can result in significant increase in the performance. When we have a fixed number of sub-states, we assume that all activities have the same complexity level. While this assumption may hold for some cases, we cannot always make that assumption. For example, the activities of daily living like *having a shower* or *shaving* can share the same level of complexity depending on the sensor types and deployment places. In that case, allowing different number of sub activities do not help. On the other hand, it is more likely that different activities have different complexity levels. Our results with an equal level of complexities for all activities with levels of two, three and five states failed to give the highest performance. We chose these levels since they have been suggested in the previous studies [72, 82]. Our results confirms that even though we assume the same level of complexity for every activity addition of a hierarchy model helps, yet, we can further improve the performance by allowing different complexity levels for activities.

Table 4.2. Model selection experiment results in terms of percentage f-measure.

			HHMM					
		HMM	All 2	All 3	All 5	BIC	AIC	CVL
House A	Resident 1	77.5	78.6	79.7	81.8	76.7	76.1	**82.9**
	Resident 2	70.9	72.6	69.1	72.0	73.6	73.5	**74.7**
House B	Resident 1	79.8	80.3	80.3	79.8	79.5	79.5	**81.0**
	Resident 2	70.6	70.5	69.8	73.2	71.3	70.9	**73.3**

We experimented with three alternatives for model selection. In terms of model complexity selection strategies, we obtained the best results with CVL method consistently. Selection using AIC and BIC measures resulted in less complex models. Based on the experimental results, we conclude that AIC and BIC measures generally underestimates the complexity of the models for several activities leading to a degradation in recognition performance. However, it is possible to find a better assignment methodology in order to fully make use of the power of hierarchical models.

In order further elaborate on the activity recognition performance, we present an activity level comparison between the flat HMM and HHMM with a model selection using CVL in Table 4.3. We observe a general increasing tendency on the performance

Table 4.3. Activity level performance comparison.

	House A				House B			
	Resident 1		Resident 2		Resident 1		Resident 2	
Activity	HMM	CVL	HMM	CVL	HMM	CVL	HMM	CVL
Idle	61.4	63.8	45.8	46.8	50.0	61.4	57.4	68.5
Outside	97.0	98.6	94.9	99.5	98.9	99.5	98.8	99.5
Preparing a meal	88.6	93.0	77.0	84.1	85.6	86.5	-	-
Having a meal	59.5	79.9	77.1	74.3	87.9	88.6	93.2	92.5
Washing dishes	65.5	82.8	46.7	64.6	-	-	-	-
Sleeping	99.9	99.9	86.4	96.4	99.9	99.9	99.9	99.9
Watching TV	84.9	87.1	84.6	86.9	92.7	92.7	82.6	83.2
Working	79.3	89.1	80.1	68.0	95.6	96.1	77.4	78.6
Taking shower	96.2	95.6	93.6	93.9	85.0	86.4	89.3	89.6
Toileting	87.1	91.1	90.3	89.9	62.8	62.1	76.1	77.4
Relaxing	35.4	55.5	30.1	37.8	72.0	68.2	61.8	57.2
Brushing teeth	83.4	80.9	82.6	82.6	40.4	47.2	40.0	61.2
Telephone	69.1	66.2	43.1	43.4	-	-	-	-
Changing clothes	78.0	77.3	60.5	77.7	86.7	83.8	69.0	72.6

for each activity, yet there are some exceptions. Take for example the *sleeping* activity that has already a quite high recognition performance for most of the residents so that we cannot improve the performance any further. Nevertheless, for the second resident, using a higher number of sub-states increases the performance considerably. The force sensor attached under the bed for the second resident was problematic and stopped firing during sleeping activity. It is evident that using a hierarchical model helps handling the sensor failures in an efficient way. Similarly, the activities requiring more complex interactions with the environments such as *washing the dishes* and *preparing a meal* benefit from the tailored hierarchy levels most. We also observe minor degradations in terms of f-measure for some activities. This is mostly due to the nature of the f-measure. Since it is a harmonic mean of precision and recall providing a compact measure, the opposite movements in each of the measures can lead to a reduction in f-measure. Also, the similarities between the activities should be considered while evaluating the activity level performances. For example, for the second resident in House B, we see an increase in *preparing a meal* performance together with a degradation in *having a meal performance*. Our top-down approach, i.e. fixing the top level activities

and searching for the sub-states within each activity separately, is the most probable cause of this drop since it is very likely for these two activities to share some sub-actions for that resident specifically. The global increase in the activity recognition performance compensates such exceptions.

4.6. Conclusion

In this chapter, we have presented a hierarchical model for the recognition of human activities from sensor data that allows for different model sizes for different states. The proposed model uses a semi-supervised learning approach to automatically cluster the inherent structure of activities into actions. Our experimental evaluations on ARAS datasets shows that the use of a hierarchical model consistently outperforms its non-hierarchical counterpart in terms of recognition performance, given that an adequate number of states is used for modeling the actions in the hierarchy. As opposed to previous work, we employed a model selection mechanism to determine the optimal number of sub-states for each activity. In order to determine the optimum model selection strategy, we experimented with three different criteria. We used model selection using BIC and AIC in a penalized likelihood setup. Also, we experimented with cross validated likelihood approach. Our experiments showed that the model selection using CVL methodology, consistently outperformed the penalized likelihood methods. This finding confirms the previous studies stating that AIC and BIC measures have a tendency to over-penalize the model complexity. Although, the CVL method has a much higher computational complexity, the high increase in the performance redeems.

Our results suggest a great potential in further research for improving the ways of finding the optimal model that can grasp the complexity of human activities. As a future work, it would be interesting to have a bottom-up approach for determining the complexity for the upper-layer activities. Also, rather than finding the optimum model size, we can assume an infinite number of states in the hierarchy by using an infinite hidden Markov model (iHMM) [84] or a hierarchical iHMM model [85].

5. BEHAVIORAL PERFORMANCE EVALUATION

5.1. Introduction

Daily behavior is closely related with the health state of an individual and can be deduced by examining the activities of daily living in terms of start time, duration, and frequency. If changes in human behavior can be detected, situations that require further health evaluation can be identified. Some of these changes concern short term, like recent changes in the last few days, like very frequent usage of the toilet that may indicate that the person may have a urinary infection. On the other hand, some of the behavior changes concern, several months or even years, like preparation of meals are getting longer and longer, and newspaper reading is getting shorter and shorter which may indicate either mild cognitive impairment or more serious forms of dementia.

Other short term behavior change examples which may raise the flag for further inspection are: skipping meals which indicate lack of appetite, excessively long sleeping and lack of social interaction which may be caused by depression. In fact, other sleep disorders, such as shorter and fragmented sleeping may also be caused by certain health problems, or at least should be attended so that they will not become health problems [86, 87]. Identifying short-term behavior change is easier since automated everyday behavior monitoring systems can follow the start times, durations and frequencies of everyday activities. Of course, the weekday, weekend behaviors or seasonal everyday behaviors can be quite different and the monitoring system should be flexible enough to adapt to these expected changes.

Long term behavior changes which may be indicators of health problems are more difficult to identify. It is quite normal to expect that after a certain age, every year, an elderly person may have degrading physical and mental capacity. However, even for a human caretaker, it is not easy to tell when to raise the flag and call for health personnel for further evaluation. Performing some tasks slower may be caused by some orthopedic problems, as well as some form of dementia. On the other hand, introduction of a new

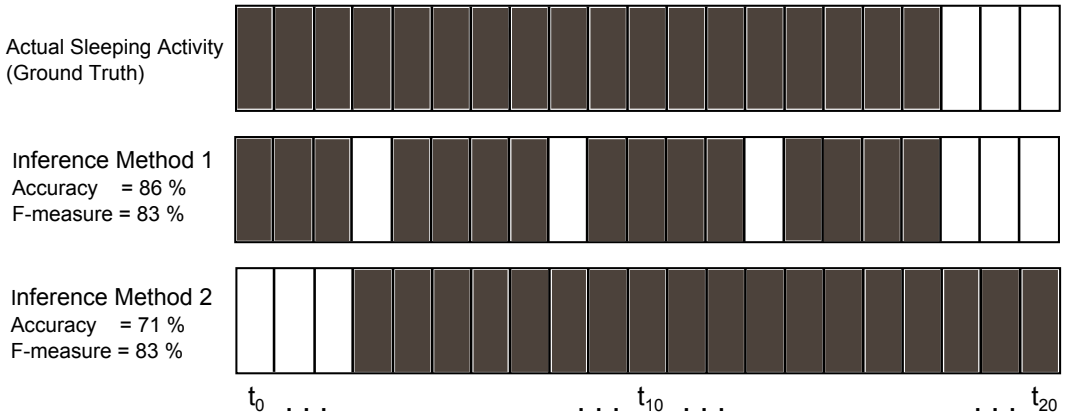


Figure 5.1. Two example of inference output sequence for sleeping activity with the same f-measure performance according to time-slice based evaluation.

home appliance with a new user interface or some other environmental changes can also play a role in these changes.

Some everyday behaviors are indirectly related with health such as eating behavior or social interaction among residents of the house. On the other hand, some everyday behaviors may have a more direct impact on the health such as the behavior of the person related to the medication intake. Frequent changes in the time of medicine, skipping or duplicating medicine intake may have immediate consequences in terms of the health of the person. It is a well-known fact that the quality of the sleep is also directly related with the health of the person. Sleep disorders such as insomnia either may be an indicator of deeper health problems or if not attended may result in serious health problems.

In order to make automated health monitoring systems accurate and robust enough to be commercialized, significant research effort is currently being spent [1]. Several research groups built test environments equipped with sensors and recorded annotated datasets in order to evaluate the performance of novel machine learning methods. However, most of these evaluations are performed in terms of recognition of activities on a time-slice level. The metrics widely used in machine learning domain such as accuracy, precision, recall and f-measure are directly being used in the behavior understanding domain. Although the metrics are solid, they may fail to reveal the ac-

tual performance in terms of behavior understanding. Consider the scenario in Fig. 5.1 and assume that inference methods 1 and 2 are being proposed in order to identify sleeping behavior and their time-slice level outcome is being compared to the ground truth sleeping activity. From a machine learning perspective, both methods have the same F-measure performance of 83% and the first method have higher accuracy than the second method. From a behavior monitoring perspective, the output of the first method indicates that the person may have a sleeping disorder whereas the output of the second method identifies the normal sleeping behavior correctly with a shift in starting time.

This chapter extends the previous work with an evaluation of the state-of-the-art from a behavior recognition perspective rather than using standardized machine learning metrics. We use ARAS datasets for experimental evaluation. We use two separate machine learning models from two different categories in order to compare and contrast the strengths and weaknesses of each category. We use an HMM from the generative model family and use a time windowed neural network (TWNN) from the discriminative model family.

5.2. Related Work

In our previous work [88], we concentrated on performance evaluation for deeper analysis on the strengths and weaknesses of a recognition method. We presented the substitution, occurrence, timing and segmentation errors and showed how to calculate these measures to account for class imbalance and compactly represent them in a single table. The results show that conventional measures such as accuracy is not suitable for representing the recognition performance, because it does not take class imbalance into account. The use of f-measure allows a quick comparison between recognition methods. The use of different error metrics provide a further insight into the strengths and weaknesses of the recognition method.

There are several studies that conduct benchmarking experiments across different datasets for evaluating activity recognition performance [17, 57, 89]. All of these

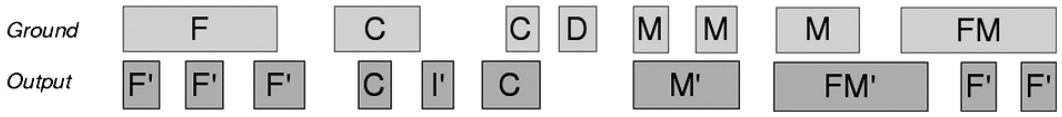


Figure 5.2. Sample event error assignment graph showing each type of error.

studies use standard metrics directly taken from machine learning domain. In machine learning, the standard evaluations are based on four different types of outcomes. The correct outcomes are true positives (TP) and false negatives (FN) and the errors are false positives (FP) and false negatives (FN). Ward *et al.* [90] introduced an extended set of performance metrics for evaluating continuous activity recognition methods. The proposed metrics make use of six different types of errors. They call FP as an *Insertion* error and FN as *Deletion* error. Then they extend the error definitions to include also *Merge*, *Overfill*, *Underfill*, and *Fragmentation* errors. These error definitions are not based on time-slices but based on segments. A *segment* is defined as the largest part of an output sequence on which the comparison between ground truth and the output sequence can be made in an unambiguous way. An *event* is defined as any single occurrence of an activity. Using segment level error assignments, event level error assignments can be made in 8 different categories. Four of these categories belong to ground truth events: deletions (D), fragmented (F), fragmented and merged (FM) and merged (M). The remaining four belong to output events: merging (M'), fragmenting and merging (FM'), fragmenting (F') and insertions (I'). The events that do not belong to any error category are identified as correct (C). In Figure 5.2, the error types are depicted on a sample recognition output scenario.

5.3. Evaluation Methodology

In our proposed evaluation method, we use two levels. In the first level, we use sensitivity (true positive rate) and specificity (true negative rate) analysis based on time-slice performance.

$$Sensitivity = \frac{TP}{P} = \frac{TP}{(TP + FN)} \tag{5.1}$$

$$Specificity = \frac{TN}{N} = \frac{TN}{(FP + TN)} \tag{5.2}$$

Sensitivity and specificity provide an overview of the recognition performance but hide the detailed information about the errors. For instance, we observe the total number of erroneous time-slices but we cannot infer any information about the nature of these errors. The errors can be grouped at a specific position or distributed across the sequence or they can be occurring at the beginning or at the end of the activity. From a machine learning perspective, all types of errors should be eliminated as much as possible. From a behavior analysis perspective, different error types can lead to different meanings about health status of the people being monitored. Depending on the activity type, some errors may not be so harmful to the outcome and some errors have more severe impact on the outcome. For that reason, a second level of analysis is required. In the second level, we analyze the performance using not the time-slices but the activity occurrences (events). For the event based analysis, we use event analysis diagrams (EAD) [90]. In the following subsections, we define time-slice and event level error types we use.

5.3.1. Time-slice Level Error Types

At time-slice level, we use the following categories for false negative (FN) errors:

- Deletion (D_t) occurs when a time-slice corresponds to a deleted event.
- Fragmenting (F_t) occurs when a FN is between two TP segments.
- Start Underfill (U_a) occurs when starting segment of an event is deleted.
- End Underfill (U_ω) occurs when an ending segment of an event is deleted.

Likewise, the following categories are defined for false positive (FP) errors:

- Insertion (I_t) occurs when an activity time-slice that has no corresponding time-slice in the ground truth is produced as output.

- Merge (M_t) occurs when a FP is between two TP segments.
- Start Overfill (O_a) occurs when starting segment of an event is inserted falsely.
- End Overfill (O_ω) occurs when an ending segment of an event is inserted falsely.

Defining several error categories provides the information about the nature of the errors, yet still event level analysis is required in order to get a glimpse of the big picture on the behavior level.

5.3.2. Event Level Error Types

At the event level, the error types are categorized according to the ground truth events and output events that are inferred by the inference method as depicted in Fig 5.2. There are two categories at the event level: ground truth events and output events complement each other's error types. For the ground truth events the error types are defined as follows:

- Deletion (D) occurs when an occurrence of an activity is completely missed.
- Fragmented (F) events occurs when a ground truth activity is output as several fragments.
- Merged (M) events occurs when several instances of ground truth activity are output as a single event.
- Fragmented and Merged (FM) events occurs when a ground truth event is both merged and fragmented.

The output event counterparts of the ground truth events are given as:

- Insertion ($I\prime$)
- Fragmenting ($F\prime$)
- Merging ($M\prime$)
- Fragmenting and Merging ($FM\prime$)

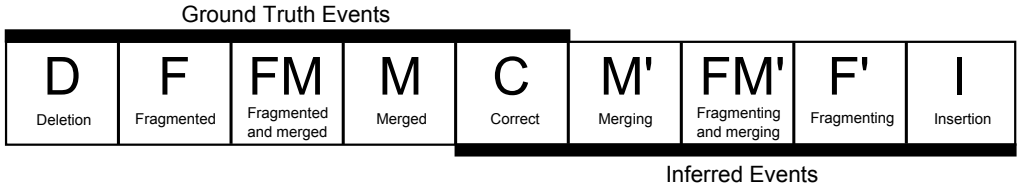


Figure 5.3. EAD graph.

Any event that does not fall in these categories is defined to be a correct (C) event. Figure 5.3 depicts the layout of the event analysis diagrams we use.

5.3.3. Evaluation of Activity Recognition Performance with a Behavior Analysis Perspective

In order to evaluate the performance in terms of behavior recognition, we map the error types to activity types as being negligible and non-negligible. We define three categories of activities for this purpose:

- *Duration sensitive activities* are the ones that only the total duration of the activity is important in terms of medical assessment. For example, relaxing activities such as watching tv, reading a book, leaving the house, or other activities like cleaning the house, studying and talking on the phone can be categorized in this group. For the duration sensitive activities, event level merging and fragmentation errors can be considered as correct events, in turn, time-slice based metrics such as overfill and underfill errors are given more weight in the performance evaluation.
- *Frequency sensitive activities* are the ones only the number of occurrences matters in terms of medical assessment. Having a snack or drink, brushing teeth, taking medicine are the candidate activities for this category. The important error types for this activity category are the fragmentation and merging errors since they can lead to wrong interpretations about the frequency of the activity. Timing errors such as overfill and underfill can be classified as correct events since they do not change the frequency output.
- *Duration and frequency sensitive activities* are the activities for which both the

duration and the frequency are essential. Sleeping, toileting, taking a shower, preparing and eating meals belong to this category. All error metrics should be considered in the recognition performance evaluation for this type of activities.

Based on these categories and the error types defined in the previous sections, we provide a more objective performance evaluation for different behavior monitoring systems. For this purpose, we first categorize the daily activities according to their frequency and duration sensitivity values. After that, we assign the relevant error metrics for the specified activity group in the second step. We provide a general categorization of activities together with our recommended evaluation metrics in Table 5.1. Proposed method is easily generalizable to other activities that are not listed. Once a domain expert such as a physician or another healthcare professional decides the type of the activity, the recommended metrics for an evaluation with a behavioral perspective is easily determined.

5.4. Experiments

In the experimental evaluation, we answer two questions: Which of the machine learning methods are better suited for behavior monitoring rather than activity recognition only? What are the strength and weaknesses of the methods in terms of behavior monitoring? In the following subsection we describe our experimental setup. Then, we give results of the experiments on five datasets taken from ARAS and Kasteren, for both machine learning models, HMM and TWNN, using two levels, time-slice and event level.

5.4.1. Classification Methods

Machine learning methods for classification are grouped into two main categories: discriminative and generative models. Given the training data, discriminative models learn the boundary between classes whereas generative models model the distribution of individual classes. A common view on the generalization performance of generative models is that their performance is poorer than the performance of discriminative

Table 5.1. General categorization of activities.

	Activity Sensitivity		**Recommended Metric**	
Activity	**Duration**	**Frequency**	**Time-slice**	**Event**
Sleep Shower Toilet	high	high	yes	yes
Outside Watch TV Study/Work Telephone Change clothes Play piano Relax	high	low	yes	no
Prepare meal Have meal Brush teeth Shave	medium	high	yes	yes
Wash dishes	medium	low	yes	no
Snack Drink Take medicine	low	high	no	yes

models due to differences between the model and the true distribution of the data. However, generative methods are preferred when the size of the training data is limited, since they can exploit unlabeled data in addition to labelled data. When the size of annotated training data is large enough, discriminative models result in higher generalization performance [91]. Because of their differences, we selected one classifier from each category in order to generalize the evaluation. We use HMM and TWNN since they are well-suited for modeling the sequential nature of human activities.

A TWNN is an artificial neural network model we proposed as an extension to the time-delay neural networks (TDNN) [92]. TDNNs aim to capture the sequential nature of time series data by also feeding previous inputs delayed in time along with the input belonging to the targeted time instance. The sequentially aggregated input is then

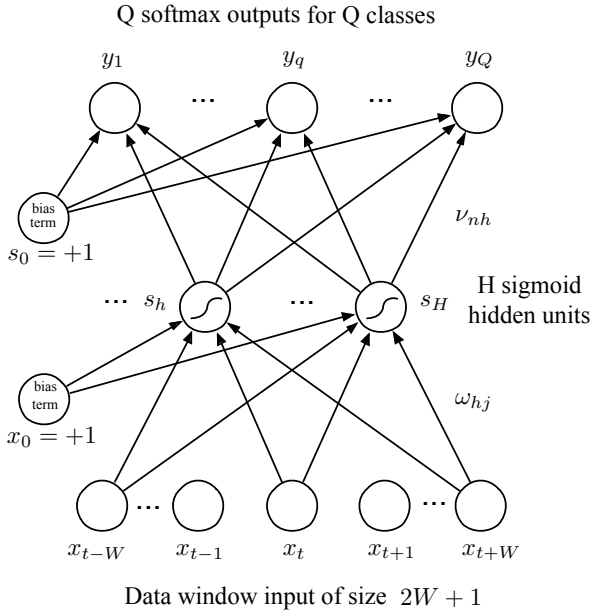


Figure 5.4. Time windowed neural network model.

fed into a feed-forward multilayer architecture which implements sigmoid activation units in its hidden layers. TWNN extends this idea by also incorporating the future inputs, thus constructing a time window around the targeted time instance. Such an approach is especially useful for human activity inference purposes, since utilizing the data related to the activities following a specific time instance can provide significant information on the activity performed at that particular time, due to the temporally dependent (both forwards and backwards in time) nature of human behavior. The TWNN model used in this work (with a single hidden layer) is depicted in Figure 5.4. The operation of TWNN is defined as:

$$s_h = \text{sigmoid}(\boldsymbol{\omega_h^T x}) = \frac{1}{1 + e^{-(\sum_{j=1}^{\Omega} \omega_{hj} x_j + \omega_{h0})}} \tag{5.3}$$

$$o_n = \boldsymbol{\nu_n^T s} = \sum_{h=1}^{H} \nu_{nh} s_h + \nu_{n0} \tag{5.4}$$

$$y_n = \text{softmax}(o_n) = \frac{\exp o_n}{\sum_i \exp o_i} \tag{5.5}$$

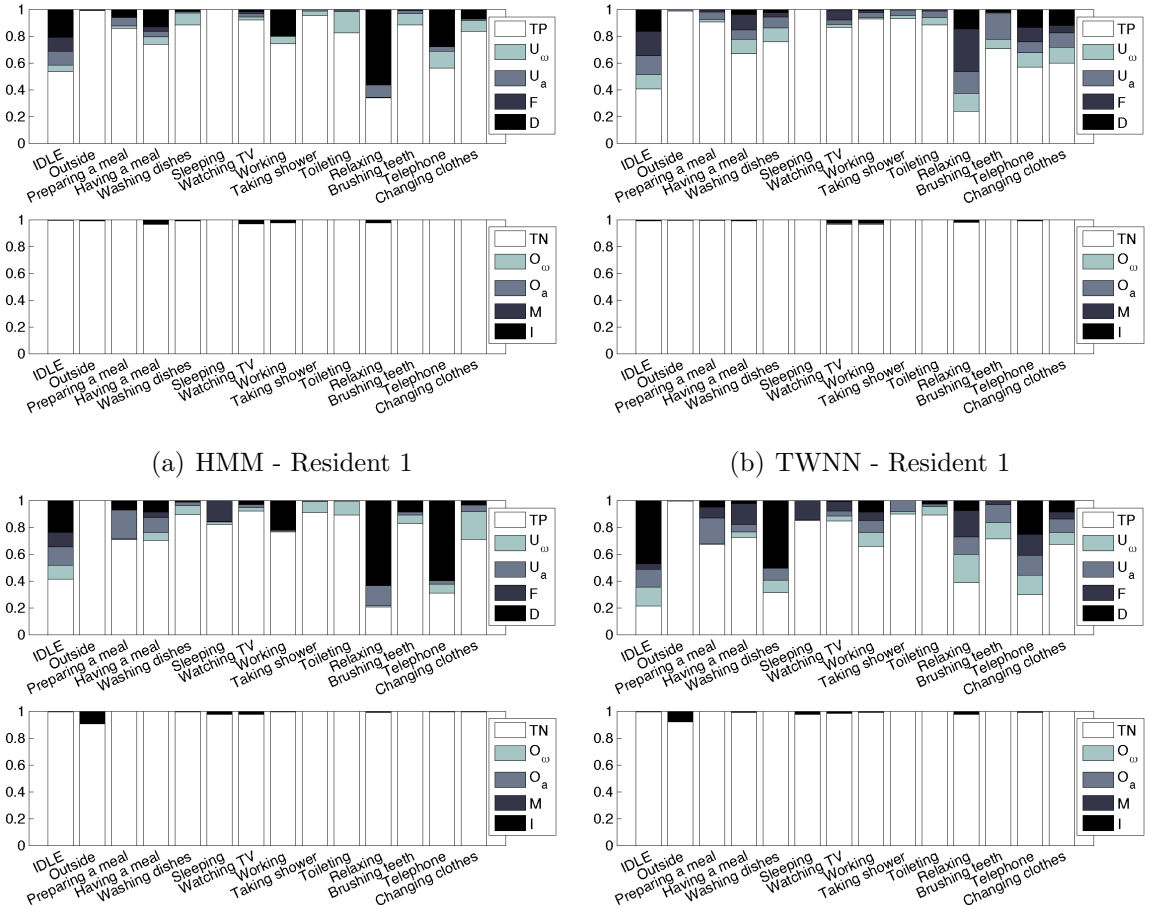
where x_j denotes individual features in a time window composed of $\Omega = (2W + 1) * N$ features where W is the half window size, N is the number of sensors, $\boldsymbol{\omega_h}$ is the first layer weights for the hidden unit h, $\boldsymbol{\nu_n}$ denotes the second layer weights for the output o_n, s_h denotes the output of the hidden unit h, and y_n denotes the output of the second layer. The softmax operator scales the output of the hidden layer ensuring that a single output is close to 1 and the other outputs are close to 0, thus acting as a selector among different classes.

For training the TWNN model, we use the back propagation algorithm. Online learning, for which individual instances of the training set are fed to the neural network in random order, is employed. The rate at which an individual instance contributes to the learning process is determined by the *learning factor* parameter. A random order pass over the whole training set denotes an *epoch*. Multiple epochs are performed to achieve good convergence. For the TWNN classifier, a single hidden layer model with 12 hidden units is constructed. The model is trained by performing 20 epochs over the training sets with the learning factor of 0.01. The window size is selected as 21 (corresponding to $W = 10$).

As a second classification method, we use the same HMM model we presented in Chapter 3. Data obtained from the sensors is transformed into time-slices of length $\Delta t = 60$ seconds. We split the data into a test and training set using a 'leave-one-day-out' approach. In this approach, one full day of sensor readings are used for testing and the remaining days are used for training. We cycle over all the days in the dataset, so that each day is used exactly once for evaluation.

5.4.2. Results

In this section, we evaluate the experimental results with our proposed methodology with a behavioral scope rather than only on a numerical score. We present the results for each resident in the houses of ARAS dataset individually in order to make proper comparisons about the classification methods experimented with.



(a) HMM - Resident 1

(b) TWNN - Resident 1

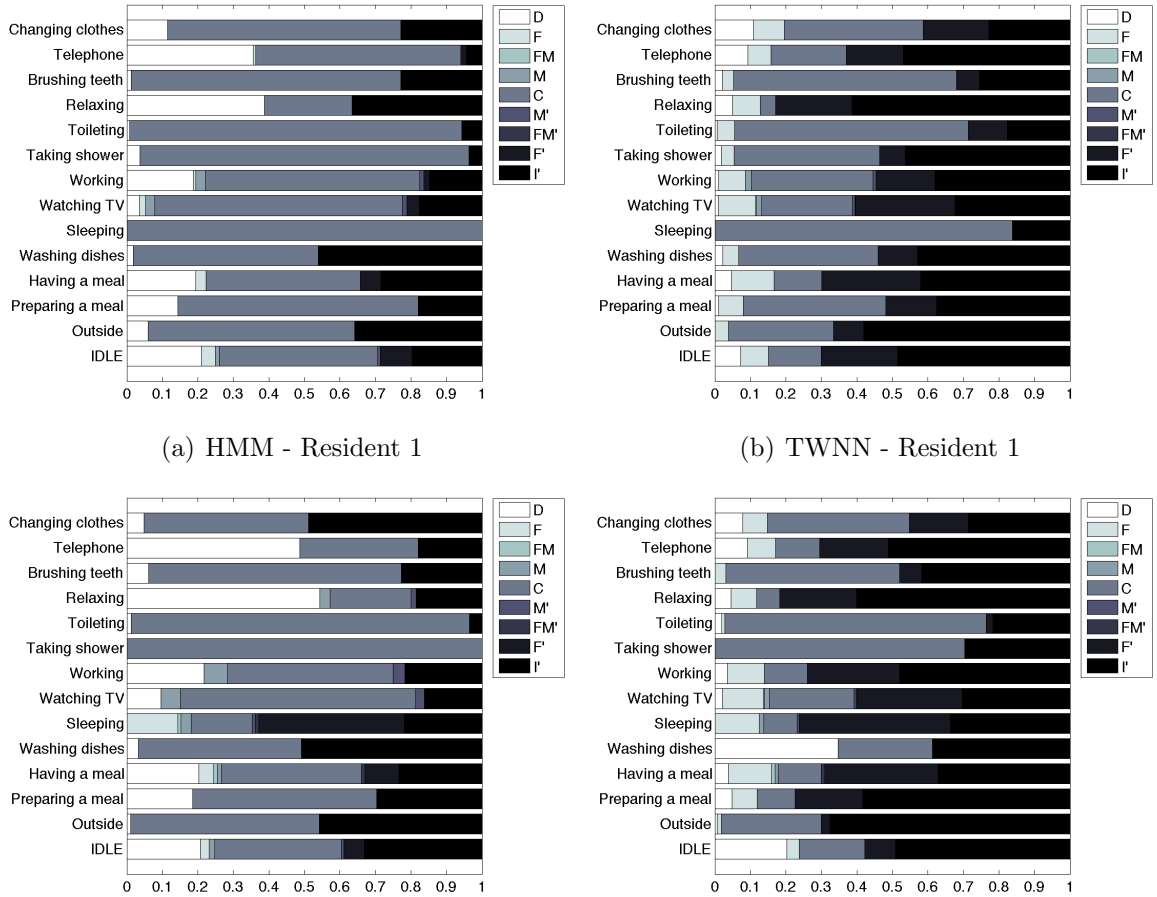
(c) HMM - Resident 2

(d) TWNN - Resident 2

Figure 5.5. Time-slice based performance evaluation of HMM and TWNN on ARAS House A.

5.4.2.1. HMM vs. TWNN on ARAS - House A. Time-slice based performances of all activities in House A for both residents are provided in Figure 5.5. For each resident, the graphs at the top depict the true positive ratio together with the false negative error types for each activity. Similarly, the graphs at the bottom provide the true negative ratio together with the false positive error types.

For the first resident, the false negative rates are rather low for both methods. In terms of true positive rates, on the other hand, there are significant differences between HMM and TWNN. For *relaxing* and *telephone* activities, HMM makes a higher number of deletions than TWNN. For TWNN, although the number of true positive time-slices are lower, deletion errors are significantly lower. Instead, TWNN makes timing and fragmentation errors. Since *relaxing* and *telephone* are duration sensitive activities,



(a) HMM - Resident 1

(b) TWNN - Resident 1

(c) HMM - Resident 2

(d) TWNN - Resident 2

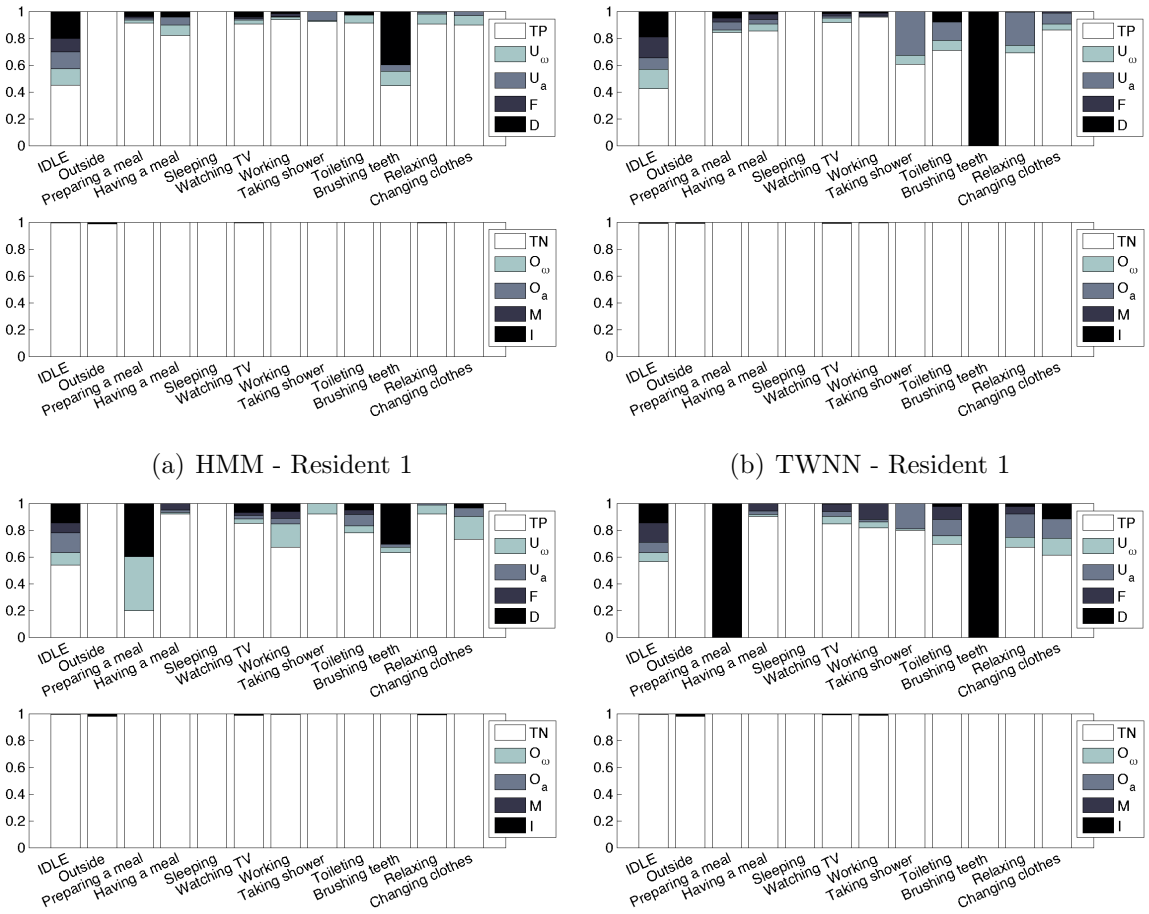
Figure 5.6. Event-based performance evaluation of HMM and TWNN on ARAS House A.

timing errors are important. For the *toileting* activity, we observe higher underfill errors with HMM. In general, the HMM classifier makes deletion errors on a time-slice level whereas TWNN makes mostly timing errors at the beginning and at the end of the activities. Also, fragmentation is observed frequently.

For the second resident, we observe the same pattern for *relaxing* and *telephone* activities. The tendency in timing and fragmentation errors rather than complete deletion errors persist for the second resident as well. Unlike the first resident, for the second resident, TWNN fails to capture the most time-slices for *washing the dishes* activity. For *working* activity, HMM makes deletion errors for both residents while with TWNN, it is possible to capture more time-slices correctly for the first resident and slightly less time-slices with timing errors for the second one. In terms of true negatives,

falsely inserted time-slices exists for the being *outside* activity for the second resident.

In order to evaluate the activity recognition performance from behavioral perspective, we also use the event level evaluations together with time-slice level metrics. The EADs of all activities in House A for both residents are provided in Figure 5.6.



(a) HMM - Resident 1

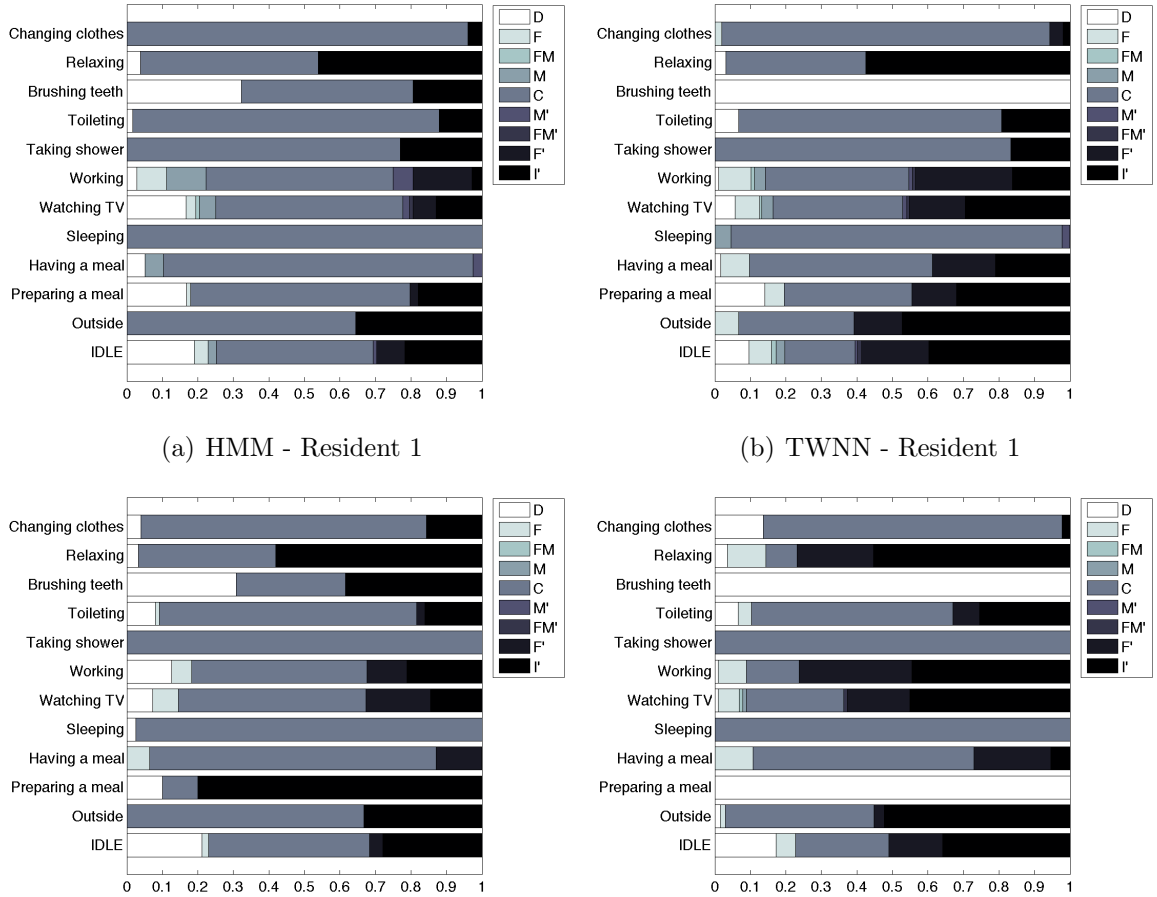
(b) TWNN - Resident 1

(c) HMM - Resident 2

(d) TWNN - Resident 2

Figure 5.7. Time-slice based performance evaluation of HMM and TWNN on ARAS House B.

In terms of correctly classified activity occurrences, HMM outperforms TWNN. The diagrams also suggest that the main error types for TWNN are fragmentation and insertion. For the frequency sensitive activities, TWNN would be worse choice for ARAS House A. One important observation is revealed when we compare the time-slice level performance with the event level performance of the *sleeping* activity. At the time-slice level, both HMM and TWNN performance metrics are extremely high for *sleeping* activity for both residents. However, the event level analysis suggests that the



(a) HMM - Resident 1

(b) TWNN - Resident 1

(c) HMM - Resident 2

(d) TWNN - Resident 2

Figure 5.8. Event based performance evaluation of HMM and TWNN on ARAS House B.

recognition for *sleeping* is more robust for Resident 1 when we use HMM since it does not make fragmentation errors for such a frequency and duration sensitive activity. For Resident 2, the *sleeping* activity is challenging even for the HMM because of the sensor failures. Yet, the correctly classified instances are larger in HMM.

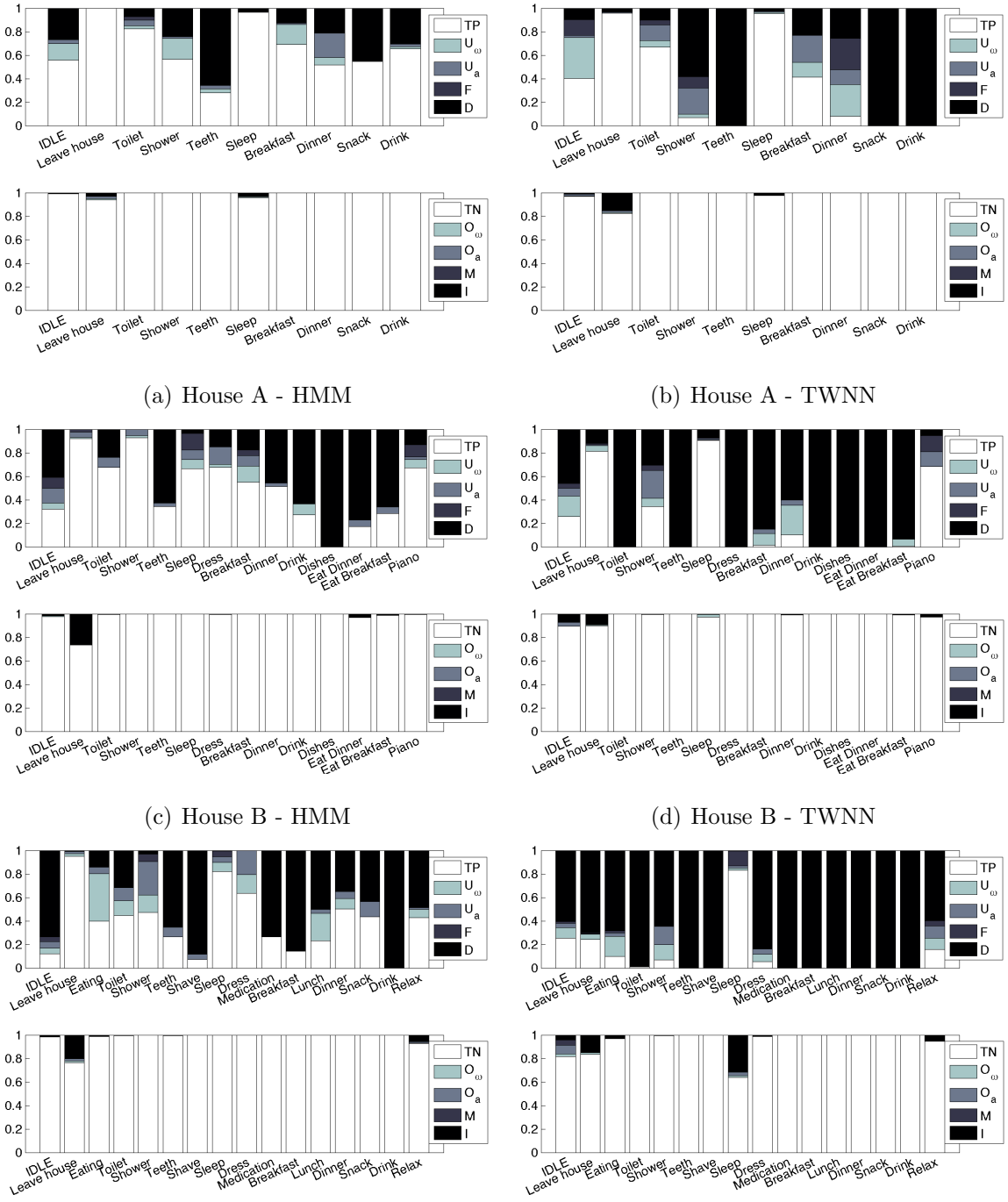
It can be concluded from the event based analysis that HMM outperforms TWNN on House A in terms of behavior recognition for well-being assessment purposes. When time-slice level analysis is also taken into account, it can be stated that TWNN fails to recognize the short duration activities efficiently and tends to fragment longer duration activities.

5.4.2.2. HMM vs. TWNN on ARAS - House B. In terms of time-slice based performances as depicted in Figure 5.7, the *brushing teeth* activity suffers the most for both residents. This is due to the lack of proper sensor for detecting this activity. Still, HMM succeeds in recognizing several time-slices correctly since it considers not only the sensor values combinations but also the transitions among the activities. For Resident 2, although there are too few occurrences of *preparing a meal* activity, with HMM we can still get several time-slices correctly although we underestimate the duration of the activity. TWNN fails to capture any time-slices. This supports the argument that states that generative models are better in terms of generalization when there are not enough training examples.

TWNN tends to make timing errors at the beginnings of *relaxing* and *taking shower* activities. For the second resident, *working* activity is more accurately captured by TWNN while HMM made tail underfill errors. In terms of true negatives, there are not any notable issues for House B.

At the second level of analysis, we consider the event analysis diagrams provided in Figure 5.8. One notable finding in EADs is for the second resident's *working* activity. On a time-slice level, we observe higher performance with TWNN, on the other hand, EAD for this activity indicates a much better performance in terms of occurrences. In general, TWNN method does perform as much as the HMM and makes more fragmentation and merging errors.

5.4.2.3. HMM vs. TWNN on Kasteren Datasets. Kasteren data sets are among the first examples of benchmarking data sets, which have been used in many studies with a variety of machine learning methods, but to the best of our knowledge, this is the first study that evaluates the datasets from a behavior recognition perspective. Time-slice based performance of all activities in Kasteren datasets are provided in Fig. 5.9 for HMM and TWNN methods. At time-slice level, the HMM performance is higher for shorter activities and the results are nearly the same for longer activities like *being outside* and *sleeping*. The general trend in performance across different houses in the



(a) House A - HMM

(b) House A - TWNN

(c) House B - HMM

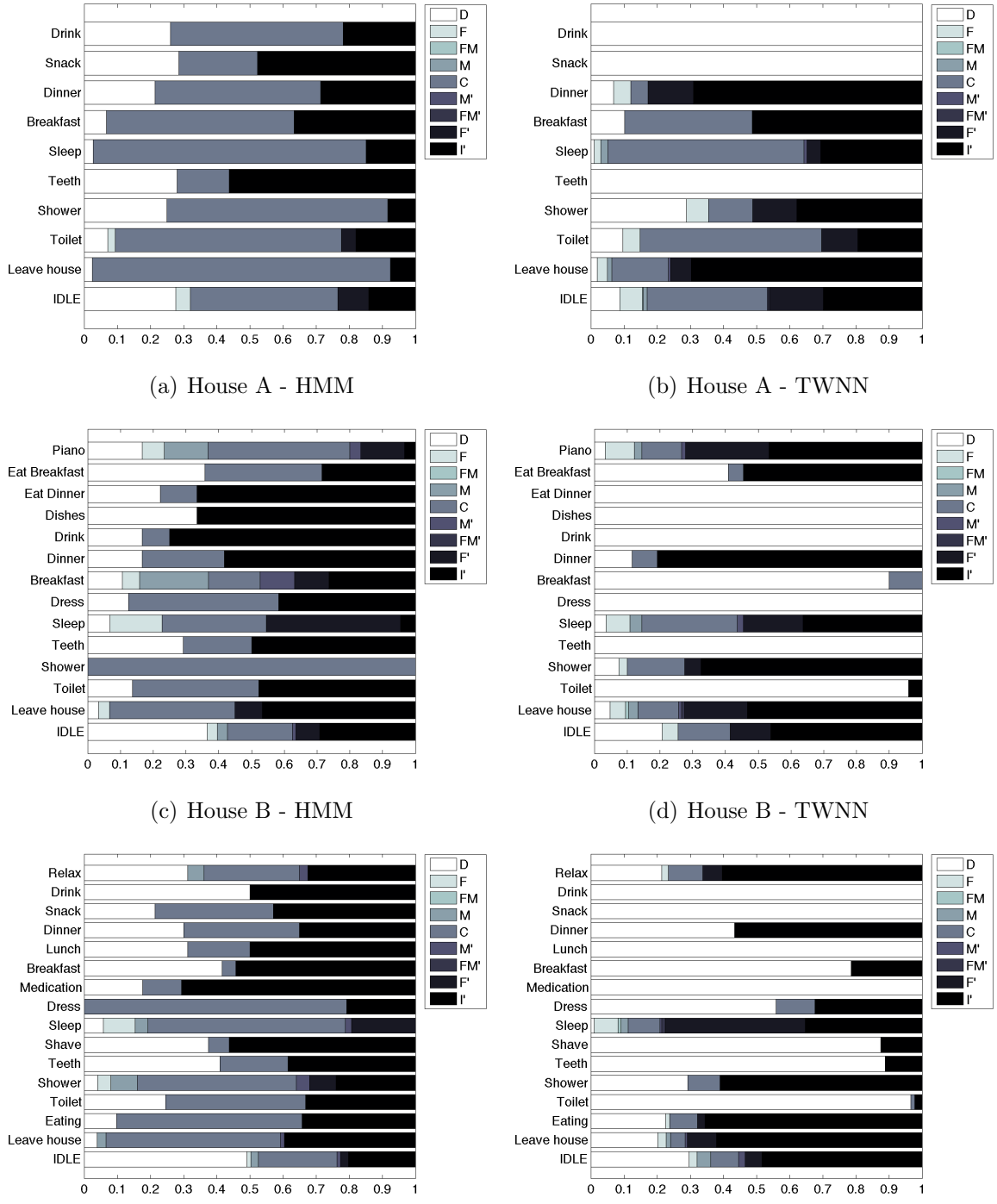
(d) House B - TWNN

(e) House C - HMM

(f) House C - TWNN

Figure 5.9. Time-slice based performance evaluation of HMM and TWNN methods on Kasteren datasets.

dataset is downwards, House C being the most challenging one. In House C, falsely inserted time-slices for *sleeping* activity has the highest ratio among all five datasets indicating a sensor failure or annotation accuracy problem.



(a) House A - HMM

(b) House A - TWNN

(c) House B - HMM

(d) House B - TWNN

(e) House C - HMM

(f) House C - TWNN

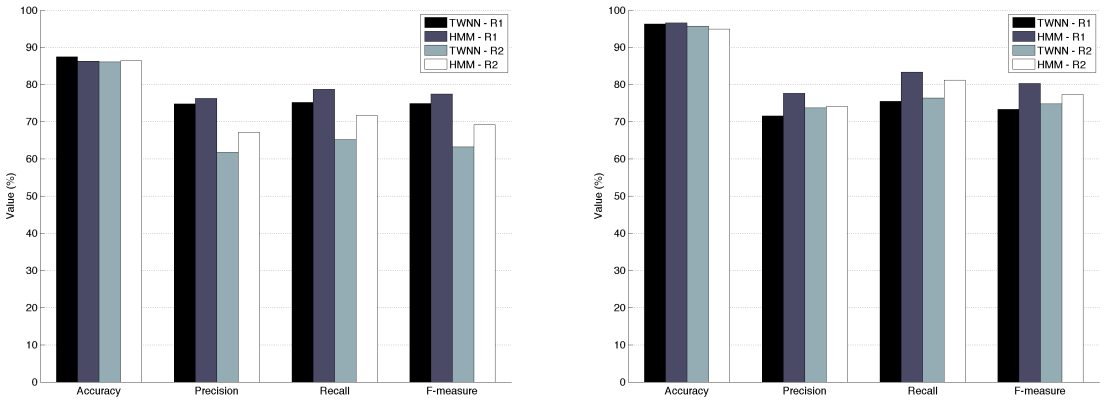
Figure 5.10. Event based performance evaluation of HMM and TWNN methods on Kasteren datasets.

The EADs of all activities in Kasteren datasets are provided in Figure 5.10. For House A, all of the activities can be recognized with HMM although with different accuracies. TWNN fails to capture any correct occurrence of *having a drink* or *snack* and *brushing teeth* activities in House A. For House B, *washing dishes* activity cannot

be captured by either method but the difference is that TWNN deletes the all instances whereas HMM also inserts wrong instances for *washing dishes* activity. The same holds for *having a drink activity* in House C. In this case, both methods yield 0% accuracy but from a behavior monitoring perspective, deletion of actual activities and insertion of false activities are more problematic for frequency sensitive activities than they are for duration sensitive activities. Hence, one method can be more preferable depending on both the application specific needs and the type of activities.

5.4.3. Comparison with Conventional Evaluation Metrics

In this section, in order to stress on the shortcomings of the conventional metrics, we compare the performance of the HMM and TWNN classifiers on the ARAS datasets in Fig. 5.11. Consider ARAS House A as an example. TWNN yields higher accuracy than HMM for resident 1, however, our behavior oriented evaluation showed the opposite. TWNN fails to capture most occurrences of frequency sensitive activities. For the particular activity of *sleeping* which is both frequency and duration sensitive, HMM outperforms TWNN. Moreover, TWNN cannot successfully recognize short duration activities and tends to make fragmentation errors on longer duration activities. Despite these deficiencies, if we were to consider the accuracy metric only, we would argue that it performed better than HMM. Similarly, for ARAS House B, the performance of the two methods are nearly equal in terms of accuracy. However, behavior oriented evaluation indicates that TWNN suffers from similar shortcomings.



(a) House A (b) House B

Figure 5.11. Performance evaluation of ARAS datasets using standard metrics.

5.5. Conclusion

In this chapter, we addressed the challenges in the evaluation of different approaches for the purposes of human behavior understanding through a well-being assessment perspective. The metrics used in the performance evaluation of newly developed algorithms are directly taken from the machine learning domain. The shortcomings of the use of general purpose metrics are demonstrated with experiments on real world data. Human behavior analysis from a medical perspective requires analysis of daily activities in terms of timing, duration and frequency. Given the high variations in these attributes for different activities, the general purpose metrics fail to accurately reflect the actual performance. We proposed an evaluation method generally applicable to real world applications that require human behavior understanding. In the proposed method, we first group the activities of daily living in terms of their duration and frequency sensitivities. Then, we map the categories to appropriate evaluation strategy using either time-slice level or event level criteria. In this way, we provide sounder evaluation criteria rather than a one-size-fits-all approach, i.e. using the same single metric for all types of activities. Using the newly proposed method, we compared the performance of two machine learning models, HMM and TWNN, on five different real world datasets from a behavior monitoring perspective. The results with real world human behavior data revealed that the use of standard metrics can be misleading in demonstrating the performance from a behavior understanding perspective.

Conventional metrics such as accuracy and f-measure are widely used for evaluation purposes because of their compactness. Yet, this compactness causes a loss in the human behavior perspective when applied to assessment of well-being in AAL systems. There exists a trade-off between compactness and informativeness. Since the human behavior understanding for healthcare monitoring purposes requires delicacy, we propose trading some of the compactness with informativeness to obtain deeper insights.

6. MULTI-RESIDENT ACTIVITY TRACKING AND RECOGNITION

6.1. Introduction

Most previous studies on human activity recognition in smart house assume a single resident inside the house. The studies that can handle the multiple residents generally assume a location identification mechanism such as RFID that allows the system to differentiate between the sensor readings for each resident. Both of these assumptions are too restrictive that they prevent the general applicability of activity recognition systems. In this chapter, we focus on making smart houses smart enough to provide long term health monitoring for not only people who live alone but also with a spouse or a flat mate. In that respect, we propose methods to recognize the individual behaviors in multi-resident environments without assuming any person identification which generally requires the use of wearable technology that can be obtrusive. We propose two different methods for handling the multiple resident case. First, we directly model the overlaid observations together with multiple chains of activity sequences using a factorial hidden Markov model (FHMM) model. Secondly, we use nonlinear Bayesian tracking for decomposing the observation space into the number of residents. Specifically, we focus on multiple target tracking problem for data association purposes, rather than determining the exact coordinates of the residents inside the house. We use a particle filter (PF) together with a joint probability data association (JPDA) method for assigning the sensor readings to multiple residents. For each method, we perform experiments on real-world data sets and discuss the advantages and disadvantages of each approach in detail.

The rest of this chapter is organized as follows. In Section 6.2, we give a brief literature review on FHMM and Bayesian tracking methods and data association methods used for multiple target tracking. In Section 6.3.1, we describe the FHMM we use, and in Section 6.3.2 we provide the details for our PF approach together with the proposed

data association mechanism. Section 6.4 gives the details of our experiments with real world data together with a detailed discussion in which we make comparisons between the two different approaches we proposed. Finally, we conclude with Section 6.5.

6.2. Related Work

Given the additional complexity of multi-resident activity recognition, there are only few studies tackling this problem. In [93], the authors collect a dataset for multi-resident activity recognition in a controlled laboratory environment using a set of activities performed following a predefined scenario. Using the pre-segmented dataset, they report an average accuracy of 60.6% for 14 activities. When they assume they knew the sensor-resident identity matching, the accuracy is 73.1%. In [94], a multi-person activity recognition study using computer vision is given. They use a feature selection mechanism in order to decompose the observation space, then they use a HHMM for activity recognition. Instead of combining all features into one single vector, they use subgroups of features for different people. They propose a feature selection and weighting mechanism to come up with a correct assignment of features to people.

Wilson and Atkeson [95] are the first to propose simultaneous activity recognition and recognition using a discrete Bayesian filter. They solve the data association problem for multiple users by Rao-Blackwellised particle filter. Unfortunately, they only report results on synthetically generated data. Their results on simulated data yields 98% accuracy for two people and 85% accuracy for three people.

The non-linear Bayesian techniques for tracking targets has many military application and has a long history [96]. One of the major problems in multi-target tracking is the data association, that is when there are multiple sensor readings and multiple targets, we need to make a mapping between the sensor readings and the targets in order to improve the tracking accuracy. Several classical data association methods exist [97]. The simplest is the method which uses only the closest observation to update the measurements. When there are too many sensor readings, the evaluation time can be long. In that case, gating mechanisms are applied such that, an observation may

only be used for the update if it is within an error tolerance area around the estimated target called the *gate*. Another widely used multi-target tracking association method is the JPDA which is an extension of the probability data association algorithm to multiple targets [98]. It estimates the states by a sum over all the association hypothesis weighted by the probabilities from the likelihood. The most computationally intensive algorithm for data association is called the multiple hypothesis tracking (MHT), which calculates every possible update hypothesis [99]. In MHT, since we keep track of every possible hypothesis for every time-step, the number of tracked hypothesis grow exponentially making the method intractable very quickly. In [100], a pruning mechanism is proposed so that the unlikely hypothesis are dropped.

Factorial hidden Markov models (FHMM) were introduced by Ghahramani and Jordan [101] and have been used in several domains such as speech recognition [102], bioinformatics [103] and computer vision [104]. Although being an efficient representation for the indirect and complex interactions among multiple separate Markov chains through a common observed variable, the additional complexity of training such models prevent them to be widely used in many other domains. To the best of our knowledge, this is the first study that uses an FHMM for the human activity recognition problem with multiple residents.

6.3. Multi-Resident Activity Recognition Methods

We present two different approaches to multi-resident activity recognition problem. First, we propose a direct modeling approach, that is, we use a FHMM with two independent chains corresponding to each resident's activities and a common observed variable corresponding to the sensor readings. Secondly, we use a PF approach in order to decompose the observation space into two, i.e. one for each resident. We consider a multi-target tracking problem and solve the data association problem using a JPDA approach. In the following subsections, we give the details of the FHMM model and the nonlinear Bayesian tracking approach we use for observation decomposition.

6.3.1. Factorial Hidden Markov Model

FHMM is a generalization of HMM in which there are multiple independent Markov chains of states and the observation distribution at a given time step is conditioned on all of the corresponding state variables in each chain at that time step [101]. Although, the chains are a priori independent, the observation of the common variable makes them correlated, bringing an additional complexity in training. We model each resident's activity sequences separately with a FHMM with $E = 2$ chains with Q different states as depicted in Figure 6.1. We define the joint probability as follows:

$$p(x_{1:T}, y^1_{1:T}, y^2_{1:T}) = \prod_{t=1}^{T} p(\vec{x}_t \mid y^1_t, y^2_t) p(y^1_t \mid y^1_{t-1}) p(y^2_t \mid y^2_{t-1}) \tag{6.1}$$

where $p(y^e_1 \mid y^e_0) = p(y^e_1) = \pi^e$ is defined as the initial state distribution for the chain e. Likewise, $p(y^e_t \mid y^e_{t-1}) = A^e$ is the state transition matrix for chain e. In this way, each chain of state variables are allowed to evolve according to its own dynamics having E distinct Q x Q transition matrices denoted by A^e. This is different from having a flat HMM using a cross product of the state variables, i.e., having a single transition matrix with Q^E x Q^E entries. The observation model is a linear Gaussian model in which the N dimensional observation vector $\vec{x}$ is Gaussian and its mean is a linear function of

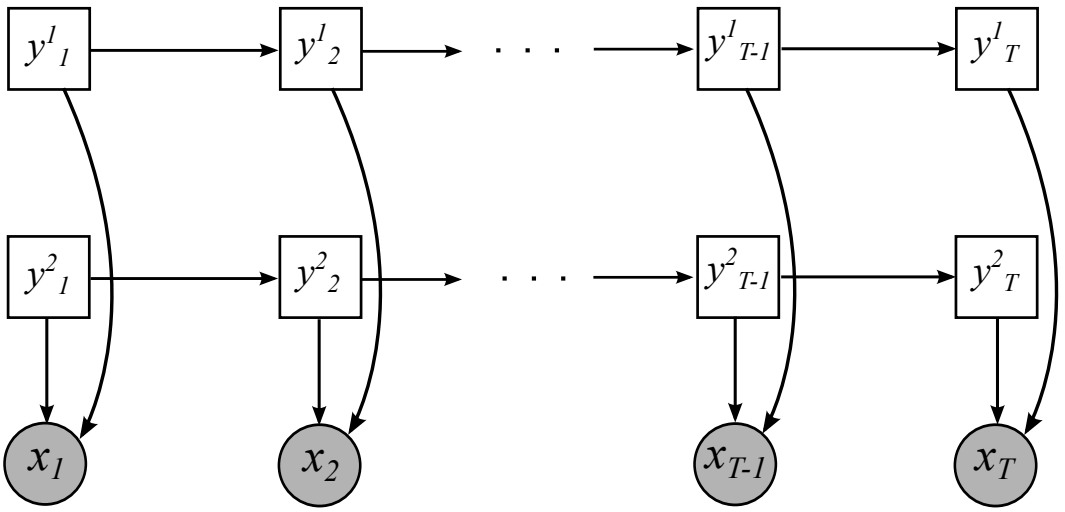


Figure 6.1. The graphical representation of a FHMM. Shaded nodes represent observable variables, the white nodes represent hidden states.

the state variables represented with a 1-of-Q encoding of y_t^e.

$$p(\vec{x}_t \mid y_t^e) = \mathcal{N}(\vec{x}_t, \mu_t, \Sigma) \tag{6.2}$$

where

$$\mu_t = \sum_{e=1}^{E} \mathbf{W}^e y_t^e \tag{6.3}$$

Each $\mathbf{W}^e$ matrix, whose columns represent the contributions to the means for each state configuration y_t^e, is N x Q. Σ is the N x N covariance matrix. The entire model is parametrized by $\psi = \{\mathbf{W}^e, \pi^e, A^e, \Sigma\}$.

Even though each chain is a priori independent, they become coupled in the posterior due to having an observed common child, x_t. This coupling makes the exact inference of the FHMM model intractable. As with the HMM, the parameters of an FHMM can be estimated via the EM algorithm. In the E-step, we compute the posterior probabilities of the hidden states in an FHMM, followed by the M-step which is simple and tractable. If we use the naive exact algorithm which requires translating the FHMM into an equivalent flat HMM with Q^E states followed by the execution of forward-backward procedure, the complexity of the procedure is $O(TQ^{2E})$, where T is the length of the sequence. In our case, with $E = 2$ chains, the complexity is $O(TQ^4)$. Instead, by exploiting the graph structure of the FHMM, we can use the junction tree algorithm that also provides an exact E-step but with lower time complexity. When we moralize and triangulate the graph, we obtain a junction tree with $T(E + 1) - E$ cliques of size $E + 1$. The junction tree algorithm on this model has $O(TEQ^{E+1})$ time complexity. In our case, the time complexity is $O(2TQ^3)$. For smaller models, the exact inference is achieved within a reasonable amount of time, however, for larger models with $E > 2$ chains, both algorithms become intractable. Therefore, several approximate inference methods have been proposed in the literature such as structural variational approximation, factorized variational approximation and Gibbs sampling. Since our model is relatively small, we use the junction tree algorithm for the exact

inference. The exact inference algorithm we use is given in Appendix A. Nevertheless, the model is generalizable to multiple resident activity recognition problem having more than two residents by employing approximate inference methods such as variational approximation or Gibbs sampling [101].

6.3.2. Nonlinear Bayesian Tracking

We use a state-space approach for modeling the time-series data in a nonlinear Bayesian tracking setting. In state-space modeling, the state vector contains all the relevant information required to describe the system. The measurement vector represents the noisy observations related to the state vector. We capture the dynamics of the system using two models. *The system model* describes the evolution of the state with time. *The measurement model* relates the noisy measurements to the state. When these models are represented as probabilistic functions, we can use the Bayesian approach to find the posterior probability density function (pdf) of the state vector based on the set of available measurements. Also, we can use a recursive approach in order to process data sequentially when a measurement is received. Such a filtering mechanism is composed of two stages.

- *Prediction stage:* We use the system model to predict the state pdf one step forward in time, i.e., before we actually receive the measurement. Since this prediction is subject to noise, the predicted pdf generally spreads and deforms.
- *Update stage:* We use the latest measurement to modify the prediction pdf of the state and obtain the posterior pdf using the Bayes theorem.

Generally, it is not possible to calculate the full posterior analytically. Therefore, sequential Monte Carlo (SMC) approaches are employed in order to approximate the optimal Bayesian solution. In this chapter, we use a sequential importance sampling (SIS) algorithm also known as *bootstrap filter*, for implementing a recursive Bayesian filter. The main idea in SIS is to approximate the full posterior distribution $p(x_{0:k-1}|z_{1:k-1})$ at time $k-1$ with a weighted set of samples called the particles, $P = \{x^i_{0:k-1}, w^i_{k-1} : i = 1, .., N\}$, and recursively update these particles and their

weights to obtain an approximation to the posterior distribution $p(x_{0:k}|z_{1:k})$ at time k. To maintain a consistent sample, the new importance weights are set to w_k^i as follows:

$$\begin{aligned} x_k^i &\sim q(x_k|x_{k-1}^i, z_k) \\ w_k^i &\propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \end{aligned} \tag{6.4}$$

In the bootstrap filter, we choose the importance density $q(x_k|x_{k-1}^i, z_k)$ as the prior density $p(x_k|x_{k-1}^i)$ and we approximate the posterior filtered density $p(x_k|z_{1:k})$ by the following discrete representation

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \tag{6.5}$$

where $\delta(.)$ is the Dirac delta measure. This implementation of the SMC method corresponds to the bootstrap filter as proposed in [105]. The full derivation of the bootstrap filter is given in Appendix B. The overall procedure for the bootstrap filter is given in Figure 6.2.

We use a grid structure for modeling the house layouts. We use one meter square grids since our purpose is not the exact location estimation but the sensor-resident association only. In our particle filter, the system model, $p(x_k \mid x_{k-1})$, is a random walk model with α movement probability. Between the two consecutive time-steps, the target either maintains its location or moves in one of the four neighboring grids. The measurement model is Gaussian around the actual state, $p(z_k \mid x_k) \sim \mathcal{N}(z_k; x_k, R_m)$. While evaluating the distances, we cannot use Euclidian distances since we need to take into account the architectural constraints of the house, such as walls and doors. For that reason, we model the house layout as a maze structure using a graph, and evaluate the distances between any two points as the shortest path distance in the graph using the breadth first search (BFS) algorithm.

Require: State model $p(x_k \mid x_{k-1})$

Measurement model $p(z_k \mid x_k)$

Set of N_p particles $\{x_{k-1}^i\}$

Ensure: $TotalWeight \leftarrow 0$

{Move particles}

for $i = 1$ **to** N_p **do**

$x_k^i \leftarrow p(x_k \mid x_{k-1}^i)$

$w_k^i \leftarrow p(z_k \mid x_k^i)$

$TotalWeight \leftarrow TotalWeight + w_k^i$

end for

{Normalize weights}

for $i = 1$ **to** N_p **do**

$w_k^i \leftarrow w_k^i / TotalWeight$

end for

{Resample according to weights w_k^i}

$c_1 = 0$ {Construct CDF}

for $i = 2$ **to** N_p **do**

$c_i = c_{i-1} + w_k^i$

end for

$i = 1$ {Start at the bottom of the CDF}

$u_1 \sim \mathcal{U}[0, 1/N_p]$ {Draw a starting point}

for $j = 1$ **to** N_p **do**

$u_j = u_1 + (1/N_p)(j-1)$ {Move along the CDF}

while $u_j > c_i$ **do**

$i = i + 1$

end while

$x_k^j \leftarrow x_k^i$ {Assign sample}

end for

Figure 6.2. SIR particle filter algorithm.

In order to make the matching between the sensor readings and the residents more accurate, we use a set of heuristics in the JPDA algorithm. These heuristics are described as follows:

- *Sensors are categorized into two groups as single occupancy sensors and multiple occupancy sensors.* Examples of single occupancy sensors are pressure mats, contact sensors, photocells. These sensors can only be assigned to a single user at a given time-step. In the second group, there are sensors that can be assigned to more than one resident. The examples of such sensors are infrared sensors that can sense the use of the remote controller for the TV and motion and presence sensors that can detect multiple people.
- *For any given single occupancy sensor-resident matching at the previous time-step, the same matching persists in the following time-steps until the sensor value changes.* This heuristic ensures that, once a resident is assigned to a single occupancy sensor such as a pressure mat on the couch, she/he stays there until the sensor stop firing. In other words, if you are sitting on the couch, you have to get up first before that couch can be used by other people again.
- *There can be favorite sensor-resident matchings known apriori.* This heuristic is used only for tiebreaking purposes. For example, when both residents are known to be in the bedroom that contains two pressure mats on the bed corresponding to each resident's preferred side, we feed this information into the tracking algorithm to make the correct assignment out of the two equally likely assignment.

These straightforward yet realistic heuristics are easily integrated into the probabilistic data association algorithm for increasing the accuracy of the assignments. In this way, the complexity of the association problem, which would be very high otherwise, is reduced.

6.4. Experiments

In this section, we present the experiments and their results for the comparison of the two proposed approaches described in the previous section, for the multiple resident

human activity recognition. We conduct one experiment for each of the methods. In the first experiment, we aim to reveal the performance of the FHMM approach to multiple resident activity recognition. The second experiment is designed for evaluating the performance of the proposed tracking based observation decomposition technique. Finally, we provide a comparison between both approaches and discuss their advantages and disadvantages.

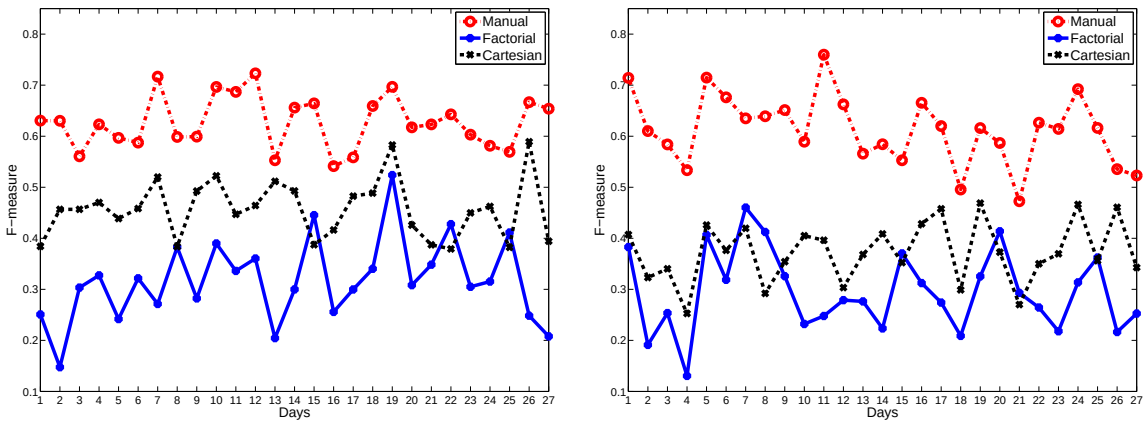
We use the ARAS datasets in all experiments. We use leave-one-out cross validation approach. We measure the recognition performance on a time-slice level using the f-measure, which is the harmonic mean of precision and recall values. We also use the event based evaluation since in the previous chapter, we already showed the benefit of using an event based evaluation in terms of revealing the actual performance in the behavior identification level.

6.4.1. Experiment 1: Direct Modeling Techniques

In this experiment, we use a FHMM with two independent chains, each having Q states and a single continuous observed variable. In order to achieve this representation using multiple sensors, we transform the data into $\Delta t = 60sec$ bins by taking non-overlapping sliding windows and normalizing the count of each observation. In that way, we obtain a single multivariate Gaussian variable in order to preserve the compactness of the representation in FHMM. Using the same feature extraction mechanism, we compare the FHMM model both with a naive approach using a single chain HMM with the cartesian product of the state space. We refer this approach as the cartesian HMM, which is a single layer HMM with Q^2 states. Secondly, in order to provide a more thorough comparison, we also experiment with the manually separated observations using the same Gaussian HMM model. For all models, we use a maximum likelihood approach in training and the inference is made using the Viterbi algorithm.

The results for the daily average recognition results in terms f-measure are given in Figure 6.3 for House A. The average f-measure performance for the first resident is 62.7% when we use the manually decomposed HMM. We obtain an average of 31.7%

with FHMM and 45.6% with a cartesian HMM. For the second resident, the average f-measure performances are 61.2%, 29.5%, and 37.3% for the manually decomposed HMM, factorial HMM and cartesian HMM, respectively.



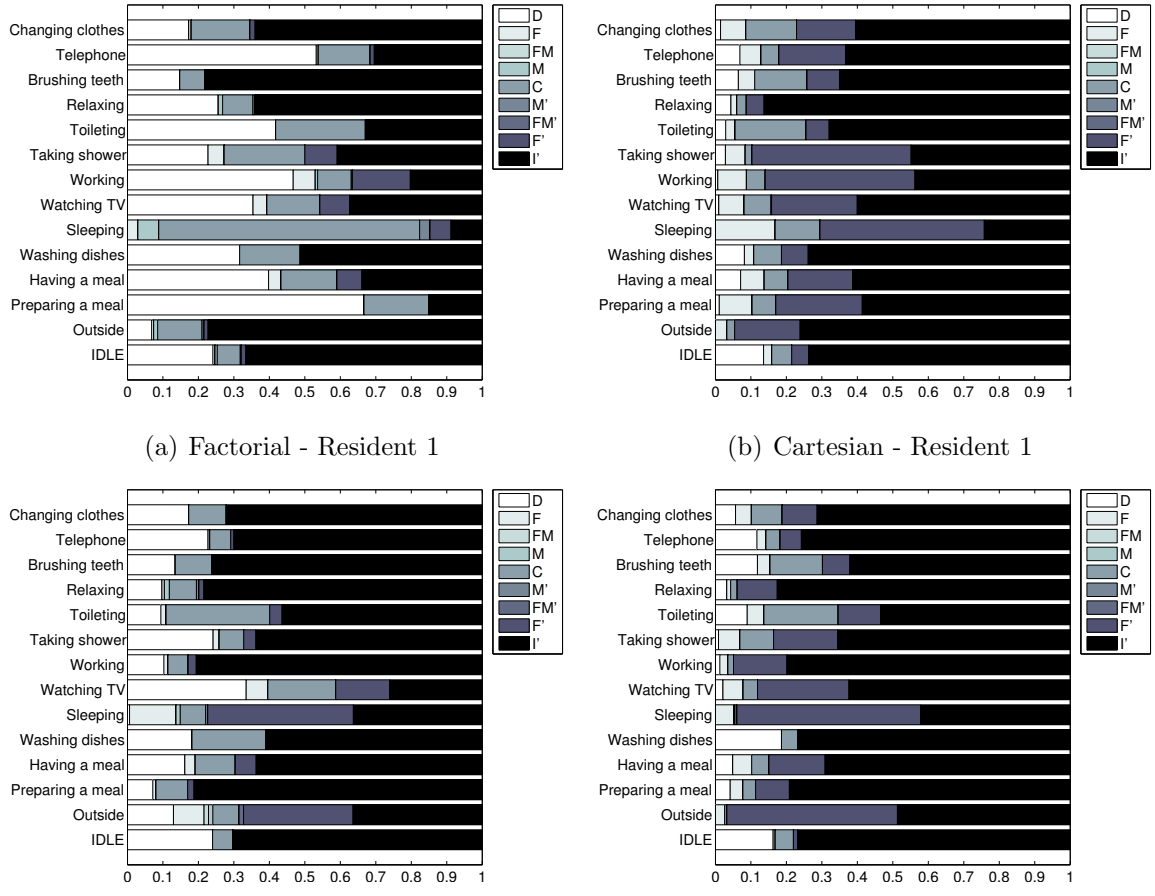
(a) Resident 1 (b) Resident 2

Figure 6.3. Daily average activity recognition performance of direct modeling techniques in terms of f-measure in ARAS House A.

According to the results, for both residents in House A, neither factorial nor cartesian HMM perform as good as the manually decomposed version. The performance degradation can be attributed to the lack of enough training data. As the number of states increases, the needed training data for efficient learning of the HMMs increase. While it is expected that the cartesian approach, having a larger number of states, performs worse than the factorial HMM, we observe a higher performance with the cartesian approach in terms of time-slice level average f-measure performance for all activities.

Since we pointed out in the previous chapter, the time-slice based metrics fails to represent the actual performance in the application level that concerns the human behavior in terms of frequency, start time and duration. For that reason, we also provide the results with a human behavior identification perspective using EADs. The EADs for House A are given in Figure 6.4 for both approaches and for both residents. According to the results, we observe that the time-slice based performance increase in the cartesian HMM comes with severe fragmentation errors at the activity occurrence level. In terms of correctly classified activity occurrences, FHMM performs better than

the cartesian HMM for both residents.



(a) Factorial - Resident 1

(b) Cartesian - Resident 1

(c) Factorial - Resident 2

(d) Cartesian - Resident 2

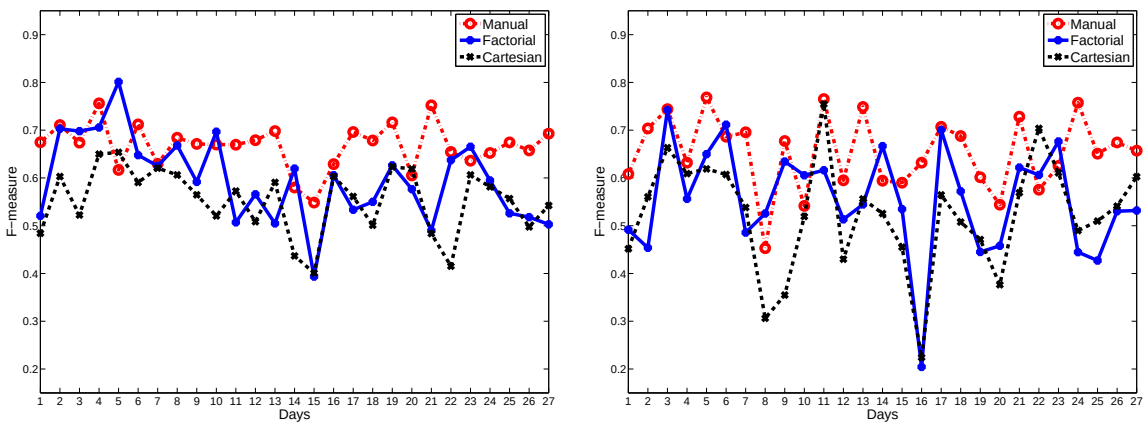
Figure 6.4. Event-based performance evaluation of factorial and cartesian HMM on ARAS House A.

Consider the sleeping activity for example. With a factorial HMM, 75% of all sleeping activities for the first resident are perfectly classified without any fragmentation or merging type errors. With a cartesian HMM, we only classify 10% of the all sleeping occurrences. Nearly half of the sleeping occurrences suffer from fragmentation errors. Similarly, 20% of the occurrences suffers from both fragmenting and merging type of errors. Since sleep is a duration and frequency sensitive activity, if we use a cartesian HMM, our activity recognition system will continuously report sleep related problems for the residents, although this is not the case. The same behavior is observed for all of the activities for both residents. The fragmentation errors with the cartesian HMM stems from the fact that the model needs to learn a different version of a single

activity for each and every activity of the other resident. For the FHMM case however, the a priori independence of the chains prevents this state explosion phenomenon. In summary, although the time-slice based f-measure suggests otherwise, using FHMM rather than a cartesian HMM is more beneficial in identifying multiple resident activity recognition.

The results for the daily average recognition results in terms f-measure are given in Figure 6.5 for House B. Unlike House A, the time-slice level performance of FHMM is higher than the cartesian HMM. Also, the performance gap between the manually decomposed HMM is much smaller. For the first resident, the average f-measure performance is 66.7% with the manually decomposed HMM. FHMM yields an average f-measure of 59.5% and cartesian HMM performance is 55.2%. For the second resident, the average f-measure performances are 65.3%, 55.4%, and 52.3% for the manually decomposed HMM, FHMM and cartesian HMM respectively. These results can be attributed to the fact that the sensor data patterns for different residents' activities for this house are more easily distinguishable with the training data available.

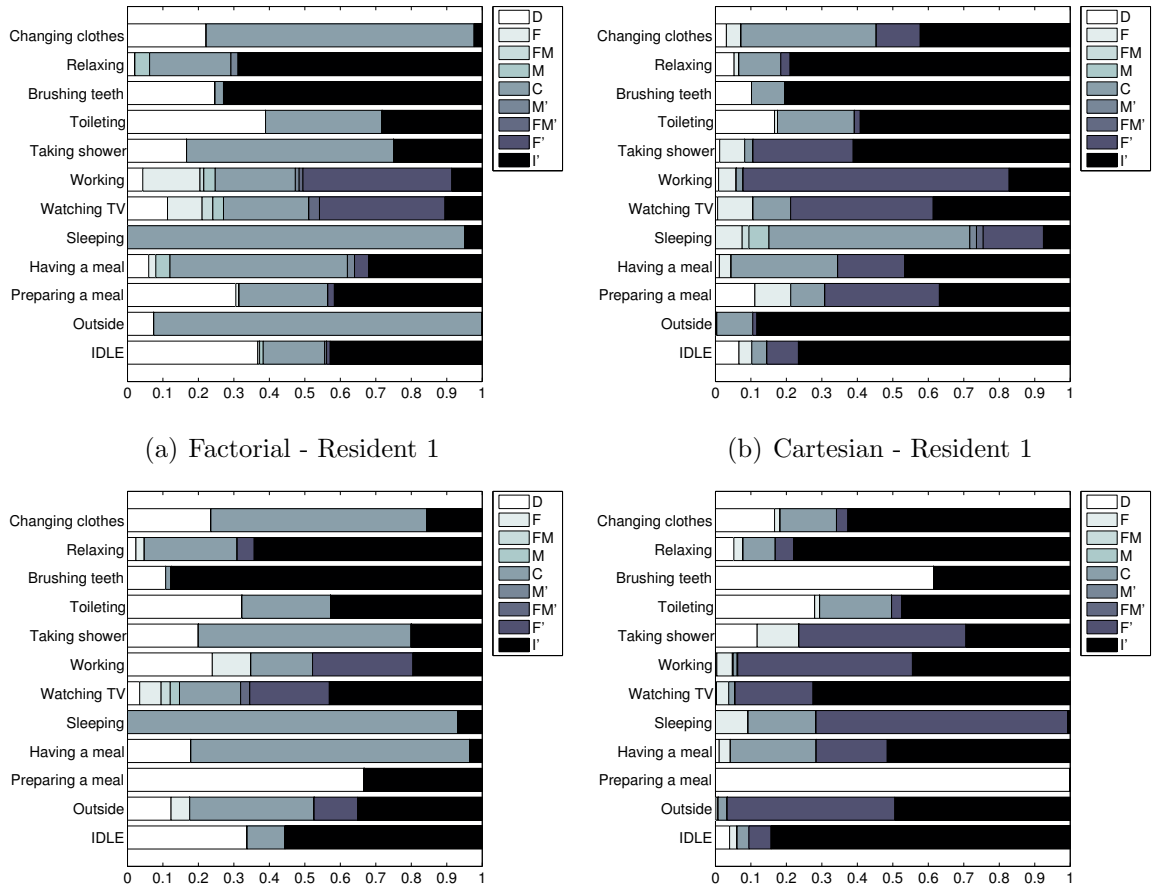
We give the occurrence level performances of all activities for both residents in Figure 6.6. For House B, we observe the same behavior as in House A when we compare the FHMM with the cartesian HMM. The cartesian HMM suffers from severe



(a) Resident 1 (b) Resident 2

Figure 6.5. Daily average activity recognition performance of direct modeling techniques in terms of f-measure in ARAS House B.

fragmentation related errors for most of the activities. In terms of correctly recognized activity occurrence percentages, FHMM is much higher than the cartesian HMM for both houses and both residents.



(a) Factorial - Resident 1

(b) Cartesian - Resident 1

(c) Factorial - Resident 2

(d) Cartesian - Resident 2

Figure 6.6. Event-based performance evaluation of factorial and cartesian HMM on ARAS House B.

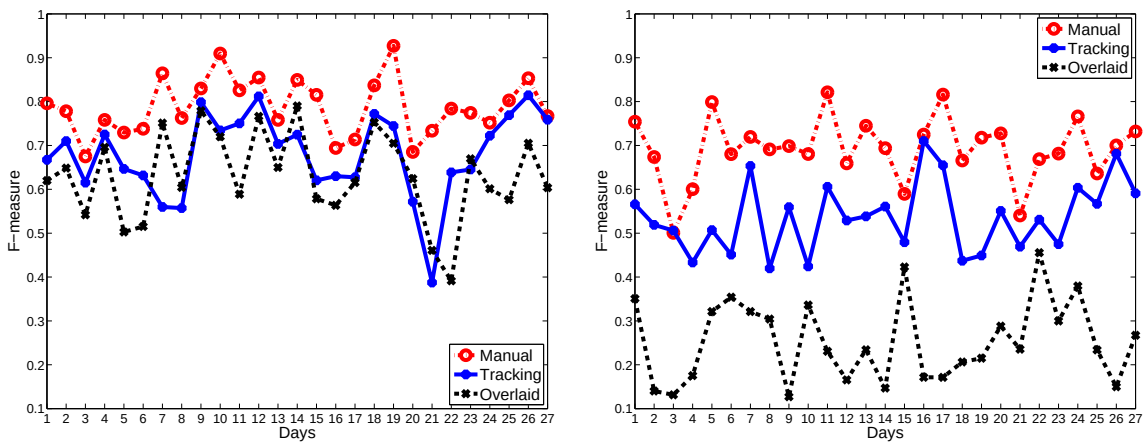
6.4.2. Experiment 2: Observation Decomposition

In this experiment, we use the JPDA method together with the PF tracking mechanism as described in Section 6.3.2. In this way, we aim to make accurate sensor resident matching and decompose the observations into two. After this decomposition, we use a separate HMM for each resident for recognizing the activities. We compare this approach with an overlaid approach. In the overlaid approach, we do not separate the observations into two, instead we use the same observations for both residents

as they are. In other words, we treat the sensor firings caused by the other resident's activities as noise for each other. Finally, we report the performance using the manually separated observations in order to allow a more comprehensive comparison.

In the tracking experiments we use $N_p = 100$ particles for House A, and $N_p = 200$ particles for House B. The system model is a random walk with a movement probability of $\alpha = 0.5$. The process noise variance for the particle filter is $R_p = 5$ and the measurement noise variance is $R_m = 1$. Since we use a SMC approach for approximation, different runs of the observation decomposition algorithm can result in different assignments across different runs. For that reason, we repeat the particle filter algorithm ten times and report the average performance achieved. Also, since we do not assume any explicit identification mechanism, it is not always possible to determine which residents are in the house or which residents are out. For that reason, we assume that we know the number of residents in the house and in cases where only a single resident is present in the house, we assume to know the identity of the resident in the house. With the recent penetration rates of the mobile phones, it is not difficult to obtain this information using smart phone location services.

The experimental results in terms of daily average f-measure are given in Figure 6.7 for House A. The average time-slice level f-measure performance for the first

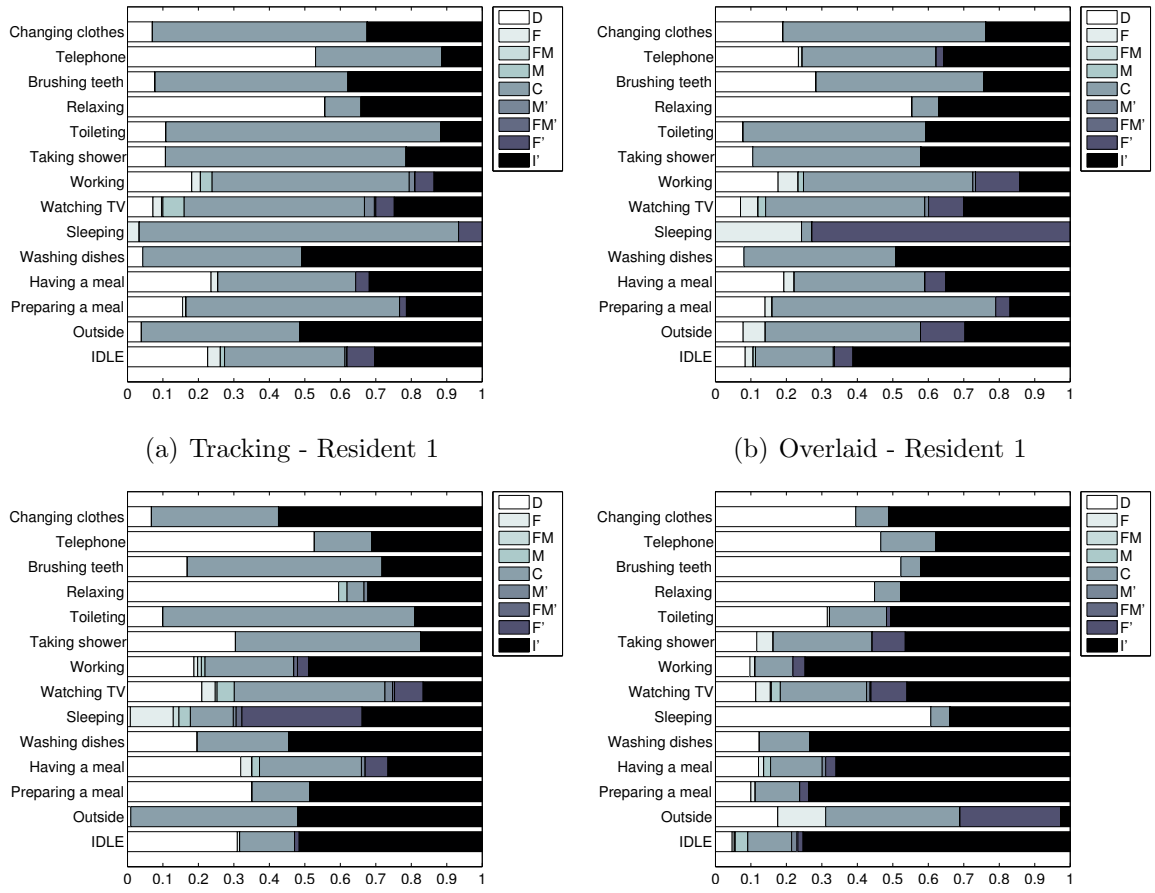


(a) Resident 1 (b) Resident 2

Figure 6.7. Daily average activity recognition performance in terms of f-measure in ARAS House A.

resident is 78.8% when we use an HMM with manually separated observations. The tracking based decomposition yields a 67.9% and the overlaid representation yields 63.0% f-measure performance. For the second resident, the performances are 69.2%, 53.6%, and 25.3% for manually decomposed, tracking based decomposed and overlaid representations respectively.

According to the results, the second resident benefits more than the first resident from the tracking based separation. The underlying reason for this finding is that the time spent inside the house is larger for the first resident. Since the second resident is at work during the working days, the noise caused by this resident is lower. In turn, the noise generated by the first resident is huge for the second resident and it affects the daily recognition performance severely. The same effect is visible in the activity



(a) Tracking - Resident 1

(b) Overlaid - Resident 1

(c) Tracking - Resident 2

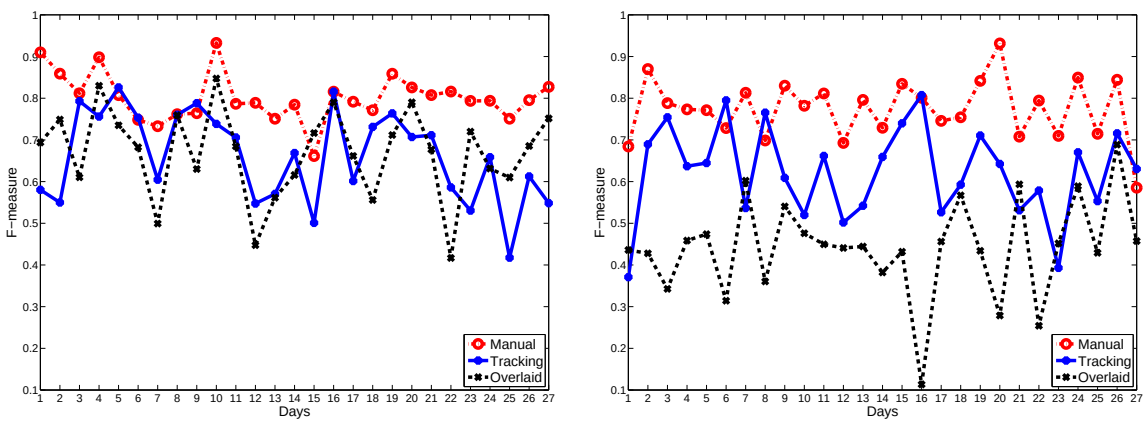
(d) Overlaid - Resident 2

Figure 6.8. Event-based performance evaluation of tracking based observation decomposition and overlaid observations on ARAS House A.

occurrence level performances as well. The EAD diagrams for House A are given in Figure 6.8 for both residents.

The results show the superior performance of the tracking based decomposition approach. The *sleeping* activity performance severely degraded with the overlaid representation for the first resident. When we consider the tracking based decomposition performances only, the most challenging activities are *talking on the phone* and *relaxing*. These activities are challenging mostly because they have no regular patterns in terms of sensor firings. Therefore, the performance of these activities do not differ much with decomposed observations or overlaid observations. For the second resident, tracking based decomposition also suffers from segmentations errors more than the first resident. Especially, the *sleeping* activity recognition performance is poor due to the sensor hardware failures rather than the tracking algorithm's inefficiency.

Figure 6.9 depicts the results of the experiments in terms of daily average f-measure for House B. The average time-slice level f-measure performance for the first resident is 80.1% when we use an HMM with manually separated observations. The tracking based decomposition performance is 66% and the overlaid representation performance is 66.9% in terms of f-measure. For the second resident, the performances are 77.3%, 62.1%, and 44% for manually decomposed, tracking based decomposed and overlaid representations respectively. Similar to the case in House A, the second res-

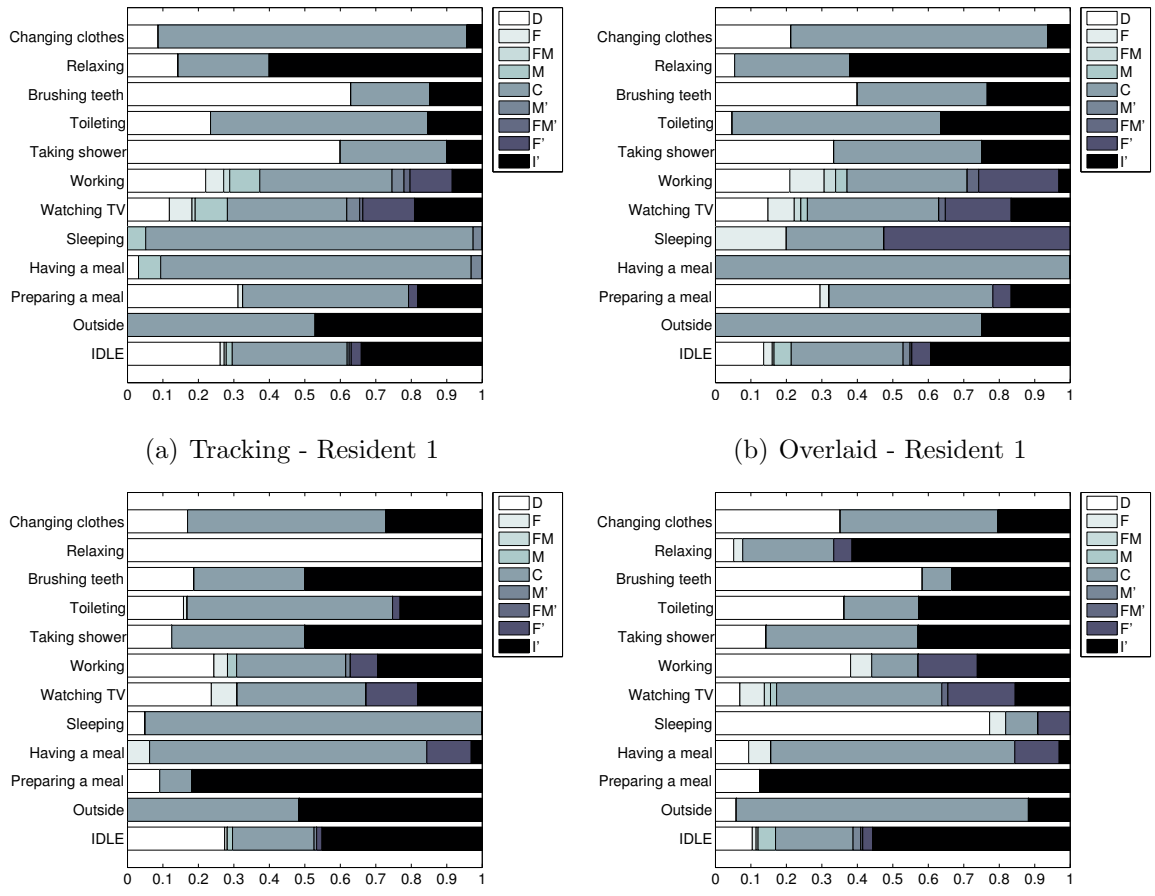


(a) Resident 1 (b) Resident 2

Figure 6.9. Daily average activity recognition performance in terms of f-measure in ARAS House B.

ident benefits more than the first resident from the tracking based separation due to the differences in the durations spent in the house. Also, the second resident has a more sedentary life inside the house unlike the first resident.

The EAD diagrams for both residents in House B are given in Figure 6.10. The overall performance of tracking based decomposition is higher than the overlaid observations, for both residents. For the second resident, however, there are exceptions that the performance with the overlaid observations are higher. The *relaxing* activity, for example, is completely deleted with the tracking based decomposition method whereas with overlaid observations, it is still possible to recognize 20% of the all activity occurrences correctly. Likewise, *being outside* activity is recognized better with the overlaid observation model for both residents.



(a) Tracking - Resident 1

(b) Overlaid - Resident 1

(c) Tracking - Resident 2

(d) Overlaid - Resident 2

Figure 6.10. Event-based performance evaluation of tracking based observation decomposition and overlaid observations on ARAS House B.

6.4.3. Discussion

The results of our experiments indicate that both direct modeling using a FHMM and decomposition of observations using PF tracking together with JPDA yield better performance than their alternative counterparts. Although the amount of performance increase varies between different houses and residents, the overall performance gain is present for all cases making both options viable for multi-resident activity recognition. However, there are several advantages and disadvantages of each approach.

To begin with, in terms of run time, FHMM has a higher complexity than the particle filter based decomposition together with an HMM approach. Therefore, for houses with more than two or three residents, FHMM method becomes intractable while with the tracking based method, the time complexity linearly increases with the number of residents. On the other hand, with a higher number of residents, the tracking problem without assuming any identification mechanism will become extremely challenging because of the additional complexity. Decomposition based methods are more preferable due to their flexibility only when the quality of the decomposition is adequate. Once the decomposition is achieved, any model and method can be employed for activity recognition. With FHMM, since the model is fixed, this flexibility diminishes. Besides, in order to train a more complex model like FHMM, more training data is required. Since the annotation is a costly procedure obtaining a larger training data set is not an easy task.

The flexibility of the decomposition based method comes at the cost of fragility. That is, when an incorrect association is made, it is highly probable that it will propagate through several time-steps. For the tracking based decomposition to work efficiently, a set of assumptions are required. For example, when the sensor on the outside door fires, we cannot determine whether someone has left the house or someone has just entered into the house. It is also possible that none of these happens when the door sensor fires or both of them can occur at the same time. For that reason, an identification mechanism is needed just for determining the number and the identities of the residents in the house. Also, in order to prevent identity switches to propa-

gate, several correction mechanisms are needed. As an example, we can use an active learning approach by asking the correct assignments to the residents themselves in challenging situations.

Neither of the methods we experimented with in this chapter is clearly better than the other. Therefore, considering the highlighted strengths and weakness of each method described, it is possible to choose the most suitable method for different settings. Also, it is possible to come up with a hybrid approach. In the beginning of the system deployment, when there is not enough training data set, it is more suitable to use the tracking based separation. As the training data gets accumulated, the system can switch to FHMM.

6.5. Conclusion

In this chapter, we focused on multiple-resident handling in smart homes for activity recognition purposes. We proposed two different approaches for handling the multiple residents in smart environments without assuming any explicit identification. In the first approach, we used a FHMM for modeling two separate chains, i.e., one for each resident. Secondly, we use nonlinear Bayesian tracking for decomposing the observation space into the number of residents. We performed experiments on real-world multi-resident ARAS data sets. In each experiment, we compared the proposed approach with a counterpart method. We also compared each approach with the manually separated observation performances. The results of our experiments revealed a great potential for both of the methods. The proposed methods consistently outperformed their counterparts for all houses and residents. Since both approaches are viable, we discuss the advantages and disadvantages of each approach in terms of run time complexity, flexibility and generalizability.

Although we obtained highly promising results on two different real-world datasets, there is still room for improvement since the models using manually decomposed observations have higher average performance than the proposed methods. As a future work, we will focus on improving the performance of the tracking based decomposition

methods with more sophisticated tracking and data association mechanisms. For the FHMM model, we will explore approximate methods in order to relax the run time restrictions that arise when there are three or more residents.

7. ACTIVE LEARNING

7.1. Introduction

All of the probabilistic models we use for human activity recognition require labeled training data to learn the model parameters. We showed that these probabilistic models can accurately recognize activities, but two problems limit the large scale applicability of these models: (i) Differences in the layout of houses and the behavior of the large scale inhabitants mean a set of model parameters used for one house cannot be used in another house. (ii) The behavior of inhabitants changes over time, therefore parameters learned at one point in time may not accurately represent the behavior at a later point in time. Although both of these problems can be resolved by recording further annotated data, this solution is far from being practical and cost effective. Instead we propose to develop novel learning methods that allow us to deal with these problems cost effectively. This would allow the installation of activity recognition systems on a large scale and provides a solution for dealing with the consequences of an aging population.

In order to decrease the annotation effort, we can use a machine learning technique called active learning to select only the most informative data points for annotation. By requesting annotation only for the most informative data points, we reduce the amount of training data needed and minimize the annotation effort. In this chapter, we propose a framework for active learning that can be used with any probabilistic model. We assess the performance of our method by conducting experiments on the multiple real world data sets.

The chapter is organized as follows. In Section 7.2, we give a brief literature review on active learning applications to activity recognition. In Section 7.3, we provide the details of the model and active learning methods we used. Section 7.4 gives the details of our experiments with real world data. In Section 7.5, we provide an example application for collecting the annotation labels. Finally, we conclude with Section 7.6.

7.2. Related Work

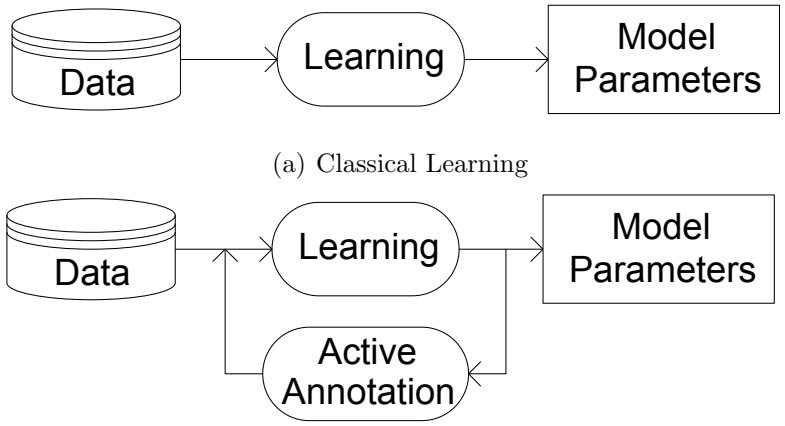
Active learning has been generally used in part of speech tagging problems in natural language processing [106,107]. There are a number of query selection strategies in the literature [108]. The use of active learning in activity recognition systems is studied by a few other researchers. In [109], Liu *et al.* use active learning with a decision tree model to classify the activities collected by a group of wearable sensors. In [110], a similar study is presented using classifiers like decision tree, joint boosting and Naive Bayes. In both studies, uncertainty based active learning methods are employed and active learning has been showed to work well. These earlier studies use classifiers that do not take the sequential nature of the data into account. Since human activities are temporal in nature it is more suitable to use models that consider the temporal nature of human activities.

Truyen *et al.* [111] propose an active learning method for a video-based activity recognition system. They use generative and discriminative temporal probabilistic models for recognizing activities from video sequences. However, video-based activity recognition systems are prone to occlusions and also have privacy constraints. Therefore, they are not widely accepted.

In [112], the authors propose to use active learning for adapting to the changes in the layout of the living place. They use an entropy based measure to select the most informative instances and they evaluate the performance under laboratory conditions making two different controlled changes in the sensor deployment. Reported results indicate 20% decrease in the amount of training data required to retrain the system.

7.3. Active Annotation

In this section, we first provide brief information about existing machine learning techniques that do not use active learning. After that, we describe our proposed active learning framework and state how it differs from the classic learning approach. Finally, we describe three measures that can be used in active learning for selecting the most



(a) Classical Learning

(b) Active Learning

Figure 7.1. Learning frameworks.

informative data points.

In order to use a probabilistic model, a set of model parameters have to be learned. In Figure 7.1a, the classical learning framework is depicted. The model parameters which we denote by θ, can be learned using a supervised method which only uses the data whose labels are obtained through annotation.

In our framework, we use only the labeled data points for obtaining the model parameters and the unlabeled data is disregarded. As depicted in Figure 7.1b, the active learning algorithm iteratively

1. Learns new parameters using supervised learning
2. Selects the most informative data points according to the current model parameters and obtain their labels

More formally, we define $x = \{x_1, x_2, ...x_T\}$ as the set of data points (i.e. data collected from the sensors), $y = \{y_1, y_2, .., y_T\}$ as the set of true labels (i.e. activity performed by the user). The *labeled data set* is $\mathcal{L} = \{x_i, y_i \mid x_i \in x, \quad y_i \in y, \quad 1 \leq i \leq T\}$. The *unlabeled data set* is $\mathcal{U} = \{x_i \mid x_i \notin \mathcal{L}, \quad 1 \leq i \leq N\}$. Typically we have a lot more unlabeled data than labeled data, $N \gg T$. We define the union of these data sets as $\mathcal{D} = \{\mathcal{L} \bigcup \mathcal{U}\}$ and the size of $\mathcal{D}$ is fixed.

At each iteration, we transfer the data points from $\mathcal{U}$ to $\mathcal{L}$ by performing annotation. The size of $\mathcal{L}$, denoted by T, increases while the size of $\mathcal{U}$, denoted by N, decreases. The data points that will be transferred from $\mathcal{U}$ to $\mathcal{L}$ are selected by the active learning method according to some informativeness measure. We use uncertainty for assessing the most informative data points [113]. Probabilistic models need to calculate the probability distribution of the activities at each data point to perform inference. For many probabilistic models, there exist efficient algorithms to calculate these quantities, for example, the forward-backward algorithm is used for HMMs [114]. The forward-backward algorithm gives the probabilities for each activity at each time slice. While performing the inference, the model selects the activity that has the highest probability value for that time slice. We use the forward-backward algorithm to obtain the probabilities of each activity at each time slice according to the current model parameters θ, which we denote with P_θ. After that, to select the most informative data point, x^*, we use three different methods.

1. *Least Confident Method* considers only the most probable class label and selects the instances having the lowest probability for the most likely label.

$$x^* = \arg\max_x (1 - P_\theta(\hat{y} \mid x)) \tag{7.1}$$

   where $\hat{y} = \arg\max_y P_\theta(y \mid x)$ is the class label with the highest probability according to the current model parameters θ.
2. *Margin Sampling* selects the instances that the difference between the most and the second most probable labels is minimum.

$$x^* = \arg\min_x (P_\theta(\hat{y_1} \mid x) - P_\theta(\hat{y_2} \mid x)) \tag{7.2}$$

   where $\hat{y_1}$ and $\hat{y_2}$ are the two most probable classes.
3. *Entropy based* method selects the instances that have the highest entropy values

among all probable classifications.

$$x^* = \arg\max_x - \sum_i (P_\theta(\hat{y}_i \mid x) log P_\theta(\hat{y}_i \mid x)) \tag{7.3}$$

7.4. Experiments

We search for the effect of active learning for reducing the annotation effort in activity recognition. That is, we want to recognize the activities as accurate as possible while using the minimum amount of labeled data. Also, we do not want to disturb the user for a label that he possibly does not remember. Asking about the label of the activity that had been performed a month ago is not realistic. In this study, we propose a daily querying approach and evaluate its performance on real world data sets. Our experiments aim to answer three questions: (i) Does active learning reduce the annotation effort?, (ii) What is the best uncertainty measure for selecting the most informative data points?, and (iii) What is the most suitable setup for the number of data points and for the number of iterations?

7.4.1. Experimental Setup

We use ARAS datasets with a manually decomposed observation space as described in Section 3.4. The data are discretized in $\Delta t = 60 sec$ using raw feature representation. We use HMM for activity recognition model and leave-one-day-out cross validation in all experiments. We use one full day of data for testing and the remaining days for training. We use training days in a sequential manner, that is, after we process a day's data, we move to the following day and do not use the data of the previous day for obtaining labels. As stated previously, we iteratively learn new model parameters and select the most informative points to be annotated. In the learning phase, we use all the data points whose labels we already obtained. However, we do not select data points for annotation except from the current day. In other words, in each iteration, we learn model parameters with all the data that we obtained thus far. After that, according to the newly learned parameters, we select the data points to be

annotated from only the current day. We cycle over days for testing and use every day once for testing. We report the average of the performance measure.

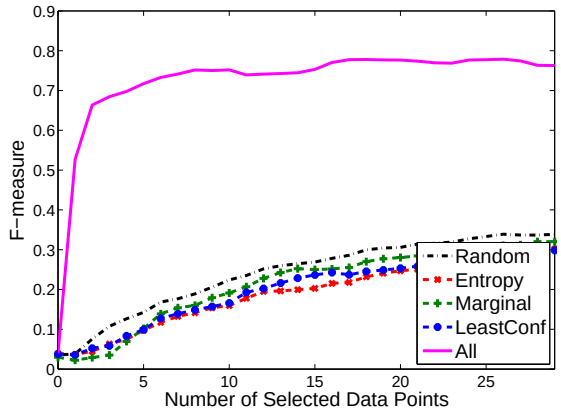
With respect to the research questions we aim to answer, (i) we use a random selection approach together with uncertainty sampling to show the effect of the active selection, (ii) we experiment with three different uncertainty measures to find the most suitable measure for selecting the most informative data points, and (iii) we experiment with four different setups of active annotations namely, we select,

1. a single data point from each day in a single iteration,
2. ten data points from each day in a single iteration, resulting in ten data points from each day,
3. a single data point but we make ten iterations per day, resulting in ten data points from each day, and
4. ten data points in ten iterations per day, resulting in 100 data points from each day

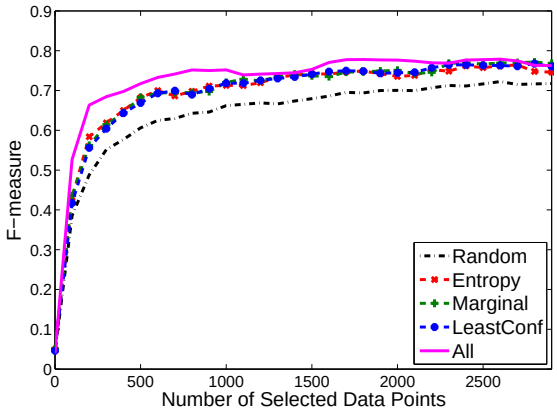
7.4.2. Results

We present the results of our experiments for each house and for each resident separately. For each case, we also include the fully annotated performance into the graphs in order to make realistic evaluations. The fully annotated performance graphs, drawn as solid magenta lines, indicate the scenario in which we select the whole 1440 data-points from each day for annotation as opposed to the actively or randomly selected portions. The results for House A for Resident 1 is given in Figure 7.2. The results shows that with a single data point from each day we severely undershoot the maximum achievable performance. With ten data points in ten iterations case, on the other hand, we observe a highly comparable performance when we use active learning. When we randomly select the data points instead of wisely selecting them according to one of our uncertainty measure, we cannot achieve the optimum performance. When we consider the ten data points per day configurations, we observe similar performances with one iteration and ten iterations cases. For these configurations, entropy based

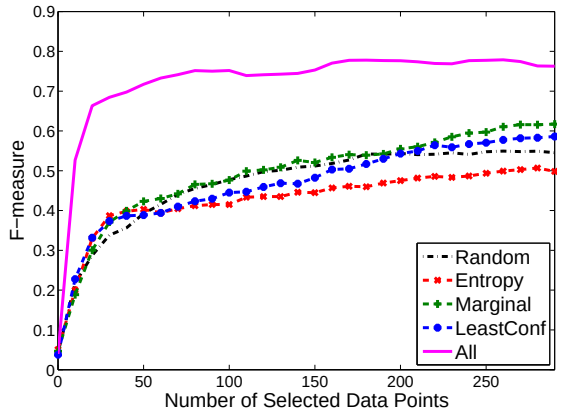
selection underperforms when compared to other selection methods.



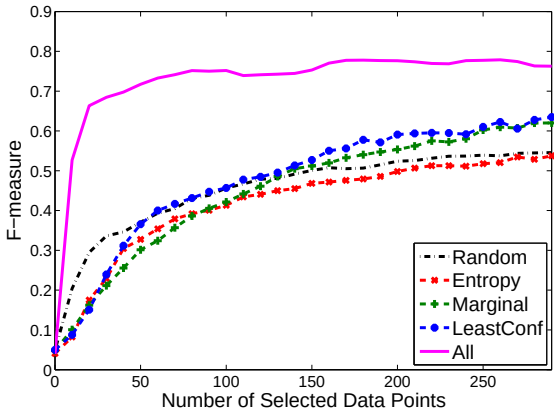
(a) 1 Data Point/Iteration - 1 Iteration/Day



(b) 10 Data Point/Iteration - 10 Iteration/Day



(c) 1 Data Point/Iteration - 10 Iteration/Day

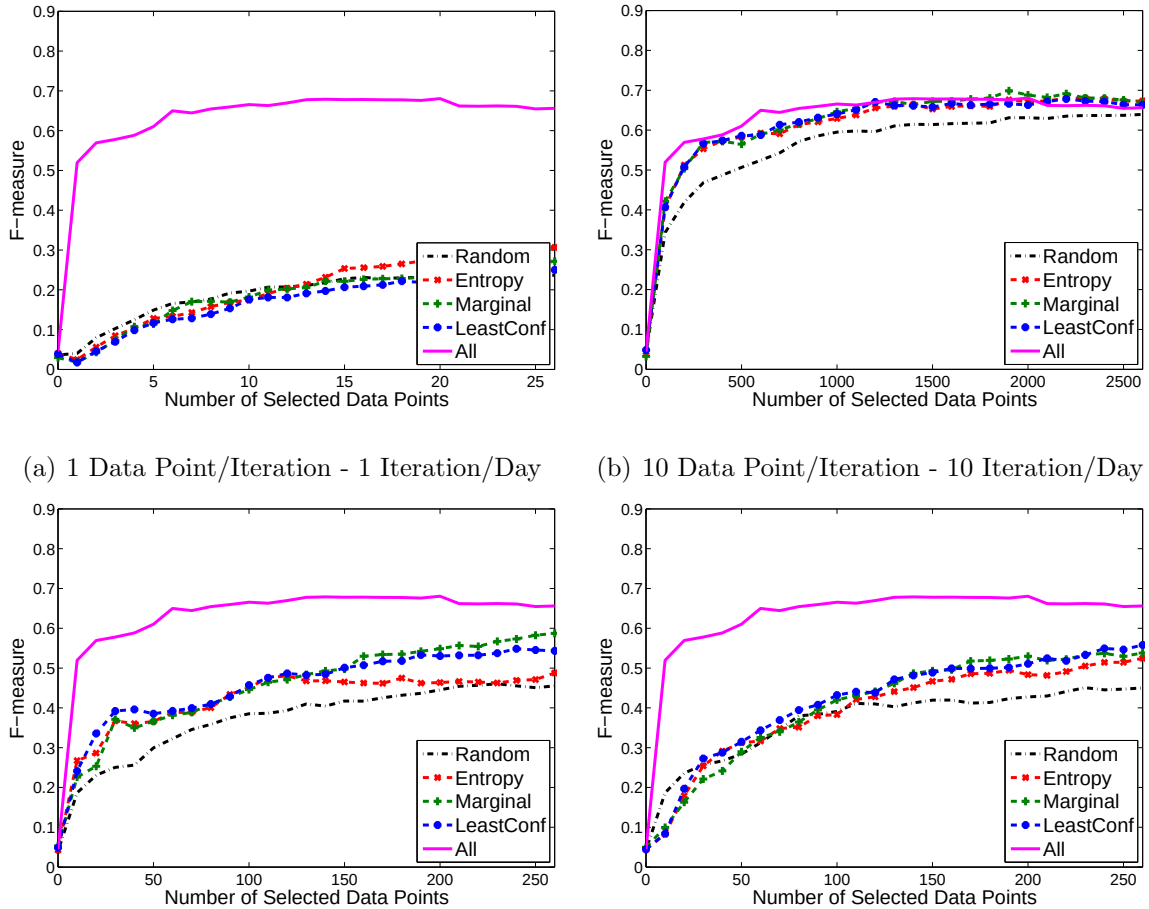


(d) 10 Data Point/Iteration - 1 Iteration/Day

Figure 7.2. Active learning experiment results for House A - Resident 1.

In Figure 7.3, we show the results for House A, Resident 2. Similarly, the single point per day case yields a very low performance whereas the 100 points case reveals a significantly higher performance. Also, it is interesting to observe a higher performance than the fully annotated case. This can be attributed to the change in the resident's annotation behavior. The downward trend in the performance towards the end supports this argument. When we have the full annotation, our observation model changes according to the annotator's overall average behavior immediately. When a difference in the way a specific activity is performed occurs, or a difference in the annotation behavior is observed, it is directly reflected on the performance. With the active learning, however, if we do not select those data points causing this discrepancy between the training and the test sets, we do not have any effect on the performance

on the test sequence. Although in this case we obtain a higher performance with active learning, it is important to note that this effect can also cause a degradation in the performance of active learning for other settings.



(a) 1 Data Point/Iteration - 1 Iteration/Day

(b) 10 Data Point/Iteration - 10 Iteration/Day

(c) 1 Data Point/Iteration - 10 Iteration/Day

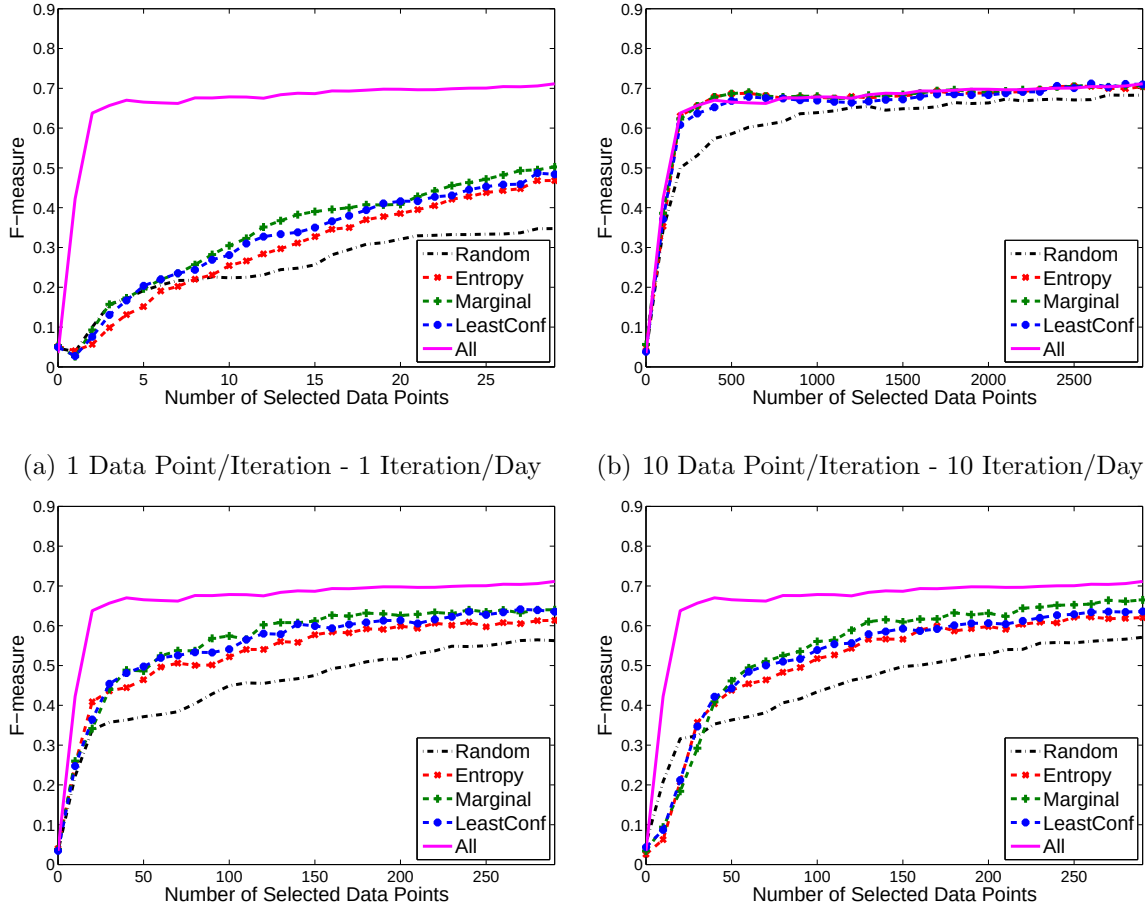
(d) 10 Data Point/Iteration - 1 Iteration/Day

Figure 7.3. Active learning experiment results for House A - Resident 2.

In terms of ten point per day selection configurations, entropy based selection performs worse than the other selection mechanisms. This effect is more prominent in ten iterations case. Also, marginal method performs slightly better than the least confident method.

The results for House B for the first resident is depicted in Figure 7.4. Most of the previous findings persist for this configuration as well but with a higher general performance increase with respect to the maximum achievable performance. With a 100 point selection per day, the performance converges to the maximum within five

days. Also, the benefit of using uncertainty based measures over the random selection is more prominent in this house.



(a) 1 Data Point/Iteration - 1 Iteration/Day

(b) 10 Data Point/Iteration - 10 Iteration/Day

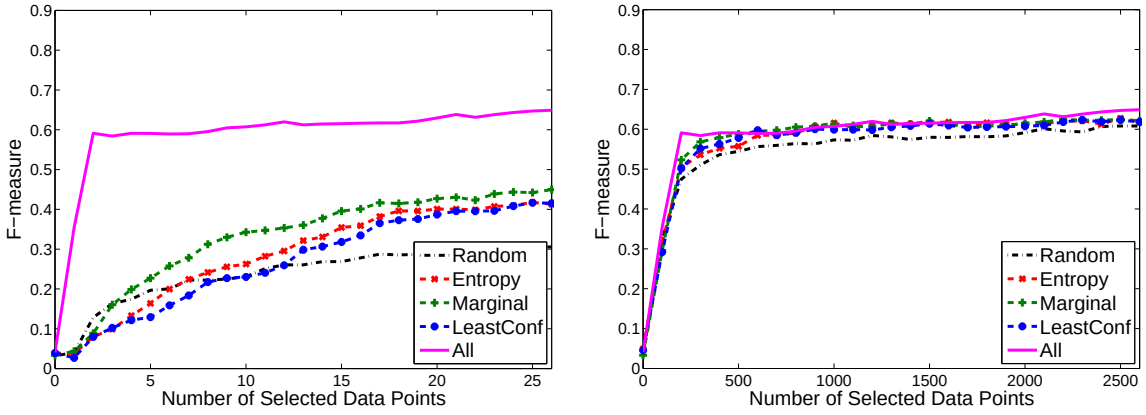
(c) 1 Data Point/Iteration - 10 Iteration/Day

(d) 10 Data Point/Iteration - 1 Iteration/Day

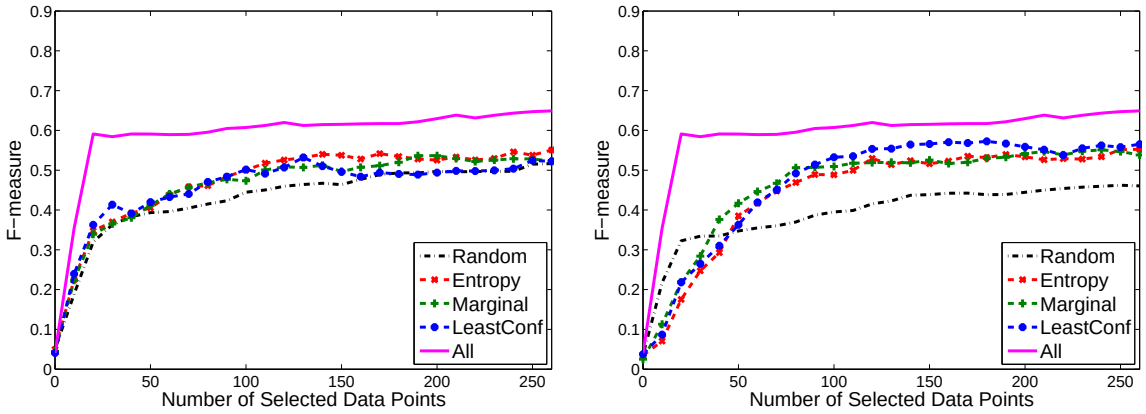
Figure 7.4. Active learning experiment results for House B - Resident 1.

Finally, the second resident for House B results are given in Figure 7.5. Similar to the other resident's case for this house, the benefit of using active learning even with a low number of data points is prominent. With a single data point per day, the performance of marginal selection method is better than the other methods. For the other cases, there are not significant differences between the selection methods.

In the experiments, we use one minute discretization, therefore, in each day there are 1440 data points. When we consider the best performing setup with selecting 100 points in each day, we only use the 7% of all the available data points and obtain almost fully annotated recognition performance only after a couple of days. When we compare



(a) 1 Data Point/Iteration - 1 Iteration/Day (b) 10 Data Point/Iteration - 10 Iteration/Day



(c) 1 Data Point/Iteration - 10 Iteration/Day (d) 10 Data Point/Iteration - 1 Iteration/Day

Figure 7.5. Active learning experiment results for House B - Resident 2.

the annotation effort of real time setup (i.e. at the time of data collection and whenever an activity starts or finishes) versus active annotation (i.e. after the day is completed in an offline fashion), again the offline active annotation is more preferable. During the data collection phase, the first and the second residents in House A made an average of 43 and 30 annotations per day, respectively. Similarly, for House B, the average number of annotations per day was 21 and 14 for resident 1 and 2, respectively. In summary, even with the most demanding active annotation setup, we query the residents 30% - 75% less and ask only a few number of data points in order to achieve the same recognition performance.

7.4.3. Discussion

We show that active learning works well for an activity recognition application with experiments on real world datasets. With the active learning framework, the activity recognition system selects the most informative points. Then, the system is trained iteratively, using only the most informative points' labels. In our experiments, we selected the points that needed to be annotated on a daily basis. At the end of each day, the system asks the user what he/she has been doing during the time slices that are chosen to be the most informative. In our scenario, it is possible that the user is disturbed only once a day, possibly before going to bed, by the system and asked about some activities he/she performed during that day. It is also possible that each iteration takes place at different times. This is important especially for the higher number of selections such as ten points in ten iterations cases. It could be difficult to obtain all 100 point in a single session.

The active learning framework we propose allows different number of data points to be selected from each day. Having more data points is always better but the number can vary from one to up to all data points. The model parameters are recalculated after each obtained label since each labeled point is of significant importance to obtain accurate model parameters. Since we use a supervised approach, recalculating the parameters is very fast and the user does not have to wait to be asked about the following label. We iteratively select points and update the model parameters, therefore, bias on selection do not propagate. Also, since we always obtain the true labels for the selected points, bias on learning the model parameters is very unlikely to occur.

7.4.3.1. Random vs. Uncertainty Sampling. In nearly all of the cases, random selection performs worse than active learning methods. The exceptions occur especially with an extremely low number of data points. When the number of data points are too low, the model is not accurate enough to correctly determine the importance of the data points. In that case, it is possible to come up with a higher performance with a random selection. Even with a random selection, the labels we obtain is the ground

truth labels so that they are useful in learning as well. However, in all the experiments, there is a clear distinction with the uncertainty measure based selection and random selection stating that these measures work better than random.

7.4.3.2. Comparison among Uncertainty Measures. As long as the different measures are concerned, we do not observe significant differences. Nevertheless, we can state that marginal selection method has a slightly higher performance than the others whereas entropy method has a slightly lower performance than the others. But the differences are quite subtle.

7.4.3.3. Single iteration vs. Multiple iterations. In the results, we provide two different configurations for ten points per day selection, i.e. we collect ten points in a single iteration as opposed to a single point in ten iterations. Similarly, we also experimented with selecting 100 points per day in a single iteration and selecting ten points in ten iterations. Since the results for the former configuration are not so different from its ten iteration counterpart, we do not provide the performance graphs of these experiments separately. The general performance trends are the same for both configurations in each case. On the other hand, when we make more iterations, we have a steeper learning curve especially for the first few iterations. As the number of labeled points increase, the effect of iterations disappear. A steeper learning curve is expected since before asking for new labeled points, we have the opportunity to update the model and ask about more informative labels with a more mature model. When we have a better model, the marginal benefit of the uncertainty measures increases. This, in turn, leads to a steeper increase in the performance in the beginning of active learning.

The number of iterations becomes an important design consideration if the learning algorithm has a high time complexity. If the running time of the learning algorithm is long enough to be noticed by the annotator then there will be pauses between the consecutive queries. If the pauses are too long then the annotator will become annoyed. In these cases, the iteration counts should be kept at minimum for usability purposes. When the learning algorithm is fast enough, using more iterations is more

beneficial for efficient learning of the activities actively. Also, a hybrid approach could be employed since we already showed the effect of the higher number of iterations are more significant in the beginning.

7.5. Annotation Tool

One important concern with offline active annotation is the memory limitations of human annotators. Since the active query selection is performed after the whole day is over in our proposed scenario, we developed a prototype application to mitigate the negative effects of incorrect retrieval. The proposed application can be used for both querying the annotator and also visualizing the sensor data for that specific moment.

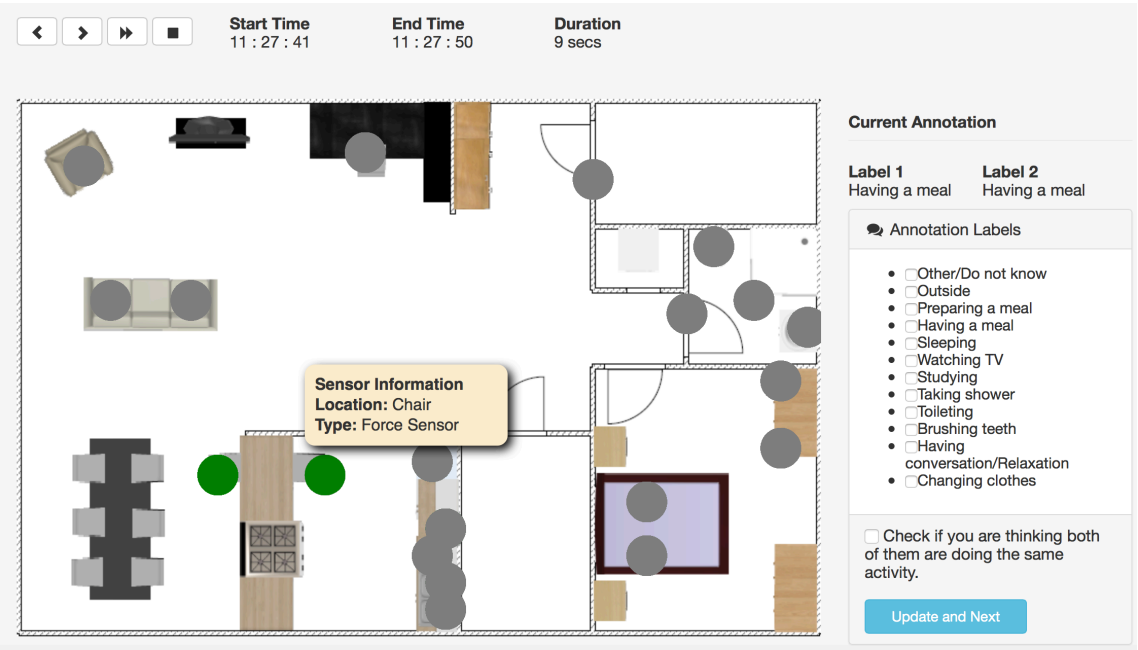


Figure 7.6. A screen shot from the web-based annotation tool.

In Figure 7.6, a sample screen is given from the developed sensor data annotation and visualization tool. This web-based simple yet efficient tool allows us to collect the necessary ground truth labels while also serving as a memory aid tool [115]. The application shows active and passive sensors together with their locations and types. While active sensors are shown as green circles, passive sensors are shown as grey circles. When the annotator moves the mouse over the circles, she/he can see location and type

of sensors. This property helps annotators make better interpretation. After the users see this visualization of active and passive sensors, they are expected to annotate the activities choosing labels from set of activity labels on the right hand side. In order to further facilitate the retrieval process, the start and end times of the specified sensor state configuration together with the sensor firing duration information is provided at the top portion of the screen. Besides, the navigation buttons help the annotator to move back and forth between the time steps. This mechanism helps in making the temporal connections between consecutive time steps and in case of sensor failures or noisy firings, the annotator can make better interpretations about the ground truth activity labels.

One important benefit of having a web-based tool for annotation is that we can utilize other people for the annotation task. This feature may become useful especially in cases where the residents are incapable of annotating their own activities due to dementia or other diseases. In that case, authorized relatives or healthcare personnel can perform the annotation tasks. Although the accuracy is expected to be lower when compared to self annotation, our preliminary experiments with several unfamiliar annotators indicate a relatively high accuracy values for most activities of daily living such as sleeping, having a meal, toileting and watching TV. Activities that are more open to different interpretations such as relaxing or working are more challenging for unfamiliar annotators. Nevertheless, the flexibility of the overall learning phenomena makes it a proper candidate for large scale deployments of activity recognition systems.

7.6. Conclusion

In this chapter, we addressed the scalability problems of automated human activity recognition systems since they require labeled data sets for adapting themselves to different users and environments. Collecting the data, annotation and retraining the systems from scratch for every person or every house is too costly. Therefore, redeploying these systems in different settings should be accomplished in a cost effective and user friendly way. For this purpose, we propose active learning methods which reduce the annotation effort by selecting only the most informative data points to be

annotated. In our framework, we also consider the user friendliness. We showed that by disturbing the user only a few times each day for obtaining the minimum amount of labels, we can still learn accurate model parameters.

We used three different measures of uncertainty for selecting the most informative data points and evaluated their performance by using real world data sets. We used HMM as the probabilistic model for all experiments. Experiments showed that all three proposed method works well for the activity recognition system. We showed through experiments on real world data sets that, by using the active learning instead of random selection, the annotation effort is reduced by a factor of two to four, depending on the house and resident setting in ARAS datasets.

Achieving high performance in activity recognition systems using probabilistic models depends on model parameters that are learned using the labeled data. With active learning, we aim to reach the most accurate model parameters iteratively using the parameters obtained from previous iterations for selecting the most informative data points. In the first iterations, the parameters are based on few number of data points, therefore, not accurately estimated. This leads to a poor estimate of the informativeness of data points at the first iterations. We can see from the results that even with a small amount of training data obtained after a few iterations, the selection gets better quickly. Therefore, instead of randomly initializing the parameters in the first iteration, we can use a method called transfer learning which allows the use of model parameters that have been learned previously to be used in another setting [116]. As a future study, using transfer learning together with active learning methods could be explored to lead better estimates of the parameters even at the first iterations.

8. CONCLUSIONS

In this thesis, we focused on human activity recognition problems in smart environments using interaction based sensing. Different from the current state-of-the-art which mostly concentrates on the single resident case, we addressed the multiple person human activity recognition problem.

We began by collecting two real-world benchmarking datasets from two different real houses. We deployed 20 interaction-based binary sensors in each house with two residents. We gathered one full month of sensor data together with the ground truth activity labels for both residents from each house. The ARAS datasets are made public so that the community can develop and benchmark novel methods' performances under realistic conditions. For data collection purposes, we proposed a multimodal WSN-based AAL system compatible for homes with multiple residents with the aim of recognizing the daily activities and routines of the users to detect the drifts and differences in their behavior, especially for monitoring their health and wellbeing status. In particular, we provided several guidelines for the design and deployment of an effective AAL system.

In order to automatically recognize the activities of daily living, we used several machine learning techniques in order to accurately and efficiently model and recognize. While doing so, we have not undermined the domain specific needs of the human behavior monitoring for health assessment purposes. Since human activities contain a complex hierarchical structure, we explored the ways for accurately and automatically finding a suitable structure for modeling them. We proposed a model that uses a semi-supervised learning approach to automatically cluster the inherent structure of activities. We used three different model selection mechanisms, namely, CVL, AIC, and BIC, for finding the number of states used to represent the actions that make up each activity. Our experimental evaluations on ARAS datasets showed that the use of a hierarchical model consistently outperforms its non-hierarchical counterpart in terms of recognition performance, given that an adequate number of states is used for

modeling the actions in the hierarchy. As opposed to the previous work, we employed a model selection mechanism to determine the optimal number of sub-states for each activity. We showed through experimental evaluation that the model selection using CVL methodology, consistently outperformed the penalized likelihood methods. This finding confirms the previous studies stating that AIC and BIC measures have a tendency to over-penalize the model complexity. Although, the CVL method has a much higher computational complexity, the high increase in the performance redeems.

Our results suggest a great potential in further research for improving the ways of finding the optimal model that can grasp the complexity of human activities. As a future work, we propose a bottom-up approach for determining the complexity for the upper-layer activities. Also, rather than finding the optimum model size, we can assume an infinite number of states in the hierarchy by using an iHMM [84] or a hierarchical iHMM model [85].

Since we use machine learning techniques, in the experiments, we mostly use conventional metrics such as accuracy and f-measure. These metrics are widely used for evaluation purposes because of their compactness. Yet, this compactness causes a loss in the human behavior perspective when applied to assessment of well-being in AAL systems. There exists a trade-off between compactness and informativeness. Since the human behavior understanding for healthcare monitoring requires delicacy, we propose trading some of the compactness with informativeness to obtain deeper insights. We proposed a method for evaluation of different approaches for the purposes of human behavior understanding through a well-being assessment perspective. We demonstrated the shortcomings of the use of general purpose metrics with experiments on real world data.

Human behavior analysis from a medical perspective requires analysis of daily activities in terms of timing, duration and frequency. Given the high variations in these attributes for different activities, the general purpose metrics fail to accurately reflect the actual performance. Our proposed evaluation method is more generally applicable to the real world applications that require human behavior understanding. In the

proposed method, we first group the activities of daily living in terms of their duration and frequency sensitivities. Then, we map the categories to appropriate evaluation strategy using either time-slice level or event level criteria. In this way, we provide sounder evaluation criteria rather than a one-size-fits-all approach, i.e. using the same single metric for all types of activities. Using the newly proposed method, we compared the performance of two machine learning models, HMM and TWNN, on five different real world datasets from a behavior monitoring perspective. The results with real world human behavior data revealed that the use of standard metrics can be misleading in demonstrating the performance from a behavior understanding perspective.

In this thesis, we also focus on making smart houses smart enough to provide long term health monitoring for not only people who live alone but also with a spouse or a flat mate. In that respect, we need to recognize behavior individually in multi-resident environments without assuming any person identification which generally requires the use of wearable technology that can be obtrusive. We proposed two different approaches for handling the multiple residents in smart environments without assuming any explicit identification. In the first approach, we used FHMM for modeling two separate chains, i.e., one for each resident. Secondly, we use nonlinear Bayesian tracking for decomposing the observation space into the number of residents. We performed experiments on real-world multi-resident ARAS data sets. In each experiment, we compared the proposed approach with a counterpart method. We also compared each approach with the manually separated observation performances. The results of our experiments revealed a great potential for both of the methods. The proposed methods consistently outperformed their counterparts for all houses and residents. Since both of the proposed approaches are viable, we discussed the advantages and disadvantages of each approach in terms of run time complexity, flexibility and generalizability as well.

Our experiments revealed highly promising results on two different real-world datasets, yet, when compared to the manual separation in which we assume that we know the identity of the person who fired a specific sensor at any given point in time, there is still room for improvement. As a future work, we propose focusing on improving the performance of tracking based decomposition methods with more sophisticated

tracking and data association mechanisms. For the FHMM model, it is important to explore approximate methods in order to relax the run time restrictions that arise when there are three or more residents. Also, further work on this subject can be using a hybrid approach so that, in the beginning of the operation, when there is limited amount of training data set, it is more convenient to use tracking based separation. As the training data gets mature, the system can switch to FHMM.

Finally, we addressed the scalability problems of automated human activity recognition systems since they require labeled data sets for adapting themselves to different users and environments. Collecting the data, annotation and retraining the systems from scratch for every person or every house is too costly. Therefore, redeploying these systems in different settings should be accomplished in a cost effective and user friendly way. For this purpose, we propose active learning methods which reduce the annotation effort by selecting only the most informative data points to be annotated. In our framework, we also consider the user friendliness. We showed that by disturbing the user only a few times each day for obtaining the minimum amount of labels, we can still learn accurate model parameters.

The effectiveness of the proposed methods depend highly on both the environment and the people, therefore the models and the parameters are subject to change across different environments and different people. In order to deploy these systems on a large scale, we need to relearn the parameters for each setting. Moreover, even for the same setting, they are subject to change over the course of the time. This change can stem from a variety of reasons such as the changes in the behavior of the people, changes in the environment or changes in the sensor behaviors. Learning the parameters for every different setting from scratch is not feasible since it requires large amount of annotated data which is hard to obtain. Instead, we can use active learning to select the most informative data points for annotation. By requesting annotation only for the most informative data points, we reduce the amount of training data needed and minimize the annotation effort.

We used three different measures of uncertainty for selecting the most informative

data points and evaluated their performance by using real world data sets. We showed through experiments on real-world ARAS data sets that, by using the active learning instead of random selection, the annotation effort is reduced by a factor of two to four, depending on the house and resident setting. With active learning, we aim to reach the most accurate model parameters iteratively using the parameters obtained from previous iterations for selecting the most informative data points. In the first iterations, the parameters are based on few number of data points, therefore, not accurately estimated. This leads to a poor estimate of the informativeness of data points at the first iterations. Our results indicate that even with a small amount of training data, the selection gets better quickly in around five iterations. Therefore, instead of randomly initializing the parameters in the first iteration, we can use a method called transfer learning which allows the use of model parameters that have been learned previously to be used in another setting [116]. As a future study, using transfer learning together with active learning methods could be explored to lead better estimates of the parameters even at the first iterations.

In conclusion, for many ambient intelligence applications such as smart homes, health monitoring applications, we need to recognize human behavior in an automated manner. In order to make such systems sustainable, we need novel solutions to the present challenges. In this thesis, we addressed several of these challenges in novel ways. In summary, we collected two publicly available benchmark datasets for the community to continue this work. We proposed a methodology for incorporating a hierarchy into the model that is tailored for various activities individually. We improved the ways of evaluating different approaches and models considering the domain specific needs. We proposed two different approaches handling multi-resident environments in an unobtrusive manner. We proposed active and semi-supervised learning techniques in order to reduce the annotation effort in large scale deployments.

APPENDIX A: EXACT FORWARD-BACKWARD ALGORITHM FOR FHMM

Let $\vec{x}_{1:T}$ denote the observation sequence, y_t^i denote the state of the i^{th} chain at time step t , and ψ denote the model parameters. The forward variable α_t is defined as

$$\begin{aligned}
\alpha_t &= p(y_t^1, y_t^2, ..., y_t^E, \vec{x}_{1:t} \mid \psi) \\
\alpha_t^0 &= p(y_t^1, y_t^2, ..., y_t^E, \vec{x}_{1:t-1} \mid \psi) \\
\alpha_t^1 &= p(y_{t-1}^1, y_t^2, ..., y_t^E, \vec{x}_{1:t-1} \mid \psi) \\
&\dots \\
&\dots \\
\alpha_t^E &= p(y_{t-1}^1, ..., y_{t-1}^E, \vec{x}_{1:t-1} \mid \psi) = \alpha_{t-1}
\end{aligned}$$

We obtain the following forward recursions:

$$\alpha_t = p(\vec{x}_t \mid y_t^1, ..., y_t^E, \psi)\alpha_t^0 \tag{A.1}$$

$$\alpha_t^{e-1} = \sum_{y_{t-1}^e} p(y_t^e \mid y_{t-1}^e)\alpha_t^e \tag{A.2}$$

The likelihood of the observation sequence is then the sum of Q^E elements in α_T.

$$p(\vec{x}_{1:T} \mid \psi) = \sum_{i=1}^{Q^E} \alpha_T(i)$$

Similarly, the backward variable β_t is defined as

$$\begin{aligned}
\beta_t &= p(\vec{x}_{t+1:T} \mid y_t^1, ..., y_t^E, \psi) \\
\beta_{t-1}^E &= p(\vec{x}_{t:T} \mid y_t^1, ..., y_t^E, \psi) \\
&\cdots \\
&\cdots \\
\beta_{t-1}^1 &= p(\vec{x}_{t:T} \mid y_t^1, y_{t-1}^2, ..., y_{t-1}^E, \psi) \\
\beta_{t-1}^0 &= p(\vec{x}_{t:T} \mid y_{t-1}^1, y_{t-1}^2, ..., y_{t-1}^E, \psi) = \beta_{t-1}
\end{aligned}$$

The backward recursions are

$$\beta_{t-1}^E = p(\vec{x}_t \mid y_t^1, ..., y_t^E, \psi)\beta_t \tag{A.3}$$

$$\beta_{t-1}^{e-1} = \sum_{y_t^e} p(y_t^e \mid y_{t-1}^e)\beta_{t-1}^e \tag{A.4}$$

The posterior state distribution at time t is given by γ_t:

$$\gamma_t = p(y_t \mid \vec{x}_{1:T}, \psi) = \frac{\alpha_t \beta_t}{\sum_{y_t} \alpha_t \beta_t} \tag{A.5}$$

The probabilities are defined over collections of state variables corresponding to the cliques in the equivalent junction tree. Information is passed forwards and backwards by summing over the sets separating each neighboring clique in the tree. This results in forward-backward type recursions of order $O(TEQ^{E+1})$.

The expectations are calculated as follows:

$$E\langle y_t^e \mid \psi, \vec{x}_{1:T} \rangle = \sum_{y_t^i (i \neq e)} \gamma_t \tag{A.6}$$

$$E\langle y_t^e y_t^f \mid \psi, \vec{x}_{1:T} \rangle = \sum_{y_t^i (i \neq e \wedge i \neq f)} \gamma_t \tag{A.7}$$

$$E\langle y_{t-1}^e y_t^{e'} \mid \psi, \vec{x}_{1:T} \rangle = \xi_t = \frac{\sum_{y_{t-1}^i, y_t^j (i \neq e \wedge j \neq e)} \alpha_{t-1} p(y_t \mid y_{t-1}) p(\vec{x}_t \mid y_t) \beta_t}{\sum_{y_{t-1}, y_t} \alpha_{t-1} p(y_t \mid y_{t-1}) p(\vec{x}_t \mid y_t) \beta_t} \tag{A.8}$$

APPENDIX B: NONLINEAR BAYESIAN TRACKING

In a discrete-time state-space model, the state sequence x_k of a target given by

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{B.1}$$

where f_k is a function of the previous state x_{k-1} and v_{k-1} which is independent and identically distributed (iid) process noise. Since the state vector is not observable directly, the purpose of tracking is to recursively estimate x_k from measurements z_k given by

$$z_k = h_k(x_k, n_k) \tag{B.2}$$

where h_k is a function of the current state x_k and n_k, which is iid measurement noise. In a Bayesian setting, the goal of tracking is to obtain $p(x_k|z_{1:k})$ which is the posterior pdf of the state at time k given a measurement sequence $z_{1:k}$ up to time k. We assume that the initial state distribution is known without any measurements. Therefore $p(x_0|z_0) \equiv p(x_0)$ is a prior. Then, the posterior can be obtained recursively in two stages. In the prediction stage, we compute the predictive pdf of the state at time k using

$$p(x_k|z_{1:k-1}) = \int p(x|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{B.3}$$

where the transition density, $p(x|x_{k-1})$ is given by the system model provided in Equation B.1 and $p(x_{k-1}|z_{1:k-1})$ is the posterior pdf of the previous time step $k-1$ and obtained via recursion.

In the update stage, Bayes' rule is applied to the prior prediction given by Equa-

tion B.3 using the measurement available at time k

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{B.4}$$

where the likelihood $p(z_k|x_k)$ defined by the measurement model given in Equation B.2 and $p(z_k|z_{1:k-1})$ is the normalizing constant which is also called the evidence.

The prediction and update equations given by Eqn B.3 and Eqn B.4, respectively can be calculated analytically only when some restricting assumptions are made. For example, when we assume that the posterior density is Gaussian at every step we can use Kalman filter, or if the state space is discrete with a finite number of states, we can use grid-based methods in order to obtain optimal solution analytically.

In general, the assumptions made for optimal solutions are too restrictive and cannot be applied in many contexts. Therefore, several approximate methods have been proposed such as Monte Carlo sampling. SIS is the most basic Monte Carlo (MC) method used for this purpose.

B.1. Sequential Importance Sampling (SIS)

The SIS algorithm, or the particle filter, is a technique for implementing a recursive Bayesian filter using MC simulations.

Importance sampling is an approximation method that is generally used when it is difficult to sample directly from a target density $p(x)$. Importance sampling is applied by drawing samples from an importance density $q(x)$ which is much easier to sample from and weighting each sample x^i by a weight $w^i \propto \pi(x^i)/q(x^i)$ where $\pi(x) \propto p(x)$ can be evaluated. Then, the target density can be approximated as

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i) \tag{B.5}$$

where $\delta(.)$ is the Dirac delta measure.

The main idea in SIS is to approximate the full posterior distribution $p(x_{0:k-1}|z_{1:k-1})$ at time $k-1$ with a weighted set of samples called the particles, $P = \{x^i_{0:k-1}, w^i_{k-1} : i = 1, .., N\}$, and recursively update these particles and their weights to obtain an approximation to the posterior distribution $p(x_{0:k}|z_{1:k})$ at time k. The sequence of all states up to time k is denoted as $x_{0:k} = \{x_s : s = 0, .., k\}$. The weights are normalized such that $\sum_i w^i_k = 1$.

When we apply importance sampling to full posterior distribution at time $k-1$, the density can be approximated by

$$p(x_{0:k-1}|z_{1:k-1}) \approx \sum_{i=1}^{N} w^i_{k-1} \delta(x_{0:k-1} - x^i_{0:k-1}) \tag{B.6}$$

In the next step, we update the particles $x^i_{0:k-1}$ and their weights w^i_{k-1} so that they approximate the posterior distribution $p(x_{0:k}|z_{1:k})$ at time k. If the importance density is chosen to be factorized as

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k}) q(x_{0:k-1}|z_{1:k-1}) \tag{B.7}$$

then we can obtain samples $x^i_{0:k}$ at time k by simply augmenting each existing particle $x^i_{0:k-1}$ at time $k-1$ with a new state sampled from $q(x_k|x_{0:k-1}, z_{1:k})$ at time k. For updating the weights, we consider the following recursion for the posterior

$$\begin{aligned} p(x_{0:k}|z_{1:k}) &\propto p(z_k|x_{0:k}, z_{1:k-1}) p(x_{0:k}|z_{1:k-1}) \\ &= p(z_k|x_k) p(x_k|x_{0:k-1}, z_{1:k-1}) p(x_{0:k-1}|z_{1:k-1}) \\ &= p(z_k|x_k) p(x_k|x_{k-1}) p(x_{0:k-1}|z_{1:k-1}) \end{aligned} \tag{B.8}$$

If the particles are drawn from an importance density $q(x_{0:k}|z_{1:k})$, then the weights

w_k^i should follow

$$w_k^i \propto \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})} \tag{B.9}$$

by combining Equations B.7, B.8 and B.9, we get

$$\begin{aligned} w_k^i &\propto \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{0:k-1}^i|z_{1:k-1})}{q(x_k^i|x_{0:k-1}^i, z_{1:k})q(x_{0:k-1}^i|z_{1:k-1})} \\ &= \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} w_{k-1}^i \end{aligned} \tag{B.10}$$

Furthermore, if we only need a filtered estimate of posterior state density $p(x_k|z_{1:k})$, we can assume that the importance density depends only on the previous state and the current measurement by stating $q(x_k^i|x_{0:k-1}^i, z_{1:k}) = q(x_k|x_{k-1}, z_k)$. Then, we only need to store x_k^i and discard both the path $x_{0:k-1}^i$ and the previous observations $z_{1:k-1}$. The simplified update equations become

$$\begin{aligned} x_k^i &\sim q(x_k|x_{k-1}^i, z_k) \\ w_k^i &\propto w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \end{aligned} \tag{B.11}$$

and the posterior filtered density $p(x_k|z_{1:k})$ can be approximated by the following discrete representation

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \tag{B.12}$$

The problem with this recursive sampling iterations is that only one particle has non-negligable weight after a few iterations. This phenomenon is called the degeneracy problem and generally solved via resampling. Degeneracy can be measured by the

effective sample size N_{eff} that can be approximated via the following formula:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_p} (w_k^i)^2} \tag{B.13}$$

where a smaller N_{eff} indicates higher degeneracy since the variance is higher. Resampling with replacement is generally performed whenever the effective sample size N_{eff} drops below a certain threshold. The goal of resampling is eliminating the particles with lower weights by replacing them with a new set of particles drawn from the approximate discrete representation of the posterior filtered density given in Equation B.12.

B.2. Sequential Importance Resampling (SIR) Filter

The Sequential Importance Resampling (SIR) algorithm is a special case of SIS where (i) the importance density $q(x_k|x_{k1}^i, z_k)$ is chosen as the prior density $p(x_k|x_{k-1}^i)$, and (ii) resampling is applied in every time step. After resampling all the weights at time $k-1$ become $1/N_p$. Based on these choices, the update equations reduce to

$$\begin{aligned} x_k^i &\sim p(x_k|x_{k-1}^i) \\ w_k^i &\propto p(z_k|x_k^i) \end{aligned} \tag{B.14}$$

REFERENCES

1. Alemdar, H. and C. Ersoy, "Wireless Sensor Networks for Healthcare: A Survey", *Computer Networks*, Vol. 54, No. 15, pp. 2688–2710, 2010.

2. *Why Population Aging Matters: A Global Perspective*, Tech. rep., National Institute on Aging, U.S. Department of Health and Human Services, 2007.

3. Bamis, A., D. Lymberopoulos, T. Teixeira and A. Savvides, "The BehaviorScope Framework for Enabling Ambient Assisted Living", *Personal Ubiquitous Computing*, Vol. 14, No. 6, pp. 473–487, 2010.

4. Salah, A. A., T. Gevers, N. Sebe and A. Vinciarelli, "Challenges of Human Behavior Understanding", *First International Conference on Human Behavior Understanding*, HBU '10, pp. 1–12, 2010.

5. Alemdar, H., H. Ertan, O. D. Incel and C. Ersoy, "ARAS Human Activity Datasets in Multiple Homes with Multiple Residents", *7th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '13, pp. 232–235, 2013.

6. Tunca, C., H. Alemdar, H. Ertan, O. D. Incel and C. Ersoy, "Multimodal Wireless Sensor Network-Based Ambient Assisted Living in Real Homes with Multiple Residents", *Sensors*, Vol. 14, No. 6, pp. 9692–9719, 2014.

7. Alemdar, H., T. van Kasteren, M. E. Niessen, A. Merentitis and C. Ersoy, "A Unified Model for Human Behavior Modeling using a Hierarchy with a Variable Number of States", *IEEE International Conference on Pattern Recognition*, ICPR '14, Stockholm, Sweden, 2014.

8. Alemdar, H., C. Tunca and C. Ersoy, "Daily Life Behaviour Monitoring for Health Assessment Using Machine Learning: Bridging the Gap Between Domains", *Per-*

sonal and Ubiquitous Computing, Vol. 19, No. 2, pp. 303–315, 2015.

9. Alemdar, H., "Multi-Resident Human Behaviour Identification in Ambient Assisted Living Environments", *16th ACM International Conference on Multimodal Interaction*, ICMI '14, Istanbul, Turkey, 2014.

10. Alemdar, H., T. van Kasteren and C. Ersoy, "Using Active Learning to Allow Activity Recognition on a Large Scale", *International Joint Conference on Ambient Intelligence*, AmI '11, Amsterdam, Netherlands, 2011.

11. Alemdar, H., T. van Kasteren and C. Ersoy, "Activity Recognition with Hidden Markov Models Using Active Learning", *IEEE 19th Signal Processing and Communications Applications Conference*, SIU '11, Antalya, Turkey, 2011.

12. Chen, L., J. Hoey, C. D. Nugent, D. J. Cook and Z. Yu, "Sensor-based Activity Recognition", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42, No. 6, pp. 790–808, 2012.

13. Weinland, D., R. Ronfard and E. Boyer, "A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition", *Computer Vision and Image Understanding*, Vol. 115, No. 2, pp. 224–241, 2011.

14. Aggarwal, J. and M. Ryoo, "Human Activity Analysis: A Review", *ACM Computing Surveys*, Vol. 43, No. 3, pp. 16:1–16:43, 2011.

15. Chaaraoui, A. A., P. Climent-Pérez and F. Flórez-Revuelta, "A Review on Vision Techniques Applied to Human Behaviour Analysis for Ambient-Assisted Living", *Expert Systems with Applications*, Vol. 39, No. 12, pp. 10873 – 10888, 2012.

16. Popoola, O. and K. Wang, "Video-Based Abnormal Human Behavior Recognition: A Review", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42, No. 6, pp. 865–878, 2012.

17. Xu, X., J. Tang, X. Zhang, X. Liu, H. Zhang and Y. Qiu, "Exploring Techniques for Vision Based Human Activity Recognition: Methods, Systems, and Evaluation", *Sensors*, Vol. 13, No. 2, pp. 1635–1650, 2013.

18. Chaquet, J. M., E. J. Carmona and A. Fernández-Caballero, "A Survey of Video Datasets for Human Action and Activity Recognition", *Computer Vision and Image Understanding*, Vol. 117, No. 6, pp. 633–659, 2013.

19. Shotton, J., A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-time Human Pose Recognition in Parts from Single Depth Images", *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pp. 1297–1304, 2011.

20. Jalal, A., S. Kamal and D. Kim, "A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments", *Sensors*, Vol. 14, No. 7, pp. 11735–11759, 2014.

21. Holte, M., C. Tran, M. Trivedi and T. Moeslund, "Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 6, No. 5, pp. 538–552, 2012.

22. Khan, Z. and W. Sohn, "Abnormal Human Activity Recognition System Based on R-transform and Kernel Discriminant Technique for Elderly Home Care", *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 4, pp. 1843–1850, 2011.

23. Uddin, M. Z., N. D. Thang, J. T. Kim and T.-S. Kim, "Human Activity Recognition Using Body Joint-Angle Features and Hidden Markov Model", *ETRI Journal*, Vol. 33, No. 4, pp. 569–579, 2011.

24. Amiri, S., M. Pourazad, P. Nasiopoulos and V. Leung, "Non-intrusive Human Activity Monitoring in a Smart Home Environment", *IEEE 15th International Conference on e-Health Networking, Applications Services*, HealthCom '13, pp.

606–610, 2013.

25. Cheng, H., Z. Liu, Y. Zhao, G. Ye and X. Sun, "Real World Activity Summary for Senior Home Monitoring", *Multimedia Tools and Applications*, Vol. 70, No. 1, pp. 177–197, 2014.

26. Romdhane, R., C. Crispim, F. Bremond and M. Thonnat, "Activity Recognition and Uncertain Knowledge in Video Scenes", *10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '13, pp. 377–382, 2013.

27. Vacher, M., F. Portet, A. Fleury and N. Noury, "Development of Audio Sensing Technology for Ambient Assisted Living: Applications and Challenges", *International Journal of E-Health and Medical Communications*, Vol. 2, No. 1, pp. 35 – 54, 2011.

28. Vacher, M., B. Lecouteux, P. Chahuara, F. Portet, B. Meillon and N. Bonnefond, "The Sweet-Home Speech and Multimodal Corpus for Home Automation Interaction", *9th International Conference on Language Resources and Evaluation*, LREC '14, pp. 4499–4506, 2014.

29. Stork, J., L. Spinello, J. Silva and K. Arras, "Audio-based Human Activity Recognition Using Non-Markovian Ensemble Voting", *21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 509–514, 2012.

30. Karpov, A., L. Akarun, H. Yalçın, A. Ronzhin, B. Demiröz, A. Çoban and M. Zelezny, "Audio-Visual Signal Processing in a Multimodal Assisted Living Environment", *15th Annual Conference of the International Speech Communication Association*, INTERSPEECH '14, pp. 1023–1027, 2014.

31. Hollosi, D., J. Schroder, S. Goetze and J.-E. Appell, "Voice Activity Detection Driven Acoustic Event Classification for Monitoring in Smart Homes", *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ISABEL '10, 2010.

32. Zhan, Y. and T. Kuroda, "Wearable Sensor-based Human Activity Recognition from Environmental Background Sounds", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 5, No. 1, pp. 77–89, 2014.

33. Nguyen-Dinh, L.-V., U. Blanke and G. Tröster, "Towards Scalable Activity Recognition: Adapting Zero-effort Crowdsourced Acoustic Models", *12th International Conference on Mobile and Ubiquitous Multimedia*, MUM '13, pp. 1–10, 2013.

34. Mozer, M. C., "The Neural Network House: An Environment that Adapts to its Inhabitants", *AAAI Spring Symposium on Intelligent Environments*, pp. 110–114, 1998.

35. Helal, S., W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, "The Gator Tech Smart House: A Programmable Pervasive Space", *Computer*, Vol. 38, No. 3, pp. 50–60, 2005.

36. Abowd, G. D., A. F. Bobick, I. A. Essa, E. D. Mynatt and W. A. Rogers, "The Aware Home: A Living Laboratory for Technologies for Successful Aging", *AAAI-02 Workshop Automation as Caregiver*, 2002.

37. Tapia, E. M., S. S. Intille and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors", *International Conference on Pervasive Computing*, Pervasive '04, pp. 158–175, 2004.

38. Philipose, M., K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz and D. Hahnel, "Inferring Activities from Interactions with Objects", *IEEE Pervasive Computing*, Vol. 3, No. 4, pp. 50–57, 2004.

39. Fishkin, K., M. Philipose and A. Rea, "Hands-on RFID: Wireless Wearables for Detecting Use of Objects", *9th IEEE International Symposium on Wearable Computers*, ISWC '05, pp. 38–41, 2005.

40. Patterson, D., D. Fox, H. Kautz and M. Philipose, "Fine-grained Activity Recog-

nition by Aggregating Abstract Object Usage", *9th IEEE International Symposium on Wearable Computers*, ISWC '05, pp. 44–51, 2005.

41. Hodges, M. R. and M. E. Pollack, "An Object-use Fingerprint: The Use of Electronic Sensors for Human Identification", *9th International Conference on Ubiquitous Computing*, UbiComp '07, pp. 289–303, 2007.

42. Buettner, M., R. Prasad, M. Philipose and D. Wetherall, "Recognizing Daily Activities with RFID-based Sensors", *11th International Conference on Ubiquitous Computing*, UbiComp '09, pp. 51–60, 2009.

43. van Kasteren, T., A. Noulas, G. Englebienne and B. Kröse, "Accurate Activity Recognition in a Home Setting", *10th International Conference on Ubiquitous Computing*, UbiComp '08, pp. 1–9, 2008.

44. van Kasteren, T., *Activity Recognition for Health Monitoring Elderly Using Temporal Probabilistic Models*, Ph.D. Thesis, University of Amsterdam, Netherlands, 2011.

45. Cook, D. J., M. Schmitter-Edgecombe, A. Crandall, C. Sanders and B. Thomas, "Collecting and Disseminating Smart Home Sensor Data in the CASAS Project", *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009.

46. Singla, G., D. J. Cook and M. Schmitter-Edgecombe, "Recognizing Independent and Joint Activities Among Multiple Residents in Smart Environments", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 1, No. 1, pp. 57–63, 2010.

47. Gaddam, A., S. Mukhopadhyay and G. Gupta, "Elder Care Based on Cognitive Sensor Network", *IEEE Sensors Journal*, Vol. 11, No. 3, pp. 574–581, 2011.

48. Suryadevara, N. and S. Mukhopadhyay, "Wireless Sensor Network Based Home

Monitoring System for Wellness Determination of Elderly", *IEEE Sensors Journal*, Vol. 12, No. 6, pp. 1965–1972, 2012.

49. Ordonez, F. J., P. de Toledo and A. Sanchis, "Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors", *Sensors*, Vol. 13, No. 5, pp. 5460–5477, 2013.

50. Fatima, I., M. Fahim, Y.-K. Lee and S. Lee, "A Unified Framework for Activity Recognition-Based Behavior Analysis and Action Prediction in Smart Homes", *Sensors*, Vol. 13, No. 2, pp. 2682–2699, 2013.

51. Memon, M., S. R. Wagner, C. F. Pedersen, F. H. A. Beevi and F. O. Hansen, "Ambient Assisted Living Healthcare Frameworks, Platforms, Standards, and Quality Attributes", *Sensors*, Vol. 14, No. 3, pp. 4312–4341, 2014.

52. Bao, L. and S. S. Intille, "Activity Recognition from User-annotated Acceleration Data", *International Conference on Pervasive Computing*, Pervasive '04, 2004.

53. Chavarriaga, R., H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. del R. Millán and D. Roggen, "The Opportunity Challenge: A Benchmark Database for On-body Sensor-based Activity Recognition", *Pattern Recognition Letters*, Vol. 34, No. 15, pp. 2033 – 2042, 2013.

54. Ghasemzadeh, H. and R. Jafari, "Physical Movement Monitoring Using Body Sensor Networks: A Phonological Approach to Construct Spatial Decision Trees", *IEEE Transactions on Industrial Informatics*, Vol. 7, No. 1, pp. 66–77, 2011.

55. Kuo, C.-H., C.-T. Chen, T.-S. Chen and Y.-C. Kuo, "A Wireless Sensor Network Approach for Rehabilitation Data Collections", *IEEE International Conference on Systems, Man, and Cybernetics*, SMC '11, pp. 579–584, Anchorage, Alaska, 2011.

56. Avci, A., S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga, "Ac-

tivity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey", *23rd International Conference on Architecture of Computing Systems*, ARCS '10, 2010.

57. Bulling, A., U. Blanke and B. Schiele, "A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors", *ACM Computing Surveys*, Vol. 46, No. 3, pp. 1–33, 2014.

58. Lara, O. and M. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors", *IEEE Communications Surveys Tutorials*, Vol. 15, No. 3, pp. 1192–1209, 2013.

59. Kose, M., O. D. Incel and C. Ersoy, "Online Human Activity Recognition on Smart Phones", *2nd International Workshop on Mobile Sensing*, Beijing, China, 2012.

60. Assam, R. and T. Seidl, "Activity Recognition From Sensors Using Dyadic Wavelets and Hidden Markov Model", *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMob '14, pp. 442–448, 2014.

61. Dernbach, S., B. Das, N. C. Krishnan, B. Thomas and D. Cook, "Simple and Complex Activity Recognition through Smart Phones", *8th International Conference on Intelligent Environments*, IE '12, pp. 214–221, 2012.

62. Incel, O., M. Kose and C. Ersoy, "A Review and Taxonomy of Activity Recognition on Mobile Phones", *Bionanoscience*, Vol. 3, No. 2, pp. 145–171, 2013.

63. Coskun, D., O. Incel and A. Ozgovde, "Position-aware Activity Recognition on Mobile Phones", *22nd Signal Processing and Communications Applications Conference*, SIU '14, pp. 1930–1933, 2014.

64. Altini, M., R. Vullers, C. Van Hoof, M. van Dort and O. Amft, "Self-calibration

of Walking Speed Estimations Using Smartphone Sensors", *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 10–18, 2014.

65. Kientz, J. A., S. N. Patel, B. Jones, E. Price, E. D. Mynatt and G. D. Abowd, "The Georgia Tech Aware Home", *Human Factors in Computing Systems*, 2008.

66. Intille, S. S., K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia and P. Kaushik, "A Living Laboratory for the Design and Evaluation of Ubiquitous Computing Technologies", *Conference on Human Factors in Computing Systems*, pp. 1941–1944, 2005.

67. Gallissot, M., J. Caelen, N. Bonnefond, B. Meillon and S. Pons, *Using the Multicom Domus Dataset*, Research Report RR-LIG-020, LIG, Grenoble, France, 2011.

68. Cook, D. J., "Learning Setting-generalized Activity Models for Smart Spaces", *IEEE Intelligent Systems*, Vol. 27, No. 1, pp. 32–38, 2012.

69. Arduino, *Arduino Fio Platform*, 2005, http://www.arduino.cc, [Accessed January 2015].

70. Digi, *Xbee ZigBee Module*, 2005, http://www.digi.com/xbee, [Accessed January 2015].

71. Oliver, N., E. Horvitz and A. Garg, "Layered Representations for Human Activity Recognition", *Fourth IEEE International Conference on Multimodal Interfaces*, pp. 3–8, 2002.

72. van Kasteren, T. L. M., G. Englebienne and B. J. Kröse, "Hierarchical Activity Recognition Using Automatically Clustered Actions", *International Joint Conference on Ambient Intelligence*, AmI '11, pp. 82–91, 2011.

73. Niessen, M. E., T. L. M. Van Kasteren and A. Merentitis, "Hierarchical Sound

Event Detection", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

74. Ghazvininejad, M., H. R. Rabiee, N. Pourdamghani and P. Khanipour, "HMM Based Semi-supervised Learning for Activity Recognition", *International Workshop on Situation Activity & Goal Awareness*, SAGAware '11, pp. 95–100, 2011.

75. Piyathilaka, L. and S. Kodagoda, "Gaussian Mixture Based HMM for Human Daily Activity Recognition Using 3D Skeleton Features", *8th IEEE Conference on Industrial Electronics and Applications*, ICIEA '13, pp. 567–572, 2013.

76. Shaikh, M. A. M., K. Hirose and M. Ishizuka, "The Systemic Dimension of Globalization", P. Pachura (Editor), *Recognition of Real-World Activities from Environmental Sound Cues to Create Life-Log*, InTech, 2011.

77. van Kasteren, T., G. Englebienne and B. Kröse, "Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software", *Activity Recognition in Pervasive Intelligent Environments*, pp. 165–186, Springer, 2011.

78. Lee, Y.-S. and S.-B. Cho, "Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3D Accelerometer", *6th International Conference on Hybrid Artificial Intelligent Systems*, HAIS'11, pp. 460–467, 2011.

79. Mannini, A. and A. M. Sabatini, "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers", *Sensors*, Vol. 10, No. 2, pp. 1154–1175, 2010.

80. Fine, S., Y. Singer and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications", *Machine Learning*, Vol. 32, pp. 41–62, 1998.

81. Murphy, K. and M. A. Paskin, "Linear Time Inference In Hierarchical HMMs", *Advances in Neural Information Processing Systems*, NIPS '01, 2001.

82. Karaman, S., J. Benois-Pineau, R. Mégret, J. Pinquier, Y. Gaestel and J.-F. Dartigues, "Activities of Daily Living Indexing by Hierarchical HMM for Dementia Diagnostics", *9th International Workshop on Content-Based Multimedia Indexing*, CBMI '11, pp. 79–84, 2011.

83. Celeux, G. and J.-B. Durand, "Selecting Hidden Markov Model State Number With Cross-Validated Likelihood", *Computational Statistics*, Vol. 23, No. 4, pp. 541–564, 2008.

84. Beal, M. J., Z. Ghahramani and C. E. Rasmussen, "The Infinite Hidden Markov Model", *Advances in Neural Information Processing Systems*, NIPS '02, 2002.

85. Heller, K., Y. W. Teh and D. Görür, "Infinite Hierarchical Hidden Markov Models", *International Conference on Artificial Intelligence and Statistics*, 2009.

86. Alvarez, G. G. and N. T. Ayas, "The Impact of Daily Sleep Duration on Health: A Review of the Literature", *Progress in Cardiovascular Nursing*, Vol. 19, No. 2, pp. 56–59, 2004.

87. Gangwisch, J. E., S. B. Heymsfield, B. Boden-Albala, R. M. Buijs, F. Kreier, T. G. Pickering, A. G. Rundle, G. K. Zammit and D. Malaspina, "Short Sleep Duration as a Risk Factor for Hypertension: Analyses of the First National Health and Nutrition Examination Survey", *Hypertension*, Vol. 47, No. 5, pp. 833–839, 2006.

88. van Kasteren, T., H. Alemdar and C. Ersoy, "Effective Performance Metrics for Evaluating Activity Recognition Methods", *Second Workshop on Context-Systems Design, Evaluation and Optimisation*, 2011.

89. Pavel Dohnálek, T. P., Petr Gajdoš, "Human Activity Recognition: Classifier Performance Evaluation on Multiple Datasets", *Journal of Vibroengineering*, Vol. 16, No. 3, pp. 1523–1534, 2014.

90. Ward, J., P. Lukowicz and H. Gellersen, "Performance Metrics for Activity Recognition", *ACM Transactions on Information Systems and Technology*, Vol. 2, No. 1, 2011.

91. Lasserre, J. and C. M. Bishop, "Generative or Discriminative? Getting the Best of Both Worlds", *Bayesian Statistics*, Vol. 8, pp. 3–24, 2007.

92. Waibel, A., T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, "Phoneme Recognition Using Time-delay Neural Networks", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 3, pp. 328–339, 1989.

93. Crandall, A. S. and D. J. Cook, "Coping with Multiple Residents in a Smart Environment", *Journal of Ambient Intelligence and Smart Environments*, Vol. 1, No. 4, pp. 323–334, 2009.

94. Guo, P. and Z. Miao, "Multi-person Activity Recognition through Hierarchical and Observation Decomposed HMM", *IEEE International Conference on Multimedia and Expo*, ICME '10, pp. 143–148, 2010.

95. Wilson, D. H. and C. Atkeson, "Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors", *Third International Conference on Pervasive Computing*, Persuasive '05, pp. 62–79, 2005.

96. Arulampalam, M., S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 174–188, 2002.

97. Kirubarajan, T. and Y. Bar-Shalom, "Probabilistic Data Association Techniques for Target Tracking in Clutter", *Proceedings of the IEEE*, Vol. 92, No. 3, pp. 536–557, 2004.

98. Jaward, M., L. Mihaylova, N. Canagarajah and D. Bull, "Multiple Object Tracking Using Particle Filters", *IEEE Aerospace Conference*, 2006.

99. Blackman, S. S., "Multiple Hypothesis Tracking for Multiple Target Tracking", *IEEE Aerospace and Electronic Systems Magazine*, Vol. 19, No. 1, pp. 5–18, 2004.

100. Tolstikov, A., C. Phua, J. Biswas and W. Huang, "Multiple People Activity Recognition Using MHT over DBN", *9th International Conference on Smart Homes and Health Telematics*, ICOST '11, pp. 313–318, 2011.

101. Ghahramani, Z. and M. I. Jordan, "Factorial Hidden Markov Models", *Machine Learning*, Vol. 273, No. 29, pp. 245–273, 1997.

102. Deoras, A. and M. Hasegawa-Johnson, "A Factorial HMM Approach to Simultaneous Recognition of Isolated Digits Spoken by Multiple Talkers on One Audio Channel", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1 of *ICASSP '04*, pp. 861–864, 2004.

103. Husmeier, D., "Discriminating Between Rate Heterogeneity and Interspecific Recombination in DNA Sequence Alignments with Phylogenetic Factorial Hidden Markov Models", *Bioinformatics*, Vol. 21, No. 2, pp. 166–172, 2005.

104. Chen, C., J. Liang, H. Zhao, H. Hu, J. Tian and J. Tian, "Factorial HMM and Parallel HMM for Gait Recognition", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 39, No. 1, pp. 114–123, 2009.

105. Gordon, N., D. Salmond and A. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", *IEE Proceedings F, Radar and Signal Processing*, Vol. 140, No. 2, pp. 107–113, 1993.

106. Settles, B. and M. Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks", *Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 2008.

107. Anderson, B., S. Siddiqi and A. Moore, *Sequence Selection for Active Learning*, Tech. rep., Carnegie Mellon University, 2006.

108. Settles, B., *Active Learning*, Morgan&Claypool, 2012.

109. Liu, R., T. Chen and L. Huang, "Research on Human Activity Recognition Based on Active Learning", *International Conference on Machine Learning and Cybernetics*, ICMLC '10, pp. 285–290, 2010.

110. Stikic, M., K. van Laerhoven and B. Schiele, "Exploring Semi-supervised and Active Learning for Activity Recognition", *12th IEEE International Symposium on Wearable Computers*, ISWC '08, pp. 81–88, 2008.

111. Truyen, T., H. Bui, D. Phung and S. Venkatesh, "Learning Discriminative Sequence Models from Partially Labelled Data for Activity Recognition", *PRICAI 2008: Trends in Artificial Intelligence*, pp. 903–912, 2008.

112. Ho, Y., C. Lu, I. Chen, S. Huang, C. Wang and L. Fu, "Active-learning Assisted Self-reconfigurable Activity Recognition in a Dynamic Environment", *IEEE International Conference on Robotics and Automation*, pp. 813–818, 2009.

113. Lewis, D. and J. Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning", *11th International Conference on Machine Learning*, ICML '94, pp. 148–156, 1994.

114. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.

115. Pehlivan, N., H. Alemdar, C. Tunca and C. Ersoy, "Human Activity Recognition and Interpretation in Smart Home: An Annotation and Data Visualization Tool", *Akademik Bilişim*, AB '15, Eskişehir, Turkey, 2015.

116. van Kasteren, T. L. M., G. Englebienne and B. J. A. Kröse, "Transferring Knowledge of Activity Recognition Across Sensor Networks", *Pervasive Computing*, pp. 283–300, 2010.