# A Computational Framework for Wearable Accelerometer-Based Activity and Gesture Recognition

by

Narayanan Chatapuram Krishnan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

December 2010

A Computational Framework for Wearable Accelerometer-Based

Activity and Gesture Recognition

by

Narayanan Chatapuram Krishnan

has been approved

October 2010

Graduate Supervisory Committee:

Sethuraman Panchanathan, Chair
Hari Sundaram
Jieping Ye
Baoxin Li
Dianne Cook

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

Advances in the area of ubiquitous, pervasive and wearable computing have resulted in the development of low band-width, data rich environmental and body sensor networks, providing a reliable and non-intrusive methodology for capturing activity data from humans and the environments they inhabit. Assistive technologies that promote independent living amongst elderly and individuals with cognitive impairment are a major motivating factor for sensor-based activity recognition systems. However, the process of discerning relevant activity information from these sensor streams such as accelerometers is a non-trivial task and is an on-going research area. The difficulty stems from factors such as spatio-temporal variations in movement patterns induced by different individuals and contexts, sparse occurrence of relevant activity gestures in a continuous stream of irrelevant movements and the lack of real-world data for training learning algorithms.

This work addresses these challenges in the context of wearable accelerometer-based simple activity and gesture recognition. The proposed computational framework utilizes discriminative classifiers for learning the spatio-temporal variations in movement patterns and demonstrates its effectiveness through a real-time simple activity recognition system and short duration, non-repetitive activity gesture recognition. Furthermore, it proposes adaptive discriminative threshold models trained only on relevant activity gestures for filtering irrelevant movement patterns in a continuous stream. These models are integrated into a gesture spotting network for detecting activity gestures involved in complex activities of daily living. The framework addresses the lack of real world data for training, by using auxiliary, yet related data samples for training in a transfer learning setting. Finally the problem of predicting activity tasks involved in the execution of a complex activity of daily living is described and a solution based on hierarchical Markov models is discussed and evaluated.

ACKNOWLEDGMENTS

This work would not have been possible without the constant support and guidance of my mentor and supervisor, Prof. Sethuraman Panchanathan. I thank him profusely for providing me with the best environment to work. I am grateful to him for giving me the freedom to explore and the excellent opportunities to learn and grow as a researcher.

I would like to thank my committee members, Dr. Hari Sundaram, Dr. Jieping Ye, Dr. Baoxin Li and Dr. Diane Cook for sparing their valuable time to interact with me and for sharing their inputs and feedback. I am grateful to them for accommodating my requests and deadlines.

I would like to thank all the members of the Center for Cognitive Ubiquitous Computing at Arizona State University. I would like to specially thank Dr.Dirk Colbry, Colin, Ashok and Prasanth, for having supported me with my research through the formative years to its completion. Thanks to Sreekar Vineeth and Troy – it was an enriching and learning experience working with you. I thank Rita, David, Mohammad, Jeff, Jessie, Cindy, Daniel, Shayok, Sunaad, Mike, Ramkiran and Hiranmayi for all their help. To Dr. John Black and Dr.Terri Hedgpeth, sincere and heartfelt thanks for helping me when I was new to this part of the world; I thank Dr. Gaurav Pradhan for his ever willing support and understanding. I would also like to thank Dr.Morris Golbderg for providing his insights into the work during our discussions.

I would like to thank the faculty and staff of the Department of Computer Science and Engineering for all their support and effort to make my academic life a pleasant and memorable one. I also would like to thank Dr.Vimla Patel for all the support.

I would like to acknowledge the National Science Foundation and the state of Arizona who sponsored my research (NSF-ITR grant IIS-0326544 and NSF IIS-0739744).

Thanks to all my friends for supporting me all through the years. I am deeply indebted to my parents and brother for everything they have provided me. To my Lord, I offer this to thee….

TABLE OF CONTENTS

LIST OF TABLES ..... viii

LIST OF FIGURES ..... ix

CHAPTER Page

1. INTRODUCTION ..... 1

Motivation ..... 2

Challenges of automated activity recognition ..... 5

Sensing challenges ..... 5

Computational Challenges ..... 6

Goal of dissertation ..... 8

Contributions from the dissertation ..... 11

Outline ..... 12

2. SENSORS FOR HUMAN ACTIVITY RECOGNITION ..... 14

Introduction ..... 14

Sensors for activity recognition ..... 21

Wearable sensor based activity recognition ..... 23

Environmental sensor based activity recognition ..... 24

Video based activity recognition ..... 25

Conclusion ..... 25

3. SIMPLE ACTIVITY RECOGNITION ..... 26

Introduction ..... 26

Related work ..... 27

Experimental framework ..... 30

Description of the dataset ..... 30

Feature extraction ..... 30

Isolated recognition ..... 32

Post classification label smoothing for continuous classification ..... 34

CHAPTER Page

Human factors assisting in continuous classification ..... 36

Real-time simple activity recognition ..... 40

Data capture ..... 40

Data processing and feature extraction ..... 41

Classification ..... 41

Computational complexity ..... 43

Conclusion ..... 46

4. ACTIVITY GESTURE RECOGNITION ..... 49

Introduction ..... 49

Related work ..... 50

Template based approaches ..... 50

Generative approaches ..... 51

Discriminative approaches ..... 53

Data collection ..... 55

Feature analysis ..... 60

Classification- results and discussion ..... 65

Subject adaptation ..... 68

Accelerometer configuration ..... 69

Performance on activity gesture set CS2 ..... 70

Performance on activity gesture set CS3 ..... 71

Conclusion ..... 75

5. ACTIVITY GESTURE SPOTTING ..... 77

Introduction ..... 77

Related work ..... 78

Spotting without out-of-vocabulary samples ..... 79

Spotting with out-of-vocabulary samples ..... 80

CHAPTER Page

Threshold models for detecting out-of-vocabulary samples ..... 82

Multiple one class SVM based threshold model ..... 82

Adaptive boosting based threshold model ..... 85

Large margin Gaussian mixture model ..... 89

Comparison of out-vocabulary sample detection methods ..... 92

Performance on activity gesture set CS1 ..... 102

Gesture spotting network ..... 103

Results and discussion ..... 105

Conclusion ..... 109

6. LEARNING FROM AUXILIARY DATA SAMPLES ..... 111

Introduction ..... 111

Related work ..... 112

Instance transfer ..... 113

Feature transfer ..... 116

Parameter transfer ..... 117

Relational transfer ..... 118

Cost sensitive boosting for transfer learning ..... 119

Cost estimation ..... 122

Relation to other boosting based transfer learning approaches ..... 122

Theoretical analysis of the algorithm ..... 124

Experimental evaluation ..... 129

Datasets ..... 129

Comparison methods ..... 130

Results and discussion ..... 132

Limitations ..... 138

Conclusion ..... 139

CHAPTER Page

7. ACTIVITY TASK PREDICTION ..... 140
Introduction ..... 140
Related work ..... 142
Hierarchical Markov chain based prediction ..... 143
Clustering of tasks ..... 145
Hierarchical state space ..... 146
Task prediction algorithm ..... 147
Experimental evaluation ..... 149
Dataset description ..... 149
Comparison methods and measures ..... 150
Results and discussion ..... 152
Conclusion ..... 154
8. Conclusion and future directions ..... 156
Limitations and future directions ..... 158
Dissemination and outcomes ..... 159
REFERENCES ..... 161

LIST OF TABLES

Table Page

1. Subject independent, adaptive and dependent accuracies for the different discriminative classifiers ..... 34
2. The aggregate confusion matrix obtained from subject independent 10 folds cross validation using AdaBoost trained on combined features ..... 35
3. Transition matrix for the CFG defined for simple activities ..... 39
4. Computational complexity of the features extracted, where $N$ is the number of data samples ..... 45
5. Summary of past work on simple activity recognition using accelerometers presented in a chronological order. The recognition accuracies correspond to the subject independent classification scenario ..... 48
6. The mock and semi-naturalistic scenarios used for data capture ..... 58
7. Description of the activity gestures collected using the ZStar accelerometers for CS3 ..... 59
8. Dominant Features for each of the gestures as calculated using AdaBoost (X1, Y1, Z1 and X2, Y2, Z2 are the X, Y and Z axes of the accelerometers placed on the wrist and elbow respectively ..... 66
9. Aggregate confusion matrix obtained for AdaBoost classification ..... 69
10. Classification accuracies for different configurations of the accelerometers ..... 71
11. Isolated recognition performance on CS2 dataset through classifiers trained on CS1, CS2 and a combination both the datasets ..... 72
12. Aggregate confusion matrix, precision and recall values for the individual gestures obtained from RBF kernel SVM ..... 73
13. Aggregate confusion matrix, precision and recall values for the individual gestures obtained through AdaBoost with decision stump as the weak learner ..... 74

Table Page

14. Aggregate confusion matrix, precision and recall values for the individual gestures obtained through HMM ....74

15. Aggregate confusion matrix, precision and recall values for the individual gestures obtained through k-NN ....75

16. Description of the in and out of vocabulary label split for the datasets used in the study....94

17. Accuracies on in-vocabulary label samples ....95

18 Area under the ROC curve (AUC) values for the different classifiers on datasets. For each dataset the classifier with the highest AUC value has been bold faced. ....99

19. Accuracy of the threshold models on Activity Gestures from CS1 ....103

20. Gesture spotting test using the AdaBoost based threshold model....109

21. Gesture spotting test using MOC-SVM based threshold model ....109

22. Gesture spotting test using HMM based threshold model ....109

23. Description of the datasets used for the experiments....132

24. Description of the baseline methods. The methods using decision stumps were used only for activity gesture dataset ....132

25. Classification accuracies of the different techniques on Newsgroup data splits ....134

26. Accuracies of the different baseline techniques on the Activity Gesture AG CS1 and CS2 datasets ....135

27. Transformation of the original state space to hierarchical state space. (TO- Take Oil, POiB - Pour Oil into the bowl, ToC - Twist on Cap, SB - Stir Bowl....149

28. Description of the tasks for each of the two activities ....151

LIST OF FIGURES

Figure Page

1: Activities of daily living as described in Katz index [153] ..... 3

2: Some of the common wireless accelerometers available off-the-self. The images have been adapted from the [155][156][157][162] and [158] ..... 16

3: Objects of everyday use marked with RFID tags(indicated by the arrows) and the iBracelet RFID reader. The figure has been adapted from Stikic et al [132] ..... 17

4: A visualization of the motion sensor activity for the 'wash hands' task as illustrated by Singla et al [164]. The circles marked as M represents the location of the motion sensor in the indoor environment ..... 18

5: An illustration of common indoor locations and objects attached with reed switchess. The figure has been adapted from the work of Tapia et al [38] ..... 19

6: Pressure sensors used to measure the foot print of an individual. The images have been adapted from Orr and Abowd[166], Richardson et al [167] and Yin et al[210] ..... 20

7: Biaxial accelerometer data for different simple activities from an accelerometer placed on the ankle ..... 27

8: Projection of the simple activity data samples on to three dimensions obtained through PCA ..... 32

9: Class-wise acuracies using AdaBoost trained on the three features [1- Walking. 2-Sitting, 3 - Standing, 4-Running, 5-Bicycling, 6-Lying down, 7-Climbing Stairs] ..... 33

10: llustration of the post classification label-smoothing framework ..... 36

11: Output of static and temporal AdaBoost compared against the ground truth ..... 37

12: Accelerometer placement for the real-time simple activity recognition system ..... 41

13: Illustration of the software developed for real-time recognition of simple activities ..... 43

Figure Page

14: Real-time continuous example comparison of AdaBoost with the ground truth ..... 44

15: Comparing the normalized computational time for different parameters of the system against the performance measured as accuracy ..... 46

16: (a) Illustrates the different locations for placing the accelerometer, (b) depicts the data capture setup and (c) shows the annotation software that synchronizes the video and accelerometer data ..... 57

17: ZStar 3 accelerometer and the ZigBee wireless receiver manufactured by Freescale ..... 58

18: Activity gesture samples from the gesture dataset CS3 ..... 60

19: The action samples collected from one subject using the accelerometer placed on the right wrist. The RGB lines stand for the x-, y- and z-axis respectively. The continuous stream of data was annotated offline using the video recording. (a) Samples recorded from a single data capture session. (b) One representative sample for each of the actions ..... 61

20: Pair wise Euclidean distance between samples of a single subject ..... 62

21: Principal Component Analysis of activity gesture samples from all the subjects ..... 63

22: The weights for the different features as computed by AdaBoost and SVM ROC for each of the activity gesture. The final plot is the comparison of the performance of the reduced feature set as derived through AdaBoost and the original feature set. ..... 65

23: Training error using AdaBoost across iterations for the different activity gestures ..... 67

24: Subject independent, adaptive and dependent accuracies for AdaBoost, HMM and k-NN using the data from the accelerometer on right wrist ..... 68

25: Subject adaptive accuracies obtained using AdaBoost and k-NN for varying number of test subject samples used for training ..... 70

Figure Page

26: Precision and recall values for the different classifiers obtained on the data belonging to the case study 3 (CS3). ..... 73

27: Decision contours for a resular SVM ..... 85

28: Decision contours for a one class SVM model (OCSVM) ..... 85

29: Multiple one class SVM (MOC-SVM) Decision surface obtained on synthetic data - three Gaussians centered at [1 1], [-1 1] and [0 -1] denoted by the circles, squares and diamonds respectively. ..... 85

30: The AdaBoost algorithm as described by Freund et al [136] ..... 87

31: The distribution of samples belonging to the three classes ..... 89

32: Illustration of the decision boundary for the valid sample as obtained by AdaBoost based threshold model on a three-label classification scenario. ..... 89

33: ROC curves for out-vocabulary label detection on USPS and SatImage datasets. Red- Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM ..... 100

34: ROC curves for out-vocabulary label detection on AGCS3 and Iris datasets. Red- Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM ..... 101

35: ROC curves for out-vocabulary label detection on Yeast and OSR datasets. Red- Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM ..... 102

36: Illustration of the performance of OC-SVM (Blue) and MOC-SVM (Green) for detecting invalid gestures samples for varying values of $\nu$. ..... 103

37: Gesture-spotting network ..... 104

38: Evolution of the likelihood of the different gesture labels and the adaptive threshold model over time ..... 106

Figure Page

39: Gesture spotting on continuous data collected in CS2. The figure on the left side is obtained from Adaboost and the right side figure is for SVM ..... 110
40: TrAdaBoost algorithm of Dai et al [107] ..... 115
41: RecVsTalk error curves ..... 136
42: RecVvsSci error curves ..... 136
43: SciVsTalk error curves ..... 137
44: CompVsSci error curves ..... 137
45: CompVsTalk error curves ..... 137
46: RecVsComp error curves ..... 137
47: Difference in the accuracies for the ratio between Td and Ts at 1% ..... 137
48: Difference in the accuracies for the ratio between Tdd and Ts at 2% ..... 137
49: Difference in the accuracies for the ratio between Td and Ts at 3% ..... 138
50: Difference in the accuracies for the ratio between Td and Ts at 4% ..... 138
51: Difference in the accuracies for the ratio between Td and Ts at 5% ..... 138
52: Difference in the accuracies for the ratio between Td and Ts at 10% ..... 138
53: Difference in the accuracies for the ratio between Td and Ts at 20% ..... 138
54: Difference in the accuracies for the ratio between Td and Ts at 50% ..... 138
55: Difference in the accuracies between CTrAdaBoost and the second best technique as an average over all the datasets for different ratios between Ts and Td ..... 139
56: Trace of sum of weights for samples in Td across boosting iterations ..... 140
57: Trace of sum of weights for samples in Ts across boosting iterations ..... 140
58: Tasks in the complex activity of making brownies ..... 142

Figure Page

59: The clusters in the activity graph for the complex activity 'making eggs' after applying a threshold. Connected components of the graph that are color-coded belong to a single cluster. Tasks with no color belong to no cluster and indicate the uncertainly of their occurrence during the sequence of the activity tasks. ..... 148

60: Matching accuracies for the task prediction for Markov Chain (MC), Mixture Transition Distribution (MTD) and Hierarchical state space Markov Chain (HMC)..... 155

61: Average DTW distance measures across all the folds for the complete task sequence prediction for the three models..... 156

62: Normalized DTW distance measure for each of the partial sequences for a particular subject for the activity Make Eggs..... 156

Chapter 1

# INTRODUCTION

Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents' actions and the environmental conditions. Since the 1980s, this research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as medicine. The ability to build computing systems that can observe, understand and act on day-to-day physical human activity has long been a goal of computing research. Such systems have profound conceptual and practical implications to quality of our life. Since the ability to reason and act based on activity is one of the central aspects of human intelligence, from a conceptual point of view, such a system could cast light on the computational models of human intelligence. More tangibly, perhaps, machines that can reason about human activity could aid humans in aspects of their daily life that are considered today outside the domain of machines.

The past 6-7 years has seen significant advancement in the area of ubiquitous, pervasive and wearable computing resulting in the development of a variety of low bandwidth, data rich environmental and body sensor networks. These sensors provide a reliable and non-intrusive methodology for capturing activity data from humans and the environments they inhabit. This paradigm of using multitude of low bandwidth, highly specific sensors for activity recognition is commonly known as sensor based activity recognition. Assistive technologies that promote independent living amongst individuals with cognitive impairment are a major motivating factor for sensor based activity recognition systems. These systems use sensor data to determine the activities that a person is trying to perform and optionally provide prompts, warnings, or other kinds of interventions to help the person perform the activities safely and independently. Automated recognition of an individual's daily activities is also useful for creating automated life logs to understand changes in the individual's behavior over long periods of time, thereby serving as a tool for monitoring cognitive decline. Systems that understand user activity patterns could be

used in new architectural design tools to identify and rank user needs and preferences over time. Activity recognition in real-time could also allow the development of just-in-time learning environments that educate and inform people by presenting information at the right time as they move in the environment. Security systems that are able to create a model of people's activities and behavior over time could predict intent and motive as people interact with the environment. Finally knowing people's activities could enable more responsive systems to automate tasks such as lighting and HVAC control.

While there are multiple applications for activity recognition as described in the previous paragraph, the work performed in this dissertation is motivated by the need for developing assistive technologies for individuals with mild cognitive impairment or Alzheimer's disease and in general for elder care. The next section describes in more detail the motivation for this work in the context of developing assistive technologies for elderly care

**1.1 Motivation**

As the world's population is aging, so is the number of elderly individuals with physical and cognitive impairments unable to have an independent functional existence. Proper care through nursing and institutionalization has to be provided to these individuals to help them lead a safe life. The increasing health care and nursing costs place a tremendous stress on the society and the government. Development of smart technologies that extend the time that individuals can live independently in their own homes by increasing functional independence is of significant value to both individuals and society. The basic requirement of these technologies is the ability to recognize and track the activities of daily living for these individuals. Though recent advances in the area of smart environments [148, 149, 150, 152 ] demonstrate the ability to recognize activities, research still needs to be carried out to make these technologies effective and reach their target population.

There are a variety of physical activities that an individual executes in his/her daily life. Within this spectrum of activities, are Activities of Daily Living (ADL) and Instrumental Activities of Daily Living (IADL) that is of immense interest to the medical community in the context of

elderly care. Figure 1 illustrates some of these activities as described in Katz index [153] referred popularly by medical practitioners for evaluating elderly individuals for independent living. These ADLs and IADLs constitute some of the activities an individual executes to lead an independent existence and is very relevant in the context of safe and independent living for the elderly.




Figure 1: Activities of daily living as described in Katz index [153]

The ADLs and IADLs include activities such as ambulation, eating and drinking, food preparation, hygiene related activities, folding laundry, working a computer etc. These activities represent varying levels of difficulty in terms of sensing and processing. At one end of spectrum are certain activities such as ambulation that are predominantly defined by movements or posture, while at the other end are activities that are defined by the unique objects of interaction such as ironing and working on a computer. Activity recognition systems for detecting this subset of physical activities that are complex in nature require a variety of sensing and processing methodologies. A single recognition system will not be sufficient for recognizing this variety of activities. To assist in the process of selecting appropriate sensing modalities and processing algorithms, these activities can be divided into three categories based on the difficulty in sensing and processing. These categories are:

- **Simple:** This category includes all the activities that are defined by a unique posture or movement of body parts. Examples of activities in this category include ambulation such as sitting, standing, walking, climbing stairs etc. While activities such as sitting and standing are predominantly defined by a static body posture, a repetitive movement of body defines walking and climbing stairs. The body movement and postural information is sufficient to develop systems for recognizing these activities. Sensing modalities that can capture the movement information form an essential aspect of these systems.
- **Medium:** The activities in this category are defined by a unique posture or movement of body parts in association with one or more objects. More than the body movement, it is object of interaction that gives away these activities. Activities such as setting the table and answering the phone fall under this category. One can easily recognize the activity setting a dinner table, with the information that the individual is interacting with multiple plates, cups and cutlery within a short span of time. Similarly one can easily predict that the activity is ironing clothes, if it is known that the object of interaction is an 'iron'. The movements involved in setting the plates or ironing is immaterial to a large extent. Thus the sensing modalities for these activities must focus on extracting the object information.
- **Complex:** This category constitutes activities that most difficult to recognize and track through automated systems. The complex activities can be further divided into smaller units called as tasks. The activity is executed through the execution of a sequence of its tasks. A unique movement called as activity gesture along with an interacting object defines each of the tasks. An example of a complex activity is making a cup of tea. This activity involves a number of tasks such as scooping tealeaves with a spoon, pouring water into a kettle, turning on the stove etc. Each of these tasks consists of a movement of the hand defined by scoop, pour, turning on, associated with the corresponding objects spoon, tealeaves, kettle and stove respectively. While information about the objects alone is sufficient to infer the activity, the gestural information is necessary for accurately

tracking and predicting the tasks. The sensing infrastructure for these activities must be able to extract the object as well as the movement information.

This tremendous variation in the structure of the activity poses significant challenges for developing sensing modalities and processing tools automated recognition. *The scope of this dissertation is limited to the recognition of the movement patterns in simple activities and activity gestures through wearable accelerometers.* The work presented in this dissertation has the potential to be integrated into a larger system for recognizing and tacking activities. The next section briefly discusses some of the sensing related challenges before delving into the computational challenges involved in accelerometer based activity and activity gesture recognition.

### 1.2 Challenges of automated activity recognition

As illustrated in the previous section recognizing physical activities of daily living poses significant challenges for developing sensing and processing tools. The focus of this dissertation is on the computational challenges rather than sensing challenges. In particular it deals with the computational challenges of activity and activity gesture recognition using wearable accelerometers. These challenges are discussed in the subsequent subsection. However for the sake of completeness, a few of sensing challenges are described in the following subsection.

#### 1.2.1 Sensing challenges for activity recognition

Development of sensing modalities for capturing activity information reliably and efficiently is a major challenge. Advances in wearable, ubiquitous and pervasive technologies have led to the development of a variety of sensors that can be used for capturing information regarding human activities. At one end of the spectrum are wearable sensors such as inertial sensors that glean movement information off the human body, and at the other end are environmental sensors such as motion detectors and RFID tags that sense location and object information. Chapter 2 gives an overview of some of these sensors commonly used for activity recognition. Even with these advances a lot of research still needs to be carried out for easy adaptation of these sensors for activity recognition in real-life settings.

Though current wearable sensors have been miniaturized a significant extent, research still needs to be carried out for their integration into everyday clothing [154] to make it unobtrusive and easy to use. Challenges related to power consumption and wireless signal transmissions are other issues that require further exploration. RFID technology has been made very popular by the retail industry due to its ease of use and efficiency. The main challenge that once faces with RFID technology in the context of activity recognition is its range of detection. Typical RFID reader either have >1m or <5cm as its range of detection. This range is insufficient in the context of identifying objects of interaction for activity recognition. There are other sensing related challenges such as optimal placement of sensors in the environment, communication protocols for reliable and safe transmission of the sensor data etc.

While wearable sensors offer a very unique and clean mechanism for capturing the movement information, there are many practical and usability issues that still have to addressed by the research community to facilitate wide spread adoption of the modality. The state of the art wearable systems require an accurate positioning and alignment of the sensing elements on the body. This is typically ensured by either using tight-fitting garments or by directly attaching the sensing elements on the intended location using tapes. However in many practical applications it is infeasible to impose these constraints. Using comfortable everyday garments for wearable computing remains a challenging research, since a fixed sensor position in relation to the body cannot be guaranteed. Even in the scenario where the sensor is attached directly to the body, the variations in the physical form of the person such as height, weight, and body mass index, in addition to age factors can result in erroneous sensor outputs. Compensating for these challenges has still to be addressed by the research community.

Furthermore, though the sensors are unobtrusive and do not invade the privacy of an individual, advances have still to be made for it to be acceptable by the elderly community. The state of the art technology requires the individual to explicitly wear the sensors. This is a primary hindrance to the adoption of the technology. Advances in the area of smart textiles and sensors that can be easily integrated into the everyday clothing of individuals will hopefully facilitate to overcome

some of challenges with respect to the ease of use. There are also concerns regarding the power consumption of the sensors to allow long term monitoring. Current systems require the batteries to be replaced at best every day. This still has to be improved to support long term usage of the sensors. In addition, there are certain contexts where wearable sensors cannot be used. For example, monitoring the movements of an individual, in a bathroom or while showering. These contexts do pose a restriction on the usage of the sensors in everyday life. Alternative modalities have to be used in these situations for monitoring the individuals.

### 1.2.2. Computational challenges

The challenges discussed here will focus on issues related to a computational framework for extracting meaningful movement patterns from wearable accelerometer streams. These movement patterns either define the set of simple activities or correspond to the activity gestures involved in complex activities. Recognizing these movement patterns poses a number of research challenges. The research challenges addressed in this dissertation are described in the following sub sections.

#### 1.2.2.1 Spatio-temporal variation in movement patterns

Movements involved in simple activities and gestures that are part of complex activities have varying spatio-temporal patterns. These patterns vary across individuals and across different activity contexts. For example, pouring water from a glass has a relatively different pattern when compared with pouring water from a bottle. Yet, both the movement patterns are associated with the same label. The framework should be able to account for these variations in the patterns. In addition, variation in the movement patterns also exist due type of activity. Movements in simple activities typically last for a longer duration, are repetitive, while gestures are of shorter duration and generally not repetitive. Thus techniques designed for recognizing one set of these movements might not work with the other set. Furthermore, the movement patterns associated with the same label potentially varies across individuals. Thus, achieving a good generalized performance across individuals is another challenge for the system.

#### 1.2.2.2 Sparse occurrence of gestures

As described earlier, complex activities are executed as a sequence of tasks each involving a specific gesture. Certain applications involving complex activities such as the monitoring of specific tasks and/or movements in a rehabilitation scenario or the classification of dietary intake gestures for an automated nutrition monitoring system require a reliable recognition on a more fine-grained level (at the level of tasks). Recognition of tasks at this level is particularly difficult, because the relevant gestures occur sporadically in between a large variety of other gestures. For example, while executing the tasks in the activity making a cup of tea, the individual might look at his/her or might just even scratch his/her head. This problem of trying to detect sporadic occurrence of relevant or in-vocabulary gestures in a continuous stream of sensor data corresponding to out-of-vocabulary gestures is typically called as gesture spotting. It is widely recognized as a particularly complex aspect of activity recognition and is still an open research challenge. This problem is of smaller relevance in the context of spotting simple activities. Here the challenge is to detect occurrence of out-of-vocabulary activities such as a fall or a transition from one activity to another in a continuous stream of in-vocabulary movements.

#### 1.2.2.3 Lack of real-world training data

Most of the computational frameworks for activity recognition learn in a supervised manner, thus requiring sufficient amounts of labeled training data. While it is relatively easy to collect training samples for simple activities such as sitting, standing or walking, it is difficult for complex activities such as making tea. The difficulty in obtaining training data is primarily because of the cumbersome process of collecting, cleaning and annotating the real-world data. Most often, humans cannot directly perceive these sensor streams for the purpose of annotation. Typically video or audio synchronized with the sensor streams is used to annotate the data. Furthermore, an execution of a complex activity results only in one or two samples of an activity gesture, which might not be sufficient to train algorithms for detecting the movement pattern in a continuous stream. Thus either devising alternate data capture modalities to provide sufficient number of representative gesture samples, or developing algorithms that learn the characteristics of a

particular gesture sample, using few samples of that label while relying on abundance of similar samples is an interesting problem.

#### 1.2.2.4 Sequential order of tasks in complex activities

Complex activities are typically carried out as a sequence of its constituent tasks. For instance, the activity-making brownie has nearly 33 different tasks such as pouring oil, cracking eggs, stirring the mix. This results in significant variations in the order in which these tasks are performed. For example, one individual might pour the brownie mix in the bowl of beaten eggs and then stir the ingredients, while another stirs while pouring in the brownie mix, and yet a third person stirs while holding a second utensil in the mix. This variation in the sequential order of task execution is an important challenge for systems that hope to predict the next task an individual is about to perform. While there is ambiguity in terms of the definition of these tasks, there is an additional amount of uncertainty introduced by low level processing that maps sensor streams to activity tasks. Thus modeling of the sequence of tasks in these uncertain scenarios is an interesting problem. Furthermore, individuals sometimes perform two or more activities concurrently. In these situations, tasks of both the activities overlap and the two activities might even share common tasks. This leads to an even more challenging situation for algorithms that predicting future tasks. The challenge here is to determine the structure of the activity.

## 1.3 Goal of the dissertation

There have been many studies on analyzing and recognizing simple activities using movement patterns gathered through wearable accelerometers as discussed in Section 3.2. Many of these studies have not explored the potential of discriminative classifiers in combination with the optimal placement of the sensors for achieving the best performance while recognizing simple activities. Furthermore, real-time considerations for recognizing simple activities have not been studied. The first objective of this dissertatoin is to study the effectiveness of discriminative classifiers for real-time recognition of simple activities. Data obtained from wearable accelerometers located on different body-parts from individuals performing simple activities such as sitting, standing, walking, running etc are used to conduct the study. State-of-the-art classifiers

such as Adaptive Boosting that builds upon the commonly used decision tables and trees, Support Vector Machines and models that output calibrated probabilities of classification such as regularized Logistic Regression are investigated. Experiments for understanding the discriminative power of different on-body sensor locations with respect to these activities are also conducted. Based on the outcome of this study a real-time simple activity recognition system has been developed.

While there has been a lot of work in analyzing movements from simple activities through accelerometer data, research on studying subtle, short-duration and non-repetitive hand movements is still under explored. There exists some similarity between these gestures and patterns in sign language and thus computer vision techniques such as hidden Markov models and its variants for sign language recognition discussed in [84] can be investigated for this purpose. The second objective of this dissertation is to study the different approaches for isolated gesture recognition using accelerometers and propose techniques that demonstrate good empirical performance. Activity gestures such as pour, stir, unscrew cap, lift to mouth and scoop is the focus of this study. Data gathered from accelerometers located on the upper-extremities of the body are used to conduct the experiments. The outcome of this study is a computational framework for isolated recognition of the gestures using discriminative techniques such as AdaBoost and SVM that can be extended for a continuous recognition scenario.

The third objective of the thesis is to develop novel algorithms for recognition of activity gestures from continuous accelerometer streams. These algorithms have to take into account the fact that activity gestures occur sporadically in a continuous stream of out-of-vocabulary gestures. Since it is difficult to obtain training samples that characterize the entire spectrum of out-of-vocabulary gestures, algorithms that make use of only valid gestures for learning to discriminate between valid and invalid samples are proposed. This thesis proposes adaptive threshold models based on Adaptive Boosting and One Class - SVM for detecting out-of-vocabulary samples. Gesture spotting networks that combine AdaBoost and SVM based isolated recognition techniques and the adaptive threshold models for detecting invalid gestures is proposed and evaluated. This study is

conducted in the context of recognizing activity gestures such as pour, stir, unscrew cap, lift to mouth and scoop involved in the complex activity of making a drink. The objective is not to propose a complex activity recognition system solely based on gestures, rather it is propose a framework for extracting the gestural information that can be integrated into a larger multi-sensory activity recognition system. Furthermore the proposed models for detecting out-of-vocabulary samples can be used in any context. The properties of the models are illustrated using other datasets.

To overcome the problem of lack of real-life training data, the framework proposes data collection through mock scenarios. During these alternative data capture sessions subjects enact the same movements with mock objects a number of times, thereby providing sufficient data samples for training. However data thus captured might not truly represent a real-world scenario. Thus a methodology for learning through a small number of samples from a real-life scenario combined with a number of samples from the mock scenario will be worthwhile. The proposed framework uses transfer learning techniques to accomplish this task. Transfer learning is an area of machine learning that deals with improving the performance of learning tasks in a target domain using knowledge gleaned from learning tasks belonging to source domains. This knowledge is in the form of actual data samples, a feature representation or even the parameters of the learning model. Since only a small number of samples from the real-world scenario are available for training, we experiment with instance transfer techniques. In particular a cost sensitive boosting based transfer learning technique is proposed for this purpose. This algorithm is evaluated with the data collected for the complex activity of making a drink. The effectiveness of the framework is tested with other bench marking datasets.

There are a number of approaches in the literature that map the sensor data streams to either tasks or directly to the complex activity. However predicting activity tasks even before it has occurred is still an under explored research area. Rule based reasoning and probabilistic models such as Markov chains and dynamic Bayesian networks are the commonly used approaches for activity task prediction [1][2][3]. Most of these approaches have been evaluated on the complex activity of

washing hands that consists of nearly 10 tasks. These approaches do not scale well with increasing number of tasks in a complex activity. This dissertation describes a novel hierarchical state space Markov chain model for predicting activity tasks. This algorithm is evaluated for predicting tasks involved in two complex activities namely; making brownie and making eggs, each consisting of 33 and 32 tasks respectively. The objective of this work is to automatically learn the structure of an activity from labels obtained through annotating actual executions of the activity by humans.

To summarize the goal of the thesis is to develop a **computational framework for movement based simple activity recognition and gesture spotting, by learning the spatio-temporal variations in the movements through a small number of true examples and a large of number of auxiliary samples**. In addition, it also proposes an activity task prediction model for complex activities. The computational framework is evaluated on activity and gesture data collected from accelerometers placed at different on-body locations

### 1.4 Contributions from the dissertation

This dissertation makes the following contributions:

- Real-time recognition of simple activities from continuous sensor streams using minimum number of accelerometers by a discriminative classifier.
- Demonstration of the feasibility of isolated recognition of subtle, short duration non-repetitive activity gestures through discriminative classifiers.
- Learning adaptive threshold models to detect out-of-vocabulary samples from only valid examples using Adaptive Boosting and multiple One Class - Support Vector Machines.
- Discriminative gesture spotting network for detecting relevant activity gestures occurring in continuous accelerometer data.
- A methodology for improving real-world performance of activity gesture recognition, by combining a cost sensitive boosting based transfer-learning framework with data samples obtained from a mock setting.
- A hierarchical state space Markov chain model for predicting activity tasks that an individual will perform based on the tasks that have already executed.

### 1.5 Outline of the dissertation

The dissertation is structured into 5 main chapters each focusing on one of the objectives discussed in the previous section. The outline of each chapter is as follows:

Chapter 2 introduces the paradigm of sensor based human activity recognition. It discusses approaches commonly found in the literature for activity recognition covering, wearable sensor, environmental sensor and video based approaches. It also provides a quick over view of different sensors that are used for activity recognition.

Chapter 3 addresses the first objective, described in the previous section, on simple activity recognition. It discusses the proposed approach for post classification label smoothing. It then elaborates on the experimental setup for testing the effectiveness of discriminative classifiers for real-time simple activity recognition. It also discusses the computational complexities of the proposed approach along with a discussion on accounting for human factors for a wearable activity recognition system.

Chapter 4 establishes the feasibility of the proposed accelerometer based activity gesture recognition. The gestures are recognized in isolation using k-NN, AdaBoost, SVM and HMM. The superiority of discriminative classifiers over generative and template-based approaches is highlighted through the proposed experiments. It introduces the test bench scenario for collecting activity gesture data upon which the work in the following two chapters is presented. A brief discussion on the feature space for accelerometer data for activity gesture recognition is also given.

Chapter 5 analyses the problem of activity gesture spotting in the presence of out-of-vocabulary samples. It presents the proposed approaches for detecting out-of-vocabulary samples using threshold models learnt from Adaptive Boosting and multiple one class – SVM. The efficiency of the proposed approaches over other techniques is illustrated on simulated benchmark datasets. It then discusses the gesture-spotting network that utilizes the proposed threshold models for spotting activity gestures in continuous accelerometer data streams.

Chapter 6 presents the proposed approach for learning from auxiliary data sources using the cost sensitive boosting for transfer-learning framework. It discusses the theoretical properties of the proposed framework and highlights its effectiveness on standard benchmark datasets. It also evaluates the proposed approach for performing real-life activity gesture recognition from models learnt from samples gathered from different settings.

Chapter 7 addresses the last objective on learning activity models from user trials for reliable task prediction. It presents the proposed hierarchical state space Markov chain model. It describes the data set used to validate the proposed models.

Chapter 8 concludes the work presented in this dissertation with a review of the findings. It discusses the limitations of the work and its relevance to other fields as well on giving suggestions for future work.

Chapter 2

# SENSORS FOR HUMAN ACTIVITY RECOGNITION

## 2.1 Introduction

Though the work presented in this dissertation focuses only on accelerometer based activity recognition, this chapter briefly reviews the different sensors used for activity recognition. Early work on activity recognition was primarily focused on using camera as the sensing modality. The recent past has seen the development of new kinds of sensors that capture a variety of information from the environment and body of individuals. Research on human activity recognition on data acquired from these sensors is one of the exciting new avenues for activity recognition. As discussed in the previous chapter the plethora of applications of human activity recognition is the motivating factor for this research.

This chapter is divided primarily divided into three sections, with each section focusing on wearable sensor, environmental sensor and video based activity recognition. Section 2.2 discusses some of the commonly used sensors for activity recognition. Section 2.3 provides an overview of wearable sensor based human activity recognition. Section 2.4 and 2.5 present a short description of environmental sensor and video based activity recognition. Section 2.6 provides the concluding remarks.

## 2.2 Sensors for human activity recognition

**Accelerometer:** Accelerometer is probably one of the most popular and ubiquitous sensors. It is has been made popular by devices and applications that change their functionality by changes in tilt or orientation such the iPhone, or capture gaming devices that capture movements of individuals to interact with the game such as the Wii mote. An accelerometer is a sensor that measures the linear acceleration that is induced by gravity or by the movement of the sensor. It is sensitive to shock, orientation, and vibrations. An accelerometer is designed around the principle that a mass in acceleration exerts force. If the exerted force and the mass of the body can be measured, acceleration can be derived based on principles of physics (force = mass*acceleration). There are different kinds of accelerometers based on its type of construction and its sensitivity

range. Most commonly found accelerometers are piezoelectric sensors that use the piezoelectric effect to measure the dynamic changes in the exerted forces. Some of the other categories of accelerometers based on its type of construction are MEMS, strain gauge and capacitive. The sensitivity of an accelerometer is defined in terms of the acceleration due to gravity (g). Low g accelerometers can sense up to +-12g accelerations, medium g accelerometers sense in the range of +-100g, while high g accelerometers offer >100g sensing capabilities. Acceleration values in human movements typically fall in the range of Low g accelerometers.

Advances in MEMS technology has enabled the development of miniaturized accelerometers that can be worn by individuals. These sensors transform into powerful wearable sensing units when they are coupled with small form factor wireless communication technologies. There are a number of commercially available wireless accelerometers. Experiments in this dissertation were conducted on data captured from two commercially available accelerometers – the WiTilt from Sparkfun [155] and ZStar from Freescale [156]. Other commercially available products are Mercury from Shimmer [157] and Wocket from MIT [158]. In contrast to an accelerometer, gyroscopes measure the angular velocity with respect to the inertial frame of reference. They are often used in conjunction with an accelerometer for activity recognition.

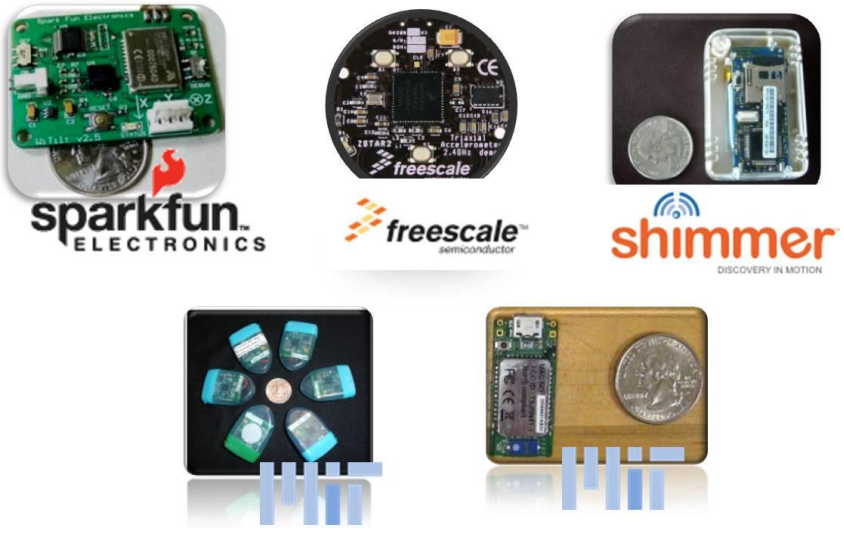



Figure 2: Some of the common wireless accelerometers available off-the-self. The imaeges have been adapted from the [155][156][157][162] and [158]

**RFID Technology:** Radio frequency identification technology has been around for many years. Popular uses of this technology have been access cards, toll road passes, supply chain and inventory monitoring and more recently pervasive shopping environments [159]. With the enforcement of the electronic product code (EPC), RFID becomes the most pervasive technology surrounding us. The most valuable property of this technology is its ability to provide a unique identification or serial number to every tag. There are two parts of this technology – the RFID reader and the RFID tag. Most applications use a passive tag that is powered wirelessly in the vicinity of an RFID reader. Thus it is important to note that while this technology detects the tags in the vicinity of a reader, it cannot accurately determine its location.



Figure 3: Objects of everyday use marked with RFID tags(indicated by the arrows) and the iBracelet RFID reader. The figure has been adapted from Stikic et al [132]

In the context of activity recognition, RFID technology is primarily used for object recognition [160]. While the problem of recognizing objects using RFID tags might sound easy, the design of an RFID reader that supports this process is nontrivial. The range and form factor of the RFID reader are the two obstacles for its adaptability for activity recognition [161]. There are a number of research issues that have to be addressed in this context. For example, what is the optimal location of the reader? Where should the tags be placed on an object? How many tags should be placed on an object? There has been some ongoing work addressing these questions. The object MITes developed by Tapia et al [162] is probably one of the earliest prototypes of wearable RFID reader for detecting interacting objects during activities. While it has a high wearability factor due to its shape as a wristband, the range of the reader is very low to be used for activity recognition. The iGlove and iBracelet are two wearable readers developed by Philipose et al [163] and Rea et al. [160] at Intel research Seattle for the object detection to facilitate activity recognition. These

sensors facilitate the reliable detection of objects that can be used to infer the high level activities. However, the sparseness of object tag information sensed by the reader and the difficulty-involving placement of tags on multiple objects makes it a difficult sensing technology for activity recognition.

**Motion Sensors:** Passive motion detectors that use infrared rays to sense heat changes in its field of view are commonly used to detect moving people and objects. A network of passive infrared sensors (PIR) embedded in an environment can be used to track the movement of people and identify their locations. This facilitates what is commonly called as location based activity recognition. For example, a system can guess reliably that an individual is cooking, if the PIR of sensor located above the stove gets triggered, or a that a person is using the bathroom if the sensor in the bathroom gets triggered. Thus PIR sensor provides an unobtrusive mechanism of sensing the location of an individual thereby recognizing the activity at a high level. The work done by Singla et al [164] and Kim et al [165] are examples of activity recognition using PIR sensors. While these sensors are easy to install and provide highly specific data to support activity recognition, they are difficult to use when there are multiple people in the environment. Furthermore these sensors do not provide the fine level of information that is required to track the progress of activities.

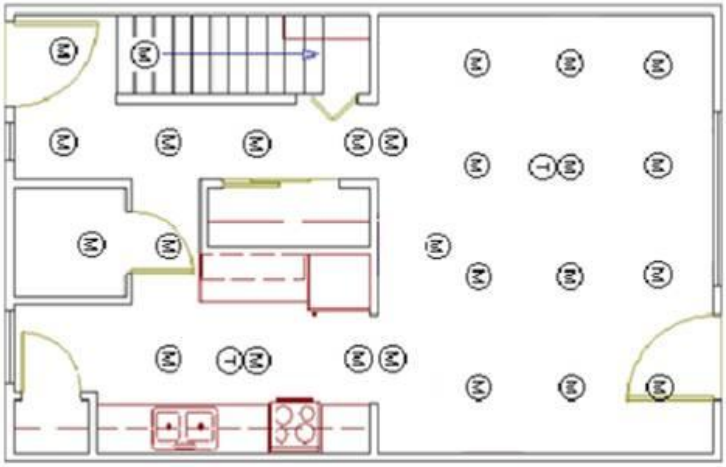


Figure 4: A visualization of the motion sensor activity for the 'wash hands' task as illustrated by Singla et al [164]. The circles marked as M represents the location of the motion sensor in the indoor environment

**Binary On/Off sensors:** Reed switches or binary on/off sensors are primarily electric switches that operate by a magnetic field. These sensors typically consist of two surfaces that turn on the sensor on contact or vice versa. In the activity recognition context, these sensors are often used for determining the state of an object. For example, placing the sensor on a door joint will facilitate detection of the door being closed or opened. They can be placed near the stove knobs to detect if a stove is on or off, or even to determine if a light switch is turned on. Akin to the PIR sensors, reed switches are inexpensive and easy to install. The sensors output binary streams that typically do not require any additional processing for extracting information related to activities. Reed switches are more versatile than PIR sensors, in the sense that they can be used in many different ways to obtain activity information. For example, when attached to a door, it can provide the location information of the individual, when attached to an object it can act as an object recognizer. However, there are certain limitations to the use of this technology for activity recognition. As discussed with PIR sensors, reed switches do not facilitate in tracking of activities of multiple residents in an environment. They also do not provide the fine level of activity information that is required to track the progress of an activity to its completion.



Figure 5: An illustration of common indoor locations and objects attached with reed switchess. The figure has been adapted from the work of Tapia et al [38]

**Pressure Sensors:** As the name suggests, these sensors measure the changes in the pressure exerted on them. They are primarily force sensing resistors that decreases its resistance with

increasing force. Multiple pressure sensors are typically used in conjunction for sensing pressure changes across a defined area. A well-known device that uses pressure sensing is the WiiFit balance board. In the context of activity recognition, pressure sensors are either integrated into the floors and carpets in an environment to determine the location of an individual as discussed by Orr and Abowd [166] and Richardson et al. [167], or can be integrated into objects of everyday use such a bed, and a chair to determine the state of the object. The resolution of the pressure-sensing units within a certain area determines the reliability of the mechanism for determining the location of an individual. Processing the data stream from these sensors to track the movement of an individual is considered to be a hard problem. With further advances in the hardware, this sensor holds promise for estimating the location of an individual.

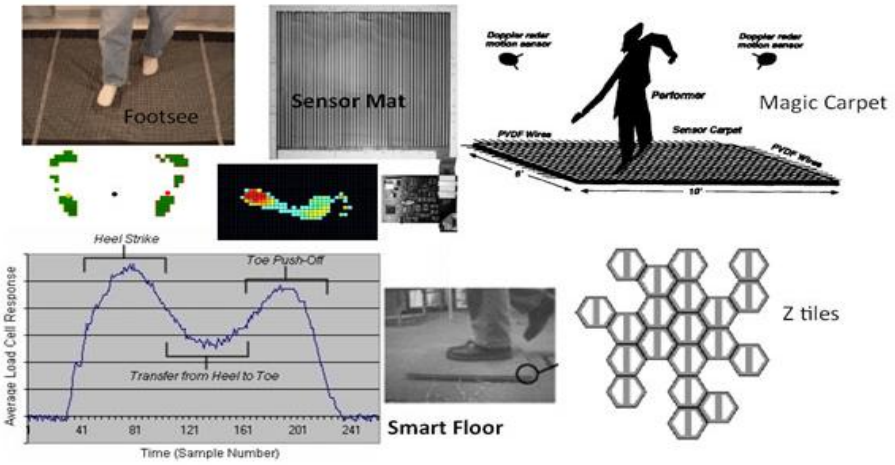



Figure 6: Pressure sensors used to measure the foot print of an individual. The images have been adapted from Orr and Abowd[166], Richardson et al [167] and Yin et al[210]

**GPS sensor:** GPS is another commonly used sensor for determining the location of an individual. With more and more integration of the sensor into every day mobile devices, this sensor is a rich source of unobtrusive and easily accessible information on an individual's activities. GPS sensor has been widely used for location based activity recognition Liao et al [168] demonstrate how GPS based location information can be used to recognize activities such as grocery shopping, meeting with friends, working at office etc.

**Eye tracking sensor:** This is a very recent technology that has been used for activity recognition. It was first demonstrated by Bulling et al [211]. It uses the principle of electro-occulography to

track eye movements. The hypothesis is that eye movements of an individual varies depending on the activity that is being performed. In their work Bulling et al [211] attempt at using this technology for recognizing activities such as working on a computer, reading, writing etc.

**Audio:** Systems that use sound as the sensing modality for activity recognition are not very common. With reliable acoustic models of an environment, audio streams support location estimation there by assisting in recognizing the activity of an individual. This however means the need for the individual to speak which can at times intrude into the privacy of the individual. On the other hand, if the activity by its inherent nature has distinctive acoustic profile, then it can be used for activity recognition. Stager et al [212] demonstrate the how audio streams can be used to recognize tasks involved in activities such as making coffee using microphones worn by an individual at the wrist. Ward et al [34] use a combination of both accelerometers and microphones to detect and recognize activities in a wood workshop. The primary difficulty in using audio for activity recognition is the noise from non-activity related tasks. It could be a person talking, or sounds from other sources. Filtering out sounds that correspond to noise from the audio signal is very challenging and computationally intensive.

In addition to the above described sensors, efforts have also been made on integrating multiple sensors into a single sensory device. Examples of this include the mobile sensing platform developed by Choudhury et al [213] that integrates microphone, light sensors tri-axial accelerometer, digital compass, barometer, and humidity sensor into a single sensing platform and the porcupine units built by Laerhoven et al [214] that combines accelerometers, tilt, light and temperature sensors.

Thus one can notice the wide variety of sensors that have been developed and used for activity recognition. Each sensor has its merits and drawbacks as discussed in the previous paragraphs. The focus of this dissertation is only on recognizing activities and gestures from accelerometer stream. We chose this sensor as it was easy to use, commercially available and gathers certain activity information that cannot be obtained from any other sensor. As discussed in chapter 1, the high level activity information can be easily detected using motion sensors or reed switches. In

addition to this, information about activity gesture is required to track complex activities and we hypothesize that this information can be gleaned from accelerometer data.

**2.3 Wearable sensor based human activity recognition**

The use of body-worn sensors for human activity recognition is relatively a new approach that has emerged as a key research focus in wearable computing. The most popular variant relies on motion sensors (accelerometers and gyroscopes) attached to relevant parts of the body, in particular hands and legs, to provide information about the movement of an individual. Other sensors that are commonly used include microphones, GPS and light sensors. These provide additional information about the environment or location of a user that might help in the task of activity recognition. By augmenting the user rather than the environment, the cost of augmentation is often reduced (both in terms of cost of the sensor and time to deploy). Using a small number of simple sensors reduces the complexity involved in signal processing and extraction of movement information when compared with vision based approaches, i.e., tasks such as background subtraction and motion estimation are eliminated, but the challenges involving discerning of activities still remains.

Non-visual on-body sensors, in particular, accelerometers have been employed for many years in the analysis of body posture and activity. The primary area of application has been ambulatory monitoring in a clinical setting (Bussmann et al. [4], collecting data to aid in the assessment of patients with a wide range of conditions such as Parkinson's disease or injury rehabilitation(Bonato et al. [5], Salarian et al. [6]) and cardiac rehabilitation [7]. Several works have been proposed to monitor the walking patterns of elderly people (Najafi et al[8] and Sekine et al. [9]).

Using two uniaxial accelerometers placed on the chest and thigh Veltink et al. [10] evaluated the feasibility of distinguishing postures such as standing, sitting and lying and also tried to separate body postures from dynamic activities such as walking and climbing stairs. Foerster et al. [11] also conduct a feasibility study for detecting postures and motion through accelerometry. Developments of this kind has motivated a number of researchers for carrying out ambulatory

activity recognition using wearable accelerometers as carried out by Aminian et al. [12] and more recently Wetzler et al. [13]. Mathie et al. [14] use a single accelerometer located on the waist of an individual to learn transition patterns in ambulation such as sit-to-stand, and stand-to-sit along with dynamic movements such as walking for the purpose of fall detection. Jafari et al. [15] use tri-axial accelerometers for measuring the kinematics of an individual to distinguish falls from other movements. Similar study was conducted by Al-ani et al. [16] using a single bi-axial accelerometer in an online mode. Karantonis et al. [17] discuss an implementation of their real-time human movement classifier using a single tri-axial accelerometer for ambulatory monitoring. Ibrahim et al. [18] propose to use accelerometers for classifying gait patterns involved in different styles of walking such as walking down a slope and climbing stairs. Identifying the age of a person by analyzing the walking patterns obtained through accelerometers is the focus of the work presented in Tseng et al. in [19]. Olguin [216] study the variation in the accuracy for detecting different ambulation across different accelerometer locations on the body.

The use of on-body sensors has been extended in understanding the context of a user as demonstrated in Mantyjarvi et al. [20], Randell et al. [21] and Van-Laerhoven [22]. Knight et al. [23] investigate the use of accelerometers in a wearable system for context aware computing and for measuring aspects of human performance that can be used for teaching and demonstrating skill acquisition in sports. Kern et al. [24] use multiple accelerometers for detecting activity context for wearable computing applications. These sensors have also been used to study activities and movements of individuals in working environments. Uiterwaal [25] conduct a feasibility study on long term monitoring of ambulatory activities in maintenance and messenger work. Bao eta al. [26] demonstrates the use of multiple accelerometers to recognize wide range full-body activities. Chambers et al. [27] investigated the recognition of certain Kung-Fu gestures by augmenting visual recordings with wrist worn accelerometer data.

Recognition of more intricate hand activities such as interaction with objects or gesticulation has been studied to a lesser extent. Antifakos et al. [28] studies hand movements while interacting with objects in a furniture assembly. Fang et al. [29] used special gloves embedded with

accelerometers for sensing different hand movements for sign language recognition. More recently, UWave [30] propose a system that uses an accelerometer located inside the Wii mote or an iphone, to train and recognize gestures defined by an individual. Zinnen et al. [31] propose to use accelerometers and gyroscopes for recognizing short duration and non-repetitive activities like turning on a car key, steering the wheel, opening the hood etc. Junker et al. in [32] propose a methodology for spotting activities such as turning a switch on, using cutlery from continuous inertial sensor streams.

Combination of multiple on-body sensors for recognition of different human activities is the focus of the work presented by Lester et al. in [33]. In addition to a tri-axial accelerometer, other sensors such as microphones, phototransistor, digital compass, barometer and digital thermometer were embedded into a multi-modal sensory board for sensing the activity and context of an individual. Ward et al. [34] use accelerometers and microphones to detect activities happening in a woodwork assembly scenario. Similar use of accelerometers and microphones for recognizing dietary activities such as eating, chewing and swallowing is proposed by Amft et al. in [35]. Traces of GPS sensor is used for determining the location of an individual, thereby leading to a high level inference about the activity of a person in the work proposed by Liao et al. in [36].

**2.4 Environmental sensor based human activity recognition**

Activity recognition from environmental sensors has basically stemmed from smart environment applications. Pervasive intelligent sensors that are embedded into the environment are used to understand the context and states such as activity and location of individuals inhabiting them. These sensors range from reed switches, motion detectors, RFID tags to pressure sensors. The various technologies differ from each other in terms of cost, intrusiveness, ease of installation and the type of data they output. Many research labs have developed test bed environments with these sensors for the purpose of activity recognition. A list of these smart environments along with the sensing approaches used is presented in Table 1.

Augmenting the environment with sensors such as switch-on, pressure sensors and movement detectors, which can tell the system about the object being used by the user or the location of the

user is another commonly found approach in the literature. Kasteren et al. [37] propose to use state-change sensors to recognize a number of activities through location and object usage. Tapia et al[38] demonstrate the use of motion sensors for location based activity recognition. Environment that has been tagged with RFID tags is the basis for activity recognition approach proposed by Patterson et al. [39]. This has the advantage of being computationally inexpensive, and potentially very reliable. However for extensive and detailed recognition, such a system would require augmenting a large part of the environment, which is a significant overhead and is expensive. In addition these sensors cannot provide fine movement information at the level of activity gestures, which is important for certain, applications

Table 1: Smart environments and commonly used sensors

| **Research labs** | **Environmental sensors** |
|---|---|
| MIT Place Lab [170], House_n project[169] | distributed temperature sensors, humidity, light, electric flow, gas flow, video camera, microphones and RFID tags |
| CASAS smart home project[171] | binary switch on sensors, motion sensors |
| Georgia tech aware home project [40] | pressure sensors, RFID tags, ultra-sonic sensors, vision sensors |
| CARE project [41] | binary switch on sensors located on certain objects, pressure mats and infra-red sensors |
| QoLT CMU [172] | video, audio, motion capture, IMUs, bluetooth accelerometers, RFID/iBracelet, eWatch, Bodymedia |

### 2.5 Video based human activity recognition

Computer vision based activity recognition has primarily focused on movement pattern recognition. An extensive survey of vision based approaches for activity/gesture recognition can be found in [40] [41]. Vision based approaches have typically focused on simple activities such as ambulation or well defined communicative gestures as in sign language. Recently there has been some effort in recognizing the activity gestures involved in complex activities such as washing hands as demonstrated by Hoey et al [1]. Messing et al. [42] propose an approach for recognizing activities in a kitchen using velocity profiles from videos. Computer vision based techniques for extracting motion and object information from images and videos are cumbersome and inaccurate and are susceptible to varying illumination conditions, poses changes and occlusion. More

importantly video based systems intrude upon the privacy of the individual and hence is not an appropriate technology for monitoring patients. However the techniques developed for activity/gesture recognition can be extended and adapted to suit the needs of other sensing modalities.

The challenge of modeling activities remains irrespective of the underlying sensing mechanism. The primary information required for modeling physical activities are objects and the associated movements. Past approaches for activity recognition, especially through videos face certain problems limiting their ability to ability to gather the necessary activity information. Wearable sensors offer a more reliable approach to `sense' the information necessary for modeling activities. However prior models for understanding movements through videos are still relevant in the context of accelerometer based movement recognition. The change in the sensing mechanism might result in the suitability of certain approaches over others. The challenges involved at the level of parsing tasks for inferring activities remain. In this context, past work on complex activity modeling through videos or any other approach is still relevant.

**2.6 Conclusion**

This chapter presented an overview of different wearable and environmental sensors and illustrated their role in capturing information related to human activities. It highlighted the merits and drawbacks of each sensor. It also summarized some of the activity recognition approaches using wearable, environmental and video sensors. Though the area of sensor based human activity recognition is still in its infancy, we believe that with advances in sensor technologies, recognizing different types of human physical activity will be a reality. For now, one has to extract the best with what is available, and this dissertation aims to do so with wearable accelerometers.

Chapter 3

SIMPLE ACTIVITY RECOGNITION

## 3.1 Introduction

Simple activity recognition deals with the process of inferring activities defined by movements from sensor streams. As described in Section 1.1, activities such as walking, running, sitting etc that are predominantly defined by movements belong to the category of simple activities. Wearable accelerometers provide one of the best platforms for sensing movement information about these activities. Simple activity recognition system finds uses in many applications in the health care domain. Systems that monitor the ambulation of an elderly individual to detect a fall or other abnormal movement patterns depend on the recognition of the simple activities. With the growing popularity of mobile devices embedded with inertial sensors, there has been a lot of focus on developing applications that can track the energy expenditure of an individual on simple physical activities. Recently there has also been a lot of focus on recognizing simple activities through inertial sensors for assisting the process of location estimation in GPS denied environments. In addition to health care applications, recognizing simple activities or other related movement patterns also find a lot of interest in gaming applications. An example of this is the

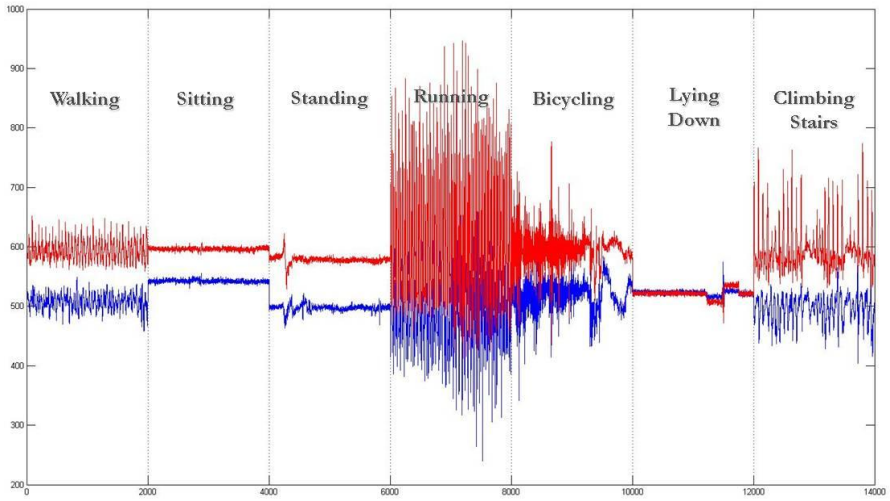



Figure 7: Biaxial accelerometer data for different simple activities from an accelerometer placed on the ankle

popular Nintendo Wii mote gaming systems.

Recognition of these activities is not a trivial problem. Figure 7 illustrates bi-axial accelerometer data for some simple activities. The movement in these activities can be either characterized by a static posture as in the case of sedentary activities such as sitting and standing or is defined by a repetitive movement pattern as in the case of walking or bicycling. The different ways in which the continuous data stream from accelerometers can be modeled has resulted in different recognition paradigms. The objective of the work presented in this chapter is to design a system that can recognize these activities with a minimal number of sensors in real-time. The rest of the chapter is divided as follows: Section 3.2 discusses the related work in the area of accelerometer based simple activity recognition. The proposed experimental framework for analyzing discriminative classifiers for simple activity recognition is presented in Section 3.3 It also discusses the post classification label smoothing framework that supports continuous classification. Section 3.4 presents the real-time simple activity recognition system implemented in this work. The computational complexity of the system is discussed in Section 3.5. Section 3.6 concludes the work discussed in this chapter.

**3.2 Related work**

Literature is abundant with techniques that use simple threshold based models such as decision trees and decision tables and generative classifiers such as HMM for recognizing simple activities. These techniques have been used in conjunction with a number of inertial sensors for simple activity recognition. Fixed length template matching using k-Nearest Neighbor (k-NN) has been quite popular among researchers for classifying simple activities using accelerometer data. Foerster et al. [11] employ k-NN for recognizing ambulatory movements using templates defined by the features extracted from fixed length window frames of accelerometer data placed at four different locations on the body. Jafari et al. [15] describe a method to detect the transitions between ambulation again using feature based templates in conjunction with k-NN. Maurer et al. [43] also investigate the performance of k-NN for recognizing ambulation. All of these

approaches first extract features from the acceleration movement pattern, which in-turn used as the templates for matching.

For its computational efficiency naive Bayes classifiers are commonly used for recognizing ambulatory movements as demonstrated by Kern et al [215], Bao et al. [26], Ravi et al [44] and Maurer et al [43]. The fixed length feature vectors extracted from movement data is used for learning the probability distribution of samples belonging to different classes. The probability distribution function is typically defined as a uni-modal Gaussian. Allen et al[45] and Ibrahim et al. [18] use Gaussian mixture models for recognizing the transitions between different ambulation and walking patterns. These approaches assume that movement patterns of a specific type adhere to a Gaussian distribution.

Lester et al [46] employ HMM as a post processing tool to smoothen out the recognition results of an AdaBoost classifier for detecting human activities using data from on-body sensors. Olguin and Pentland [216] explore HMM for recognizing ambulatory movements from inertial sensor data obtained from different on-body locations.

Ambulatory movements consist of static body postures such as sitting, standing and lying down along with dynamic movements involved in walking, running, climbing stairs etc. Techniques that determine thresholds on features such as variance or energy, offer a simple approach for classifying these different types of movements. Thus decision trees such as C4.5 are a natural choice for accelerometer based ambulatory movement recognition. Bao and Intille [26] experiment with decision tables and trees for classifying around 20 different activities from accelerometer data collected from five different locations on the body. Karantonis, et al [17] discuss a real-time human movement classifier using tri-axial accelerometer using binary decision trees for classifying walking patterns. Maurer et al. [43] use decision trees to recognize ambulatory movements using data from multiple sensors. Tseng and Cook [19] evaluate the performance of decision trees for determining the age of an individual based on movement patterns. Ravi et al. [44] experimented boosting and bagging with a number of base classifiers such as decision tables, decision trees, SVM etc for recognizing ambulatory movements from a single accelerometer and

gyroscope readings placed on the waist of an individual. Lester et al. [46] use AdaBoost to extract probabilities for classifying activities using data from multiple modalities. They also use AdaBoost as a feature analyzer to determine the importance of the different modalities for activity recognition. Artificial neural networks have also been used for activity and gesture recognition. Jafari et al. [15] explore MLP for learning the transitions between different ambulation. Yang et al [47] discuss a time-delay neural network based approach for recognizing motion patterns from trajectories extracted from image sequences. Mantyla et al [48] recognize static hand gestures from accelerometer data using self-organizing maps (SOM). Laerhoven et al. [22] also use SOM in conjunction with a Markov chain for differentiating ambulatory movements using data from a single inertial sensor placed on the thigh of the individual.

Support Vector Machines (SVM), a supervised classification or regression technique, that determines a hyper plane in the high dimensional space that maximizes the margin between samples of two classes is another approach commonly found in the literature for ambulatory activity recognition. Suutala et al. [49] use SVMs for classifying ambulatory movements using accelerometers located on the thigh, wrists and neck regions. Ravi et al. [44] use SVM along with boosted SVM for recognizing different activities using a single accelerometer placed on the waist of an individual. They observe that boosted SVM offers only a marginal improvement in classification performance over a regular SVM. Tseng and Cook [19] illustrate the effectiveness of SVM over other classification techniques such as MLP and decision trees for determining the age of an individual using movement patterns. While these are the different approaches adopted in the literature for simple activity recognition, the performance of discriminative classifiers for simple activity recognition has not been explored to a fuller extent. The work presented in this chapter is motivated by the need to evaluate powerful discriminative classifiers for simple activity recognition. The research objectives of this chapter can be summarized as follows:

- Do discriminative classifiers result in good performance?
- What is the minimal number of sensors required for simple activity recognition?

- How much data is required to detect a simple activity for achieving real-time recognition?

### 3.3 Experimental Framework

This section discusses the features extraction, classification and post classification label smoothing techniques that have been employed leading up to the real-time system for simple activity recognition.

#### 3.3.1 Description of the dataset

The data used for conducting experiments with the computational framework for simple activities is a subset of the data collected by Bao and Intille in [26]. The data was collected in two different ways - supervised approach (activity), where the subject is given explicit instructions about what action to perform, and a semi-naturalistic approach (obstacle), where the subject is given instructions to perform an activity of daily life that implicitly encodes the action patterns. The data corresponding to 10 random subjects from a pool of 20, for 7 lower body activities namely *walking, sitting, standing, running, bicycling, lying down* and *climbing stairs*, from accelerometers placed at *hip, dominant ankle, and non-dominant thigh*, for the two modes of data collection have been considered for the experiments performed in this work. The data was collected from bi-axial accelerometers that were strapped to the different body locations using Velcro. The accelerometers are sampled at approximately 76.25Hz. Figure 7 depicts typical samples that are obtained from the accelerometers. Details of the hardware that was used to capture the data and the methodology adopted for synchronization between multiple sensors can be found in Bao and Intille [26].

#### 3.3.2 Feature Extraction

The first step in the feature extraction process is to divide the acceleration stream in to frames. The acceleration stream are divided into frames of size 512 samples, with 256 overlapping samples between successive frames, as described by Ravi et al [44]. For each frame, the statistical features like mean, variance, correlation between all the axis of all the accelerometers, along with the spectral features like energy and entropy are computed. Figure 8 illustrates the projection of these features onto a three dimensional space derived through principal component analysis. For

activities that have a significant amount of motion like walking, running, etc the rate of change of acceleration is a characteristic property that distinguishes them. These variations are captured by computing statistical features like mean, variance and correlation between all the axes on the first order derivative of the acceleration data in addition to the features mentioned above.

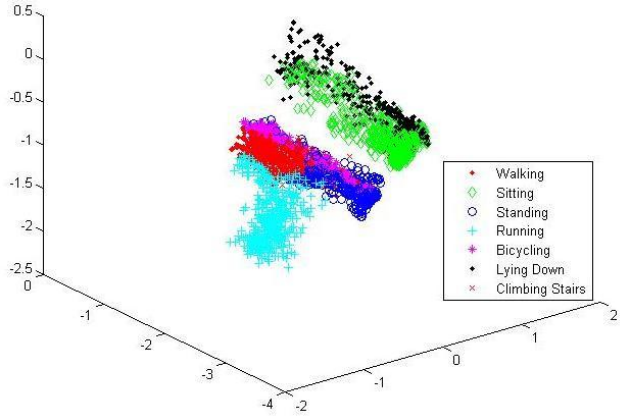



Figure 8: Projection of the simple activity data samples on to three dimensions obtained through PCA

The effect of different features on classification performance of AdaBoost was studied for determining the relevance of the features for discriminating simple activities. Separate AdaBoost classifiers were trained with the standard set of features, statistical features of the first derivative of the acceleration data and a combination of both. The accuracies for the three scenarios were 89.82%, 81.94% and 92.81% respectively. It is evident from Figure 9 that the standard features perform significantly better than the derivative features. However, there was a 3% increase in the accuracy when both the features were combined. Figure 9 presents the class-wise accuracy for the three scenarios. It can be noticed that the derivative features are able to distinguish accurately activities characterized by distinctive motion patterns like walking, running etc (1, 4, 5 and 7). The accuracies for these classes are on par with that of the standard features. This indicates that features extracted from the first derivative of acceleration data are able to capture the subtleties in the motion data.

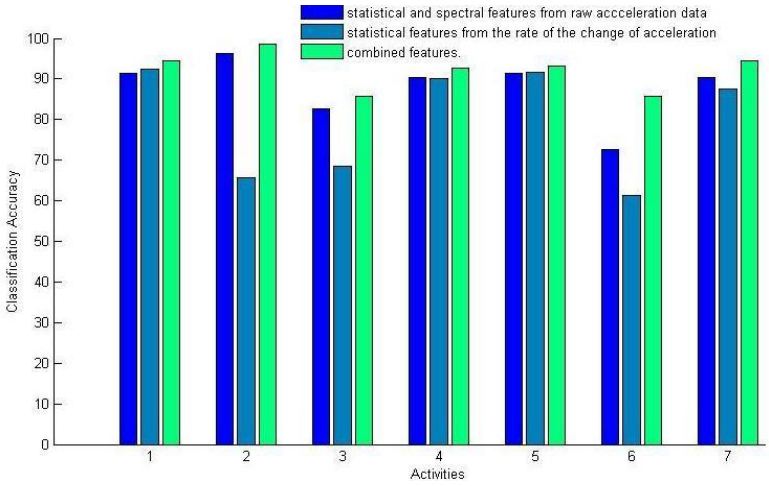



Figure 9: Class-wise acuracies using AdaBoost trained on the three features [1-Walking. 2-Sitting, 3 - Standing, 4-Running, 5-Bicycling, 6-Lying down, 7-Climbing Stairs]

### 3.3.3 Isolated Recognition

A natural extension of decision trees and tables to make them more powerful is through the use of the boosting framework. Thus the first choice for a discriminative classifier was AdaBoost. However, techniques such as AdaBoost do not necessarily model the probabilities of classification. They only provide the margin of classification that is typically approximated to a probability. However discriminative techniques such as logistic regression are designed to output calibrated probabilities of classification. Thus it makes sense to compare the performance of AdaBoost against logistic regression in the context of continuous activity recognition. In addition, SVM one of the most popular discriminative classifier was also used for conducting the comparative study.

The performance of AdaBoost, SVM and regularized logistic regression (RLogReg) on the features extracted from each frame was evaluated for developing the computational framework for isolated simple activity recognition. Binary classifiers were trained for each activity. The AdaBoost classification routine was implemented in matlab [173] based on the description of the algorithm in Duda and Hart [174]. The SVM and RLogReg implementation from the SVMLight package by Joachims [175] and Komarek's Logistic regression toolbox [176] was used

respectively. Given a test sample, the class that yielded maximum margin/probability, was considered as the predicted activity. This experiment was conducted on the dataset described by Bao et al [26]. Three different evaluation scenarios were considered for the analysis. For the subject independent scenario, activity data from nine subjects were considered as training samples and the obstacle data from the remaining one subject was the test data. The activity data of all the ten subjects were considered as the training set and the obstacle data from each of the subject formed the test set, for the subject adaptive scenario. The activity and obstacle data from only a single subject formed the training and test set for the subject dependent evaluation.

Table 2: Subject independent, adaptive and dependent accuracies for the different discriminative classifiers

| Classifiers | Subject Independent | Subject Adaptive | Subject Dependent |
|---|---|---|---|
| **AdaBoost** | 92.81 | 93.96 | 47.88 |
| **RLogReg** | 86.55 | 88.14 | 74.56 |
| **Linear SVM** | 82.28 | 83.60 | 72.64 |

The results summarized Table 2 show that AdaBoost performed best in both subject independent and adaptive scenario, while RLogReg had the highest accuracy in subject dependent case. The 90% reduction in the size of the training data for the subject dependent scenario was the cause for the poor performance of AdaBoost. We did not experiment with kernels for SVM due to the high computational costs associated with them. The confusion matrix for classification aggregated over the 10 subjects for subject independent scenario using AdaBoost is presented in Table 3. The misclassification of walking samples as climbing stairs and vice versa, suggests that the motion patterns involved in them are similar. There were also misclassifications occurring between activities that do not involve any quantitative motion in them, probably indicating that necessity of data from other parts of the body. A probable reason for this is that, data from accelerometers placed only in the lower parts of the body have been considered. Adding data from other accelerometers might improve the classification of these activities. Misclassification of bicycling samples as standing seems to be a very strange anomaly. On further analysis of the data, it was observed that these samples did not show any representative motion pattern associated with bicycling, thus can be possibly considered as outliers.

Table 3: The aggregate confusion matrix obtained from subject independent 10 folds cross validation using AdaBoost trained on combined features

| Activity | Walking | Sitting | Standing | Running | Bicycling | Lying Down | Climbing Stairs |
|---|---|---|---|---|---|---|---|
| **Walking** | 840 | 0 | 18 | 2 | 9 | 0 | 20 |
| **Sitting** | 0 | 296 | 0 | 0 | 0 | 4 | 0 |
| **Standing** | 0 | 13 | 128 | 0 | 8 | 0 | 1 |
| **Running** | 10 | 0 | 6 | 458 | 11 | 0 | 9 |
| **Bicycling** | 0 | 0 | 32 | 0 | 443 | 0 | 0 |
| **Lying Down** | 0 | 57 | 0 | 0 | 0 | 343 | 0 |
| **Climbing Stairs** | 13 | 0 | 4 | 1 | 1 | 0 | 323 |

### 3.3.4 Post classification label smoothing for continuous recognition

Human activity is a continuous process and though these discriminative techniques are effective in classifying an individual frame, they do not consider temporal continuity for classification. Consider the Figure 11, where the red line corresponds to the ground truth and the green line corresponds to the output of the static classification algorithm. It can be observed that a number of samples that have been misclassified were actually in the midst of a continuous stream of correctly classified samples. This means that the strict condition of independent samples can be relaxed in this scenario to correct some of the inaccuracies in classification. Lester et al [46]and Suttala et al [49] propose to use a Hidden Markov Model trained on the probability of classification obtained through AdaBoost and SVM respectively to correct these types of errors. In this work a classification framework that incorporates this temporal continuity of human activity is proposed does not require re-computation of the feature vector nor requires any additional training, thus remains computationally inexpensive. It relies on the similarity of successive samples in the continuous stream to combine the probabilities of classification.

Formally, classification margin $m_c(f_t)$, for a frame $f_t$, belonging to a class $c$ derived either in AdaBoost or SVM reflects the confidence of prediction. This margin can be used by the classifier to output the probability, $p_c(f_t)$, of the frame belonging to class $c$. A method to compute the probability directly is to fit a sigmoid function to the output of AdaBoost or SVM as described in the following equation.

$$p_c(f_t) = \frac{e^{\phi m_{c(f_t)}}}{1 + e^{\phi m_{c(f_t)}}}$$

The probability values computed for the frame $f_i$ at time instant $i$ can aid in classifying successive temporally close frames. For a frame $f_t$, let the frames that influence its classification be $f_i$, where $i = t - \Delta t, \cdots t$. We weight the probability $P_c(f_i)$, for the frame at $i$ belonging to class $c$, by two factors - a function of $i$ (temporal distance between the frames) denoted by $g(i)$ and a function of the similarity between the current frame and the past frame, measured as the Euclidean distance between them denoted by $h(t-i,t)$. Thus the final probability $P_c(f_t)$ for the frame at $t$, is given by the following equation where the denominator acts as a normalizing factor.

$$P_c(f_t) = \frac{p_c(f_t) + \sum_{i=1}^{\Delta t} g(i) * h(t-i,t) * P_c(f_{t-i})}{\sum_{i=1}^{\Delta t} g(i) * h(t-i,t) + 1}, c = 1 \ldots 7$$

For the experiments conducted in this work, the function $g(i)$ was treated as a Gaussian. This was done to ensure that frames that are farther away in time have minimal influence on each other. The function $h(t-i,t)$ was represented as $h(t-i,t) = e^{\{-\alpha\, d(f_{t-i}, f_t)\}}$, where $d(.)$ corresponds to the Euclidean distance between the feature vector describing the frames. This assumes that if adjacent frames are similar, then they should belong to the same class. This framework is illustrated in Figure 10. Though in this work, experiments were conducted with only AdaBoost and RLogReg, as a broader impact, the proposed framework can be adopted to work with any classifier.

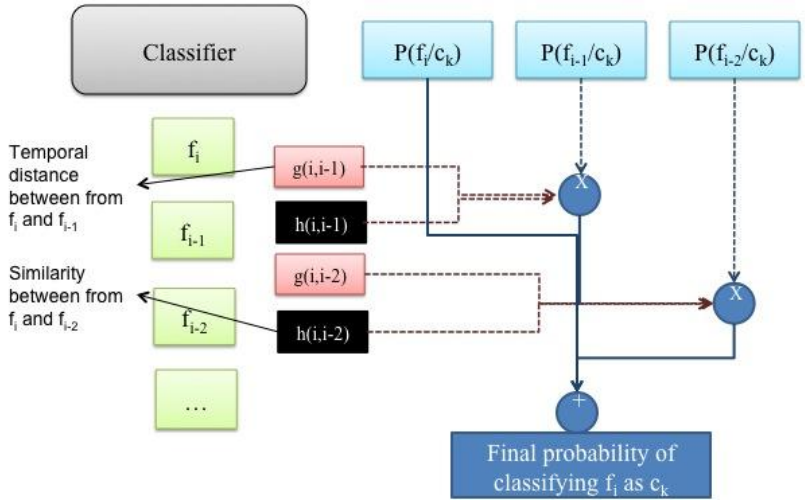



Figure 10: Illustration of the post classification label-smoothing framework

The continuous acceleration stream from the obstacle dataset as a sequence of overlapping frames was considered for evaluating the proposed methodology. The number of past frames considered for classifying the current frames was varied. The optimal performance was achieved when three past frames were considered for classifying the current frame. AdaBoost and RLogReg classification routines were considered for the evaluation of the framework. While adding temporal information to static AdaBoost resulted in an average 10 fold cross validation accuracy of 95.35%, RLogReg resulted in 89.63%. For both the algorithms, an improvement of about 2.5 - 3% was observed. Figure 11 illustrates the effect of adding the temporal component to the static AdaBoost classifier for one subject. There is a reduction in the number of misclassifications by the blue line that corresponds to the classification result of AdaBoost with temporal component added to it.

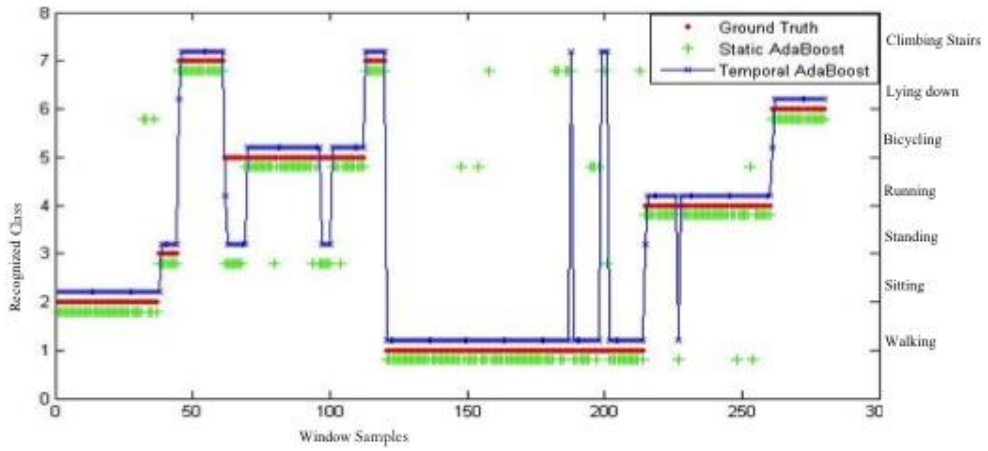



Figure 11: Output of static and temporal AdaBoost compared against the ground truth

### 3.3.5 Human factors assisting in continuous classification

The human body can be viewed as a kinematic system with well-defined degrees of freedom of movement. This kinematic system defines the transitions that can occur between different activities. For example, an individual cannot shift directly from a sitting state to a running state. The human body goes from a sitting state to a standing state and then to a running state. This sequence of states is actually reflected in the data captured through wearable sensors. A mechanism that can capture this information can be used to assist in the continuous classification

process. The action grammar work of Ivanov and Bobick is one way to capture this information. Ivanov and Bobick [177] in their seminal work on action grammars combine the syntactic and statistical schools of pattern recognition for the purpose of activity recognition. Statistical knowledge about the components or low-level primitives is combined with the structural knowledge expressed in the form of grammar. The syntactic knowledge acts as a constraint to the recognition of individual components as well as directs the process of recognition of the activity as a whole. The methodology proposed by them is useful in the context of recognizing the high level activity.

In the context of simple activity recognition, this type of a grammatical framework can be used to validate the transitions between different activities. For example, a walking state cannot be reached from a sitting state without passing through the standing state. Thus the structure of the transition relationship between these activities can be modeled using action grammars. Adding these types of constraints has the potential to improve the performance in a continuous scenario. The current framework does not explicitly model the transitions between the activities and thus a context free grammar (CFG) based approach may aid the recognition process..

The kinematic constraints in the form of a CFG could also be integrated at a lower level for assisting in activity recognition. At the lower level these activities can be broken down to smaller units. A grammar is defined for the sequencing of these units to recognize the activities. This is a very common approach in video based action or gesture recognition. But that seems to be an overkill considering that the current simpler framework reliable detects the actions. However, if the application necessitates recognition of finer details of walking and running such as stride length; CFG's will be useful in parsing the low level primitives.

While Ivanov and Bobick's work describe how to integrate an action grammar into a statistical recognition framework, they do not discuss how to build the grammar. It is assumed that the grammar exists or can be derived from the domain. In scenarios where the action is defined by a small group of gestures (such as the examples described in their work drawing a square), these models can be constructed manually. However, in the context of complex activities, it is difficult

to manually define the grammar for every activity. The activity task prediction part of the work that is discussed in chapter 7 is primarily focused on this aspect of trying to determine the structure of the complex activity on data collected from humans executing the activity. The structure thus obtained can be converted to productions and rules for developing the action grammar framework for reliable activity recognition. Currently the only approaches for determining the structure of a complex activity from user data is through a Markov chain model as demonstrated by Singla et al [164] or through mining web pages for activity recipes as discussed by Perkowitz [178].

To validate the transitions between the different simple activities, a CFG was manually constructed based on the human constraints in the performance of these activities. The five simple activities that were considered for this study are sitting, standing, walking, running and lying down. A CFG that represented the relationship between the different activities was constructed. Probability values were assigned to each of the possible transition in the CFG. This probabilistic CFG can be represented in terms of a first order Markov chain transition matrix. It was assumed that the tendency of a subject to remain in an activity was higher than moving to another activity. This is reflected in high probability values for the self transition elements in the transition matrix. The probability values for all the other transitions are equally distributed. Table 4 describes the probability values of this transition matrix

Table 4: Transition matrix for the CFG defined for simple activities

| **Activity** | **Walk** | **Stand** | **Sit** | **Run** | **Lie Down** |
|---|---|---|---|---|---|
| **Walk** | 0.6 | 0.2 | 0 | 0.2 | 0 |
| **Stand** | 0.3 | 0.3 | 0.2 | 0.2 | 0 |
| **Sit** | 0 | 0.2 | 0.6 | 0 | 0.2 |
| **Run** | 0.2 | 0.2 | 0 | 0.6 | 0 |
| **Lie down** | 0 | 0 | 0.2 | 0 | 0.8 |

It is interesting to note from the transition matrix that the ‘Stand’ is the unstable activity amongst the rest as it can easily transit to other activities. The self-transition probability of this activity is lowest. This is in accordance with what one typically observes in the real world. This matrix defines finite state automata that can be used to smoothen out the labels post classification. The

data from the previous study was used to study the properties of such a finite state automata. Samples corresponding to cycling and climbing stairs were removed from the continuous stream of sensor data. The AdaBoost classifier was used for recognizing the individual samples. The classification probabilities for every time step, derived through boosting were the input to the finite state automata. The probability values were propagated through the automata and normalized for every time step. The state resulting with the maximum normalized probability was considered as the activity label for that particular time step. The average error rates obtained from testing on data collected from 4 subjects were 0.08 and 0.09 with and without the finite state automata model. It can be seen that adding the CFG framework improves the accuracy only by a marginal amount. The static AdaBoost model itself resulted in very high recognition rates, thus adding the grammatical framework does not influence the outcome significantly.

### 3.4 Real-time Classification

Encouraged by the results obtained in the previous section, a real time system for detecting and recognizing lower body simple activities (walking, sitting, standing, running and lying down) using streaming data from tri-axial accelerometers was designed. The first step is collecting data from the two accelerometers and passing this information through the preprocessing and segmentation stage. In this second stage, spurious noise in the data is removed and the continuous stream is broken down into more manageable segments. Each data segment then passes through the feature extraction step, where salient features are extracted to characterize the properties of the raw data. These fixed-length feature vectors are then sent through the classification stage, where a trained AdaBoost classifier is used to identify the activity corresponding to the sample and the probability of the classification is computed. The next sections describe each of the steps of the activity recognition system framework in more detail.



Figure 12: Accelerometer placement for the real-time simple activity recognition system

The unique contribution of the research presented here is the use of data gathered from a limited number of accelerometers. While the performance is marginally superior to the best of the previous results, this approach is distinct in using only accelerometer data. It improves on standard feature extraction frameworks by using a boosted classifier for recognition, resulting in a system that has a very high accuracy for real-time activity recognition.

### 3.4.1 Data capture

The system relies on off-the-shelf accelerometers (WiTilt v2.5 employing a Freescale MMA7260Q triple axis accelerometer with class 1 bluetooth) connected to a computer using a wireless Bluetooth serial port. These accelerometers are sturdy and have only minimal data loss over long periods of continuous sampling (<1% for 600 seconds). Data from 3 subjects (2 males and 1 female) was collected. The accelerometers were placed on the right ankle and on left thigh, with the x-axis facing perpendicular to the ground, to maintain consistency across the subjects as illustrated in Figure 12. Each subject was asked to perform five different activities (walking, sitting, standing, running, and lying down) for duration of 1 minute. The accelerometers were sampled at the rate of 100Hz. The 60-second trial of a subject was then broken down to smaller chunks each consisting of 100 acceleration samples. This gives a total of 180 samples per activity

combining the data from all the subjects. Each sample corresponds to the acceleration data from each of the three axes of the two accelerometers.

### 3.4.2 Data Processing and Feature Extraction

The second step is to break the continuous data stream into equal-length segments of information, which is an approach proposed by Bao and Intille [26]. In this framework, each segment consists of 100 samples and successive segments have an overlap of 50 samples. The sampling rate of the accelerometers is 100hz, so each segment corresponds to 1 second from the data stream. This time interval proved to be sufficient for analyzing the activities we were trying to recognize (walking, sitting, standing, running, and lying down). In the next step statistical and spectral features described in our prior work was extracted. These features are computationally inexpensive and characterize most of the distinguishing features for separating the activities considered for the real-time implementation (walking, sitting, standing, running, and lying down).

### 3.4.3 Classification

In order to learn the parameters of the classifier, a threefold cross validation technique was used. For each fold, the data corresponding to one subject was used for the testing the model learned by training the AdaBoost on the remaining data. The classifier stabilized after 250 iterations. The average subject independent accuracy obtained in this fashion was around 95.2%. We also experimented with the data from each accelerometer: when data from only the ankle accelerometer was used, the accuracy of the classifier dropped to 62%; when data from only the thigh accelerometer was used, the accuracy was 83%. Analysis of the confusion matrix obtained after the classification revealed that data from the ankle accelerometer was insufficient for classifying two activities (sitting and standing), while data from the thigh accelerometer was in- sufficient for classifying four activities (sitting, lying down, walking, and running). Clearly, these experiments demonstrate the necessity of multiple accelerometers for recognizing the activities selected for our system.

Finally, the data from all three subjects were used to train the classifier, with the resulting model incorporated into the continuous activity recognition framework. This classifier has been presented

in a number of live demonstrations using volunteers not in the training database illustrating its robustness and dependability. Figure 14 illustrates the real-time classification for an interesting example. The data consists of a total of approximately 8000 samples (corresponding to 80 seconds), and it can be seen that the

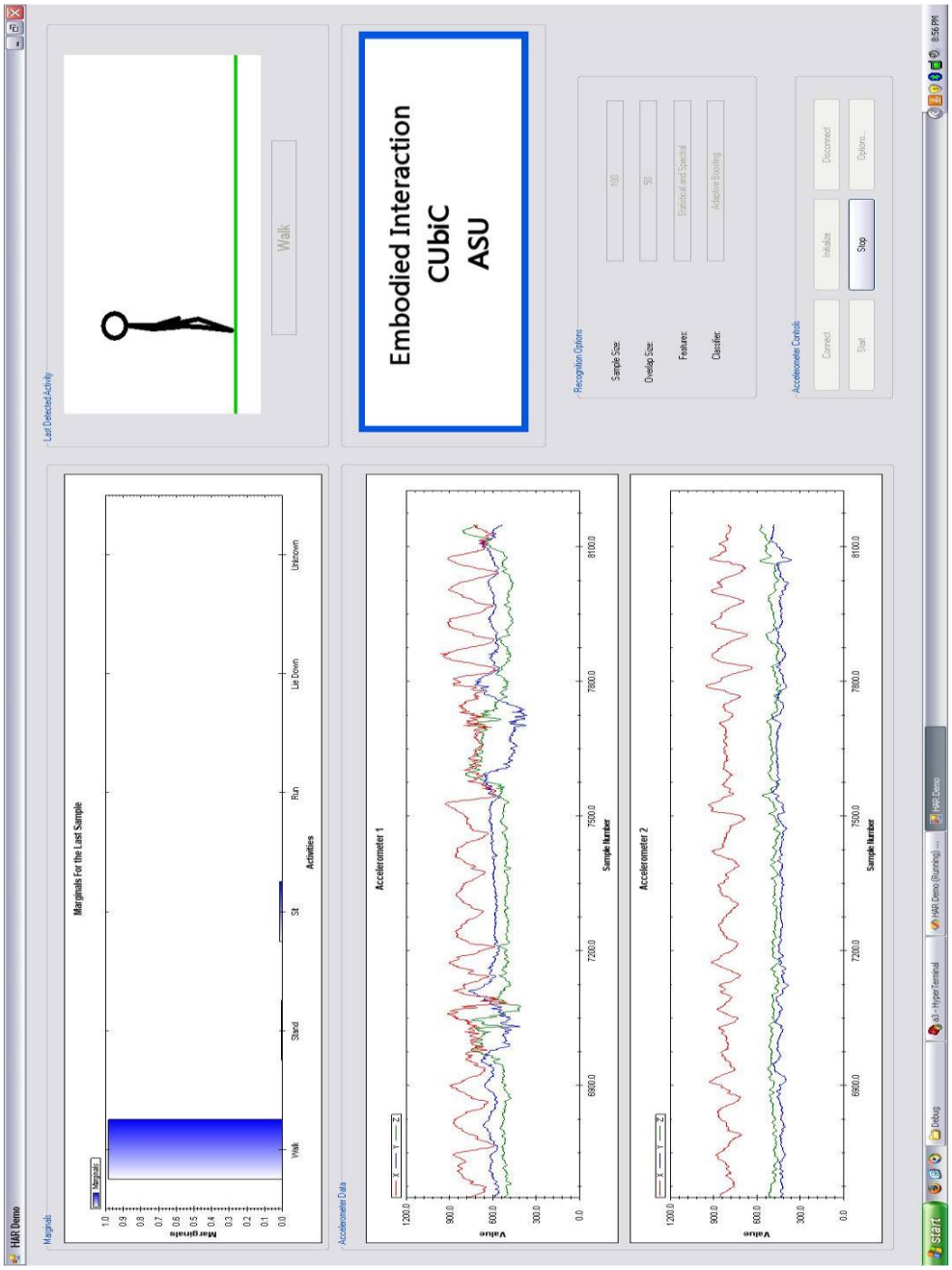



Figure 13: Illustration of the software developed for real-time recognition of simple activities

classification is only 84.4% accurate once the activity stabilizes. This particular subject walked faster than the three training subjects in the database and illustrates the most likely type of misclassification (confusion between walking and running) that can occur using the current classifier. If walking is excluded as a possible ground truth, the accuracy of the system returns to our typical 98%. It was also noticed that the probability of classifying the samples as walking was low. However the threshold that was used to remove transitions and other arbitrary activities was not sufficient to detect these anomalies. This clearly indicates the need for an adaptive threshold model for detecting out-of-vocabulary samples.

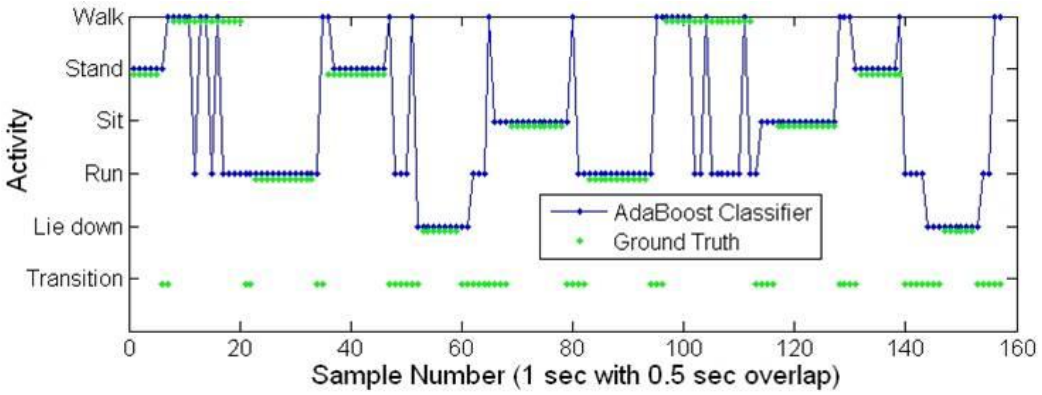



Figure 14:Real-time continuous example comparison of AdaBoost with the ground truth

**3.5 Computational Complexity**

In this section, the theoretical analysis of the computational complexity of the simple activity recognition system is first presented. The activity recognition system consists of two parts - feature extraction and classification. Thus the overall computational complexity $CC$ of the system can be broken down as follows:

$$CC = CC_{FE} + CC_{CL},$$

Where $CC_{FE}$ corresponds to the complexity of feature extraction and $CC_{CL}$ is the complexity associated with the classification step. The current real-time system consists of two accelerometers each sampled at 100Hz. Continuous data from these two sensors were classified using sliding window protocol. The window was of length 100 samples (1s) with an overlap of 50 samples between successive windows. Statistical and spectral features consisting of mean, variance,

correlation energy and entropy were extracted from each window. Multiple AdaBoost classifiers trained on samples for each of the activity was used to classify the sample.

Let us now look at the order of complexity of the feature extraction step. Table 5 lists the set of features considered in this work, along with their computational complexity. Thus every sliding window requires a total of $12\ O(N) + \ 15\ O(N^2) + \ 12\ O(N \log N)$ computations for extracting the features. $N$ refers to the number of accelerometer samples in each window. In the proposed system this value is 100. The total number of computations for extracting the feature vector is approximately $1.6\ e06$.

Table 5: Computational complexity of the features extracted, where $N$ is the number of data samples

| Feature | Order | Dimensions |
|---|---|---|
| **Mean** | $O(N)$ | 6 |
| **Variance** | $O(N)$ | 6 |
| **Correlation** | $O(N^2)$ | 15 |
| **Spectral Energy** | $O(N \log N)$ | 6 |
| **Spectral Entropy** | $O(N \log N)$ | 6 |

Focusing on the classification step, the AdaBoost classifier was trained using decision stumps. A decision stump is a primarily a threshold based classifier and has a computational complexity of $O(1)$. The maximum number of boosting iterations during training was set to 100. It was observed that some of the classifiers had learnt less than 100 weak hypotheses. However for computing the worst case scenario, assume that each of the classifier has 100 hypotheses. Thus the computational complexity of classifying a frame is 500 ($100 * number\ of\ classes\ = \ 5$).

Therefore the total computational cost of extracting and classifying a single window and can be derived as

$$CC(N) \ = \ 12\ O(N) \ + \ 15\ O(N\text{^}2) \ + \ 12\ O(N\ log\ N) + 500\ O(1)$$

and in this particular example, for $N = 100$, this value turns out to be $\approx 1.6e06$ computations. Assuming that the device needs $D$ units of battery discharge to complete one computation, the discharge lost by the battery for classifying one window is $1.6\ e06/D$. If the complete discharge of the battery happens after T discharges, the duration of the battery in this system will be $\frac{T*D}{1.6e\ 06}$s.

Boyd et al, [179] conduct an extensive evaluation on the effect of different parameters of an accelerometer based activity recognition system by identifying Pareto-optimal points in the operational parameter design space. For a practical solution to the above question, an experiment similar to the one proposed by Boyd et al. was designed to study the effect of variation in the key parameters of the system such as sampling rate, window size and features extracted on the performance of the system. The computational complexity of the system was measured in terms of normalized computation time for classifying one window. The performance of the system was measured in terms of its accuracy. Simple activity data from three subjects were used in this experiment. Figure 15 depicts the impact of these parameter variations on the accuracy of the system. The accuracy was the average value obtained through three rounds of subject independent evaluation. The sampling rates for the experiment were chosen to be 100Hz, 50Hz, 20Hz and

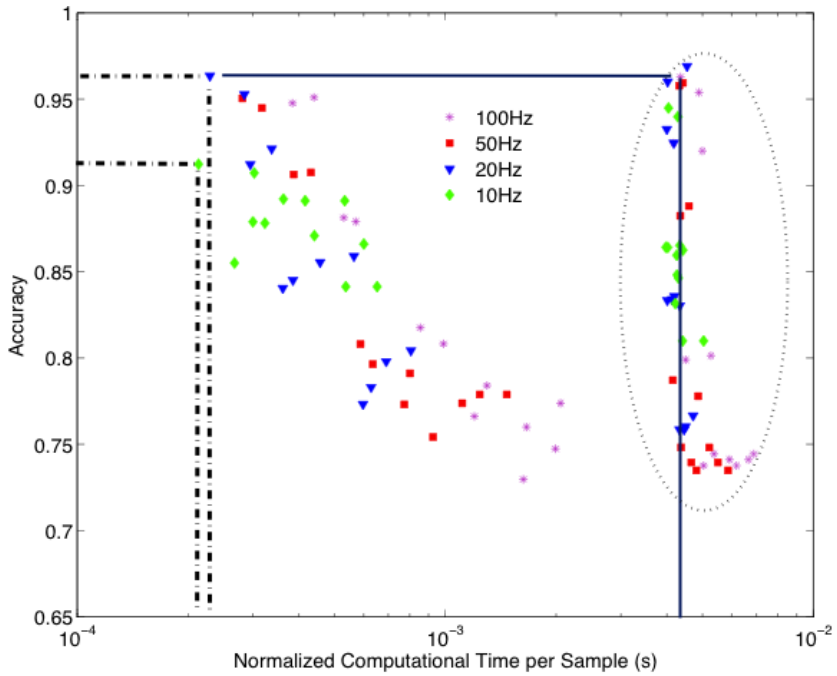



Figure 15: Comparing the normalized computational time for different parameters of the system against the performance measured as accuracy

10Hz. The window size values were 1, 2, 4, 6, 8 and 10 seconds. The features set considered in this system was divided into statistical (mean, variance and correlation) + spectral (energy and entropy), statistical only, mean + spectral, mean+ variance. Each of these combinations of

operational parameters is represented as a point in the plot according to its normalized computational time and accuracy.

The points inside the ellipse correspond to the scenarios where correlation between the axes was used as a feature for classification. The distinct separation between these points and other points with respect to the computational time clearly supports the fact that correlation is an expensive feature. At the same time, it is clear that parameters without the correlation feature did not result in change in the performance of the system. Thus the contribution of correlation towards the performance of the system is significantly less compared to other features.

It is very interesting to note that the system is able to support low power consumption without compromising on the performance as indicated by the green diamond and the blue triangle in the top left corner Figure 15. The current parameters (indicated by the solid lines) are suboptimal in terms of the computational time. However, this can be rectified by using the operational parameters described by the points on the top left part of the Figure 15 without compromising on the accuracy. Assuming that one unit of battery discharge takes place during the time required for the computation of a sample, the optimal parameters for the system would make the battery last 5% longer than the current operational parameters.

### 3.6 Conclusion

This chapter evaluates the effectiveness of different discriminative classifiers for simple activity recognition from low-resolution accelerometer data. Table 6 summarizes the results obtained by different researchers for continuous recognition for the similar set of activities. The last two rows of this table are the accuracies that were obtained using the methodology proposed, in this chapter, on the two data sets collected for simple activity recognition. The results clearly illustrate the superior performance of discriminative classifiers such as AdaBoost and SVM over other classifiers. The proposed methodology achieves better performance in-terms of the accuracy, with a minimum number of accelerometers.

This chapter also presents experiments conducted to study the effectiveness of complementing the standard features with statistical features extracted from the first order derivative of the

accelerometer data for classification. It was observed that adding these new features improves the classification performance only for certain activities. A technique for post classification label smoothing based on the temporal continuity of the data was also discussed. This proposed technique for adding temporal continuity to the classification yielded promising results with about 2.5-3% improvement in accuracy. It also discusses a methodology for taking into consideration human factors for improving the classification performance in the continuous scenario.

Based on these results, a real-time simple activity recognition system was designed and implemented. The proposed real-time system is able to accurately recognize the simple activities in real-time achieving a performance superior to existing approaches in the literature. The complexity of the system was discussed both in terms of the computational complexity and normalized time for computation for classifying a single sample.

Table 6: Summary of past work on simple activity recognition using accelerometers presented in a chronological order. The recognition accuracies correspond to the subject independent classification scenario

| **Reference** | **# of Sensors** | **Placement of Sensors** | **Classifiers** | **Accuracy** |
|---|---|---|---|---|
| Foerster et al [11] | 4 | Sternum, Wrist, Thigh, Lower Leg | k-NN | 95.8 |
| Mantyjarvi et al [20] | 2 | Hip | | 83-90 |
| Bao et al [26] | 5 | Wrist, upper arm, waist, thigh, ankle | Decision Tables, trees, Naïve Bayes | 84.26 |
| Ravi et al [44] | 1 | Waist | Boosted and Bagged decision tables, decision trees, k-NN,SVM, Naïve Bayes | 73.33 |
| Allen et al [45] | 1 | Waist | GMM | 76.6 |
| Lester et al [46] | 3 | Shoulder, Waist, Wrist | HMM | 90 |
| Maurer et al [43] | 6 | Wrist, Belt, Necklace, Shirt and Pant pockets, bag | Decision Trees, k-NN, Naïve Bayes, Bayesnet | 89.5 |
| Ibrahim et al [18] | 1 | Hip | GMM | 88.76 |
| Suutala et al [50] | 4 | Thigh, Left and Right Wrist, | SVM, HMM | 93.06 |

| | | | | |
|---|---|---|---|---|
| | | Neck | | |
| Muscillo et al [51] | 1 | Ankle | DTW based k-NN | 85 |
| Jafari et al [15] | 1 | Chest | Neural networks, k-NN | 84 |
| Krishnan et al [52] | 5 | Wrist, upper arm, waist, thigh, ankle | AdaBoost, Logistic Regression, SVM | 95.35 |
| Krishnan et al [53] | 2 | Ankle, thigh | AdaBoost | 95 |

Chapter 4

# ACTIVITY GESTURE RECOGNITION

## 4.1 Introduction

Traditional, accelerometer-based activity recognition systems have been used to monitor and assess elderly individuals by detecting and recognizing the high level ambulatory movements (walking, sitting, standing, climbing stairs, etc.) that are part of the activities of daily life (ADL). The steady posture or the repetitive movement that defines the simple activities facilitates easy and reliable recognition as discussed in the work presented in the previous chapter. However, complex activities such as making a drink or cooking consists of various complex, short duration movements (predominantly hand movements) – activity gestures, along with many interactions with objects. Tracking these complex activities relies on reliable recognition of both the objects and the activity gestures.

The primary aim of this work is to support multi-modal systems for tracking the complex activity rather than to develop an activity recognition system based solely on activity gestures. There are many approaches in the literature where the overall activity of an individual is determined using sensors embedded in the environment such as RFID tags or reed switches as demonstrated by Kasteren et al. [54], Patterson et al. [39]. The problem with these approaches is that, while they provide high-level information about the activity such as making a drink or making brownie, by gathering information about the objects, they do not reliably detect the tasks involved in these activities. For example, given that an individual is interacting with a spoon, kettle, tea bags and water, it is easy to reliably infer that the activity is making tea by exploiting object-activity relationships. But mere object information is insufficient for recognizing the tasks involved in the activity such as pouring milk, scooping sugar, stirring tea. That is, holding a spoon in hand provides information about the possible tasks that the person is doing with the spoon such as scooping sugar, stirring milk, or mixing tea, but cannot pin point which among these is he/she actually performing. This fine level of information is required to effectively predict the future tasks that an individual might perform. The movement information is essential for understanding

the activity task. While it is probably not feasible to differentiate these gestures, from movements in other activities of daily living; given the context that the current activity is making a drink, is it possible to detect and recognize the activity gestures is the problem addressed here. The context can be derived through other sensing mechanisms using information such as the location of the individual, or objects of interaction and is assumed to be given in this work.

This chapter discusses the problem of activity gesture recognition. The following chapter delves into the more complex problem of gesture spotting in the presence of invalid samples. Section 4.2 reviews the related work in the area of gesture recognition. It discusses approaches proposed in general for gesture recognition irrespective of the sensing modality. The data used to conduct this study is presented in Section 4.3. An analysis of accelerometer features for activity gesture recognition is discussed in Section 4.4. The classification results for the different evaluation scenarios are elaborated in Section 4.5, with Section 4.6 concluding the work presented in this chapter.

## 4.2 Related Work

The recognition of isolated gestures has been studied extensively over years and many approaches have been proposed to tackle the diverse problems. Given a gesture/activity sample, these approaches classify it as belonging to one of the labels. They do not attempt to detect where the activity or gesture began or ended in a continuous stream. In general, these approaches can be broadly divided into three categories based on the computational framework adopted for classification: *Template based, generative and discriminative approaches*. In the next three sub-sections a review of the different approaches in these three categories are discussed.

### 4.2.1 Template based approaches

Template matching approaches are transductive in nature and can be divided into two categories based on the type of the templates: fixed length and variable length. Fixed length template matching using k-Nearest Neighbor (k-NN) has been quite popular among researchers for classifying simple activities using accelerometer data. Foerster et al. [11] employ k-NN for recognizing ambulatory movements using templates defined by the features extracted from fixed

length window frames of accelerometer data placed at four different locations on the body. Jafari et al. [15] describe a method to detect the transitions between ambulation again using feature based templates in conjunction with k-NN. Maurer et al. [43] also investigate the performance of k-NN for recognizing ambulation. All of these approaches first extract features from the acceleration movement pattern, which in-turn are used as the templates for matching.

Variable length template matching is performed using a simple length based normalization or approaches based on the more advanced dynamic time warping (DTW) as demonstrated by Sakoe et al. [55]. Corradini et al. [56] use DTW as a matching technique for determining the similarity between the unknown input and a set of previously defined templates for video based sign language recognition. Darrell and Pentland [57] employ DTW for matching space-time motion trajectories associated with manipulating objects for recognizing isolated gestures. Niyogi and Adelson [58] and Gavrila and Davis [59] use DTW to match sequences of joint model configurations obtained from image sequences. Veeraraghavan et al. [60] propose a DTW method for action recognition that allows better modeling of variations within model sequences. More recently, [61] propose a constrained dynamic programming based DTW approach for spotting gestures from continuous data streams. The approach proposed by them unifies the spatial and temporal characteristics of the sample for the purpose of matching. While the DTW based approach is quite popular among the computer vision community, there are only a few approaches that employ DTW for accelerometer based movement pattern recognition. Muscillo et al [51] apply a DTW based k-NN for matching input samples against ambulation templates for activity recognition. One of the main drawbacks of DTW based template matching is the computational complexity of determining the warped distance.

### 4.2.2 Generative Approaches

Generative models are used in machine learning for either modeling data directly (i.e., modeling observed draws from a probability density function), or as an intermediate step to forming a conditional probability density function. A conditional distribution can be formed from a generative model through the use of Bayes' rule. Generative models contrast with discriminative

models, in that a generative model is a full probability model of all variables, whereas a discriminative model provides a model only of the target variable conditional on the observed variables. Examples of generative models include Gaussian mixture model, hidden Markov model and naive Bayes.

Hidden Markov models (HMM) are probably the most commonly used generative models for modeling movement patterns. It is a powerful generative model that includes a hidden state network. HMM is rich in mathematical structures; it serves as the theoretical basis for a wide range of applications. It can model spatio-temporal information in a natural way. It also has elegant and efficient algorithms for learning and recognition, such as the Baum-Welch algorithm and Viterbi search algorithm. HMM has attracted a lot of attention as a useful tool for modeling the spatio-temporal variability in gestures. The unique internal segmentation property of the HMM i.e., the states and transitions of a trained HMM represent sub-patterns of a gesture and their sequential order, makes it a popular choice for modeling varying length spatio-temporal patterns.

Starner and Pentland [62] implemented an HMM-based system for recognizing sentence-level American Sign Language (ASL) without explicitly modeling fingers from video sequences. Bregler et al. [63] learn a kind of switching-state HMM over a set of autoregressive models, each approximating linear motions of blobs in a video frame. Oka et al. [64] propose HMM based recognition of gestures from motion trajectories of finger tips tracked in a video stream. Gandy et al. [65] use HMM for modeling hand gestures from motion trajectories determined using an IR camera. Kahol et al [66] use HMMs for recognizing surgical gestures from data obtained through a data glove. Other computer vision approaches using HMMs for movement pattern recognition are by Wilson et al [67], Brand et al [68], Wang et al[69] and Lv et al [70].

Lester et al [46] employ HMM as a post processing tool to smoothen out the recognition results of an AdaBoost classifier for detecting human activities using data from on-body sensors. Olguin and Pentland [216] explore HMM for recognizing ambulatory movements from inertial sensor data obtained from different on-body locations. Ward et al. [34] use HMM on top of features obtained from Linear Discriminant analysis for recognizing activities in a workshop such as sawing,

drilling etc. Amft et al [35] model gestures involved in dietary activities using HMM by fusing accelerometer and microphone streams. Junker et al. [32] use HMM for recognizing gestures from body-worn inertial sensors in their two-stage user activity detection algorithm. Al-ani et al [16] model walking speeds and transitions between different ambulation using a HMM.

Various extensions to the more general class of dynamic Bayesian networks (DBN) have been proposed to overcome limitations of HMM as illustrated by Ghahramani et al. [71] . Brand et al. [68] learn coupled HMMs to model interactions between several state variables. They use a two state coupled HMM to recognize interactions between left and right hand motions during Tai Chi exercises. Kahol et al [72] propose coupled HMM based on human anatomy for segmenting and recognizing human gestures in dance sequences. Park and Aggarwal [73] use a complex DBN to model interactions between two persons, such as hugging, handshaking, and punching. Peursum et al [74] model interactions between people and objects in their work using Bayesian networks. Nguyen et al [75] propose to use hierarchical HMMs for video activity recognition. Jojic et al [76] and Toyama and Blake [77] extend HMMs with separate latent states for posture and view for pose and view invariant activity recognition.

More recently there have been many attempts at incorporating a discriminative model into the HMM framework. This has resulted in models such as large margin HMM and continuous density HMM that are being used in the domain of speech recognition as discussed by Altun et al. [78], Sha and Lawrence [79] and Chen et al [80].

### 4.2.3 Discriminative Approaches

In contrast to their generative counterparts, discriminative approaches model the conditional probability of the target variable given the observations. These approaches typically model the boundary separating the data samples between two classes. While this genre of techniques is commonly used for a variety of classification and regression tasks, it is applicable to the problem of recognizing movement patterns, when the pattern is described in terms of a feature that implicitly encodes the spatio-temporal variation in the movement pattern.

Cui and Weng [81] discuss an appearance based multidimensional discriminant analysis approach for selecting linearly discriminating features for hand gesture recognition using a recursive partition tree approximator. Lester et al [46] use AdaBoost to extract probabilities for classifying activities using data from multiple modalities. They also use AdaBoost as a feature analyzer to determine the importance of the different modalities for activity recognition. Ong et al. [82] propose a modified version of the AdaBoost algorithm using decision trees as weak learners for recognizing hand shapes (gestures) from video streams. Lv et al [70] use multi-class AdaBoost for recognizing human action patterns from 3-D data motion capture. Pentney et al. [83] discuss a virtual evidence based boosting algorithm for recognizing activities by fusing accelerometer data with RFID tags. Yang et al. [84] use SVM for classification of static or short duration hand gestures for improving the overall performance of the American Sign Language recognition system. Morency and Darrell [85] discuss an intelligent user interface through head gestures classified using SVM.

A significant amount of recent work has shown the power of discriminative models such as conditional random fields (CRF) for specific sequence labeling tasks. CRF use an exponential distribution to model the entire sequence given the observation sequence[86] [87][88]. This avoids the independence assumption between observations, and allows non-local dependencies between state and observations. A Markov assumption may still be enforced in the state sequence, allowing inference to be performed efficiently using dynamic programming. CRFs assign a label for each observation (e.g., each time point in a sequence), and they neither capture hidden states nor directly provide a way to estimate the conditional probability of a class label for an entire sequence. Sminchisescu et al [88] applied CRFs to classify human motion activities (i.e. walking, jumping, etc) and showed improvements over an HMM approach. When the sequence under consideration has distinct sub-structure, models such as hidden-state CRFs (HCRF) [89] that exploit hidden state is advantageous. Wang et al [89] propose a gesture recognition technique based on HCRF, which can estimate a class given a segmented sequence. Since they are trained on sets of pre-segmented sequences, these HCRF models capture only the internal structure and not

the dynamics between gesture labels. Morency et al [90] proposed latent-dynamic hidden CRFs (LDCRFs) for vision based gesture recognition. LDCRFs are a framework for detecting and recognizing sequential data, which can model the sub-structure of a label and learn dynamics between labels.

One can thus see that there has been substantial amount of work in the literature on gesture recognition, which is predominantly vision based. There has not been much work in exploring the feasibility of these approaches for accelerometer based gesture recognition. It evaluates the performance of the classifiers from each of the three genres for recognition of activity gestures. It focuses on the recognition of gestures that build the basis for the inference of more abstract activities. The primary aim is to support complex activity spotting systems rather than to develop an activity spotting system based solely on hand gestures. Nonetheless, this work shows how, for suitable domains, good performance can be achieved without any additional information. Within this scope the work done in this chapter makes the following contribution: *It demonstrates the feasibility of recognizing subtle short duration and non-repetitive activity gestures using discriminative classifiers. Since most of the work on accelerometer based movement recognition focuses on simple activity recognition, the work presented in this chapter is unique and has merit.*

**4.3 Data collection**

Two different accelerometers were used for collecting data for this study. The first accelerometer is the Wi-tilt unit that is described in Section 3.4. Five activity gestures namely – lift to mouth, pour, stir, scoop and unscrew cap were collected using these sensors. Two different scenarios were used for collecting these data. While a semi-naturalistic mode of collecting the data that permits a greater degree of freedom to the subject in performing the activity is proposed by Bao et al [26], for the activity gestures considered in this work it is not practical for a subject to perform the same activity a number of times in one session. Instead, an alternative data capture session was devised during which the subjects enacted the same movements with mock objects a number of times, thereby providing sufficient data samples for training. For each of the five gestures, alternate scenarios representing the actual movement needed to perform the activity were designed. The

mock scenarios are explained in Table 7. Explicit instructions were given on how to perform each movement. Each subject was asked to perform each of the actions 20 times. Figure 16 describes the settings for the data capture session, and a video camera was used to record the sessions. The subject started in a 'rest' state, where the hands rested on the table, and the subject began the action after receiving a cue from the experimenter. The video was used off line to synchronize the accelerometer data and to extract relevant portions from the continuous data stream that corresponded to the actions. In addition data was also collected from the subjects performing multiple trials of the actual activity in an unrestricted manner. This consisted of two sessions performed during different days. During each session the subjects were asked to make a glass of powdered drink and drink it, twice. The video recordings from these sessions were later used for annotating the actions.

All the subjects in our experiments were college students aged 22-28 years, and all of the subjects were right-handed. Three accelerometers were used for data collection. The first accelerometer was placed on the wrist of the right hand and the second accelerometer was placed just above the right elbow. We observed during our experimental data capture sessions that subjects sometimes used their left hand as support during the actions, so the third accelerometer was placed on the left wrist. Each of the accelerometer-sensing units was attached using Velcro tapes.

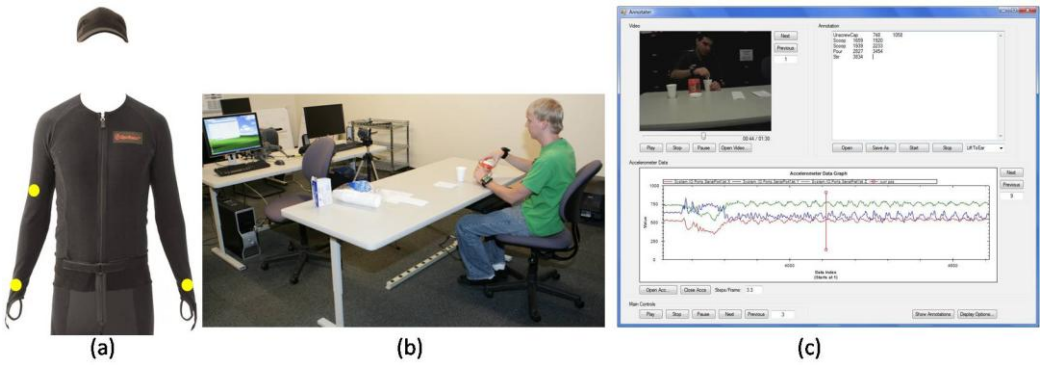



Figure 16: (a) Illustrates the different locations for placing the accelerometer, (b) depicts the data capture setup and (c) shows the annotation software that synchronizes the video and accelerometer data

Table 7: The mock and semi-naturalistic scenarios used for data capture

| Activity Gesture | Mock Scenario (CS1) | Semi-Naturalistic Scenario(CS2) |
|---|---|---|
| **Pour** | Take the glass that is full and pour its contents into the empty glass. Pour a small quantity every time. | Pour the water from the glass. |
| **Scoop** | Use a spoon to scoop contents from the glass that is full into the empty glass | Use two scoops of powder for making the drink |
| **Unscrew Cap** | Unscrew the lid of the water bottle. Pause for a couple of seconds. Screw on the lid on the bottle | Open the powder drink jar, and close it after you finish using it |
| **Stir** | Take the spoon and stir the contents of the glass for 30 seconds | Ensure the powdered drink has dissolved by stirring the mixture |
| **Lift to Mouth** | Take an empty glass and pretend that you are drinking water from the glass by taking several short sips. | Drink the glass of beverage that was prepared |




Figure 17: ZStar 3 accelerometer and the ZigBee wireless receiver manufactured by Freescale

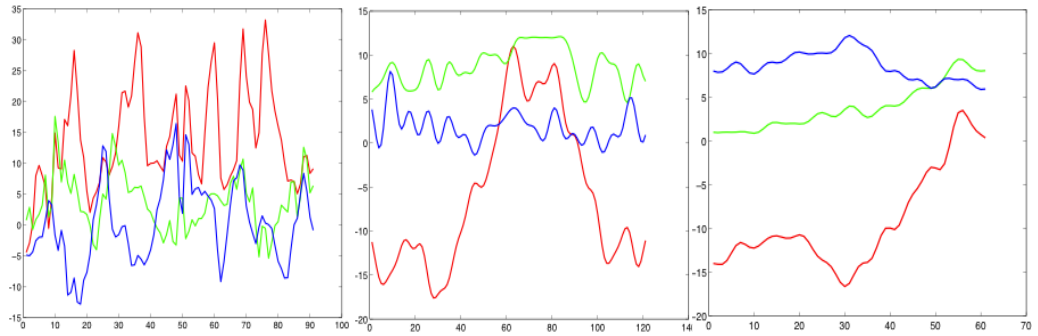
The second set of activity gestures was collected using three ZStar accelerometers (Figure 17) [156] developed by Freescale. The ZStar accelerometers follow the ZigBee wireless standard for communicating to a base station. They are much smaller in form factor when compared to Sparkfun's Wi-Tilt accelerometers Figure 12. They offer three different sampling rates (30, 60 and 120Hz). In this study the sampling rate was set to 30Hz, to support data collection from multiple sensors. The sensors were attached to the dominant wrist, lower and upper arm through Velcro. Data was collected from three subjects for 11 activity gestures. Each subject repeated the activity gesture 20 times with a short interval between successive repetitions. The description of the activity gestures is presented in Table 8. Similar to the data capture setup described previously, video recordings of the session were used to synchronize and annotate the accelerometer data. Figure 18 presents a sample of the accelerometer data for each of the activity gestures. These

activity gesture samples were used only for the classification process described in the chapter and the spotting problem described in the next chapter.

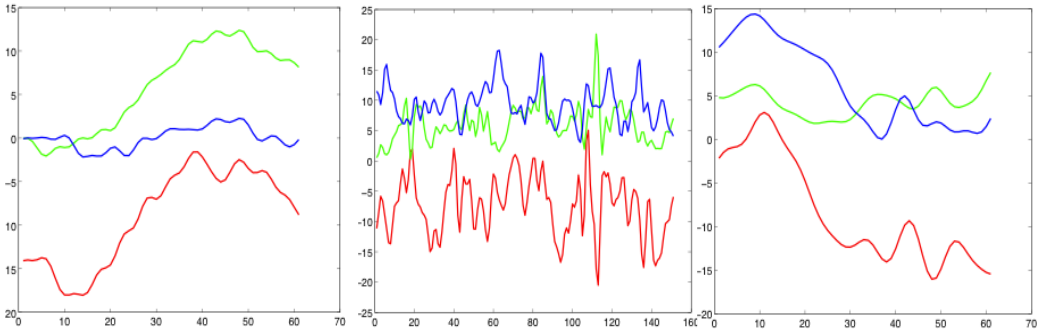
Table 8: Description of the activity gestures collected using the ZStar accelerometers for CS3

| **Activity Gesture** | **Mock Scenario (CS3)** |
|---|---|
| **Lift to Mouth 1 (LM1)** | Lift the glass on the table to your mouth, imitating a drink action. |
| **Lift to Mouth 2 (LM2)** | Lift the spoon/fork on the table to your mouth, imitating an eat action. |
| **Scoop** | Use the spoon to scoop the contents of one glass into another |
| **Pour** | Pour the contents in one glass into another. |
| **Stir** | Use the spoon to stir the water in the glass |
| **Mix** | Mix the contents of the vessel |
| **Unscrew Cap** | Unscrew the lid of the bottle |
| **Spread** | Perform a spread action to imitate spreading jam/butter on a slice of bread using the knife and the piece of flat board. |
| **Sprinkle** | Use the sprinkler and sprinkle the contents into the bowl. |
| **Comb** | Comb your hair |
| **Lift to ear** | Lift the cell phone to your ear, imitating the action of answering a telephone call. |

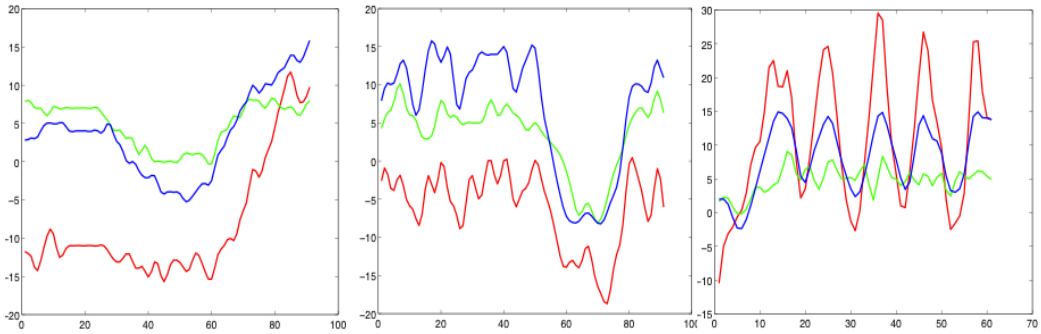
These gestures represent a variety of movement patterns. For example, 'Lift to Ear' and the two 'Lift to Mouth' gestures have a *well-defined smooth motion* involving the bending of the elbow, while the wrist reaches for the ear or mouth. These movements are carried out in short intervals leading to a *steady posture* of the wrist and the lower arm. A *well-defined continuous motion* of the wrist characterizes gestures such as 'Scoop' and 'Pour'. A *repetitive twisting motion* of the wrist defines gestures such as 'Unscrew Cap' and 'Stir'. *Ill-defined repetitive* movement of the wrist characterizes gestures 'Spread' and 'Mix'. 'Sprinkle' is defined an *jerky repetitive movement* of the wrist



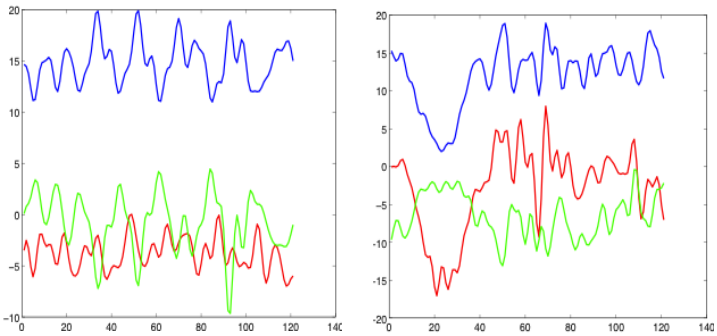
Comb Lift to Ear Lift to Mouth 1



Lift to Mouth Mix Pou



Scoo Sprea Sprinkl



Stir Unscrew

Figure 18: Activity gesture samples from the gesture dataset CS3

### 4.4 Feature Analysis

Figure 19 illustrates the samples collected from a single subject using the mock setup (activity gesture set CS1). These signals were obtained from the accelerometer placed on the right wrist. The most evident observations are that samples are of varying length and that each action can be distinguished by observing the acceleration patterns. For example, unscrewing the cap can be defined by a number of rapid repetitive movements, while slower repetitive movements represent stir. A dip in the *z*- axis acceleration appears for the actions scoop and lift to mouth, but the *y*- axis values increase for scoop and fall significantly for lift to mouth. Similar observations can be made for other actions, leading to the conclusion that it is possible to differentiate these actions using the accelerometer data we gathered.

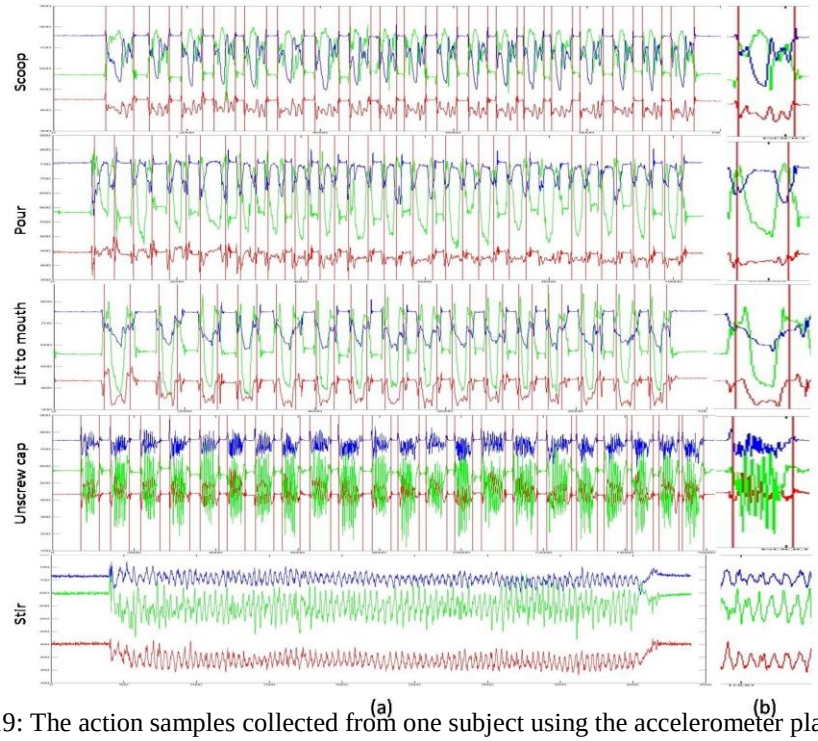



Figure 19: The action samples collected from one subject using the accelerometer placed on the right wrist. The RGB lines stand for the x-, y- and z-axis respectively. The continuous stream of data was annotated offline using the video recording. (a) Samples recorded from a single data capture session. (b) One representative sample for each of the actions

To analyze these gestures, several statistical and aggregate spectral features: mean, variance, correlation, spectral energy, and spectral entropy were extracted. The spectral energy value was computed as the sum of the squared amplitude of discrete FFT coefficients, ignoring the DC

coefficient. Similarly, the spectral entropy was computed as the normalized information entropy of the magnitudes of the discrete FFT coefficients, also ignoring the DC coefficient. Since the DC coefficient represents the mean of the signal, and is already being captured explicitly, we ignored it while computing these additional features. Each feature was computed for each axis of each accelerometer. Pair wise correlation between all the accelerometer axes (across all accelerometers) was also computed.

The feature analysis presented here is based on the data collected from the right wrist only. The problem of variable length sequences was circumvented by extracting fixed length feature vectors (mean: 6, variance: 6, correlation: 15, energy: 6, entropy: 6, totality of dimensions: 39) from the sequence. Pair wise Euclidean distance between all samples of all actions from a single subject, using normalized feature vectors was computed. This is illustrated in Figure 20 which shows that each action sample is very coherent, as indicated by the dark bands along the diagonals, and at the

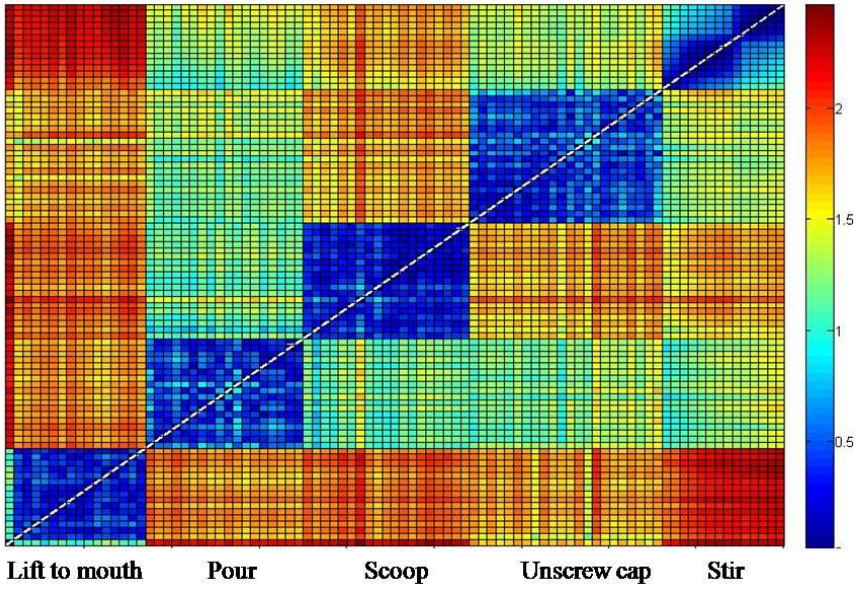



Figure 20: Pair wise Euclidean distance between samples of a single subject

same time, samples from different classes are relatively dissimilar indicated by the lighter shades. The first band at the bottom left of the matrix corresponding to the lift to mouth action is very different from all the other actions. While this illustration is from just one subject, similar patterns were seen across all the other subjects.

Similarity of samples from different subjects was visualized by conducting a principal component analysis (PCA) on the feature vectors. The distribution of the samples belonging to the different classes reduced to a 3-dimensional space is illustrated in Figure 21. The action lift to mouth represented by the cross marks is distinct, but there was some overlap between the other actions (probably indicating that each subject performed the actions differently). Data points, belonging to the same class and the same subject, obtained after PCA, resulted in tight clusters. The multiple clusters belonging to the same class corresponded to different subjects.

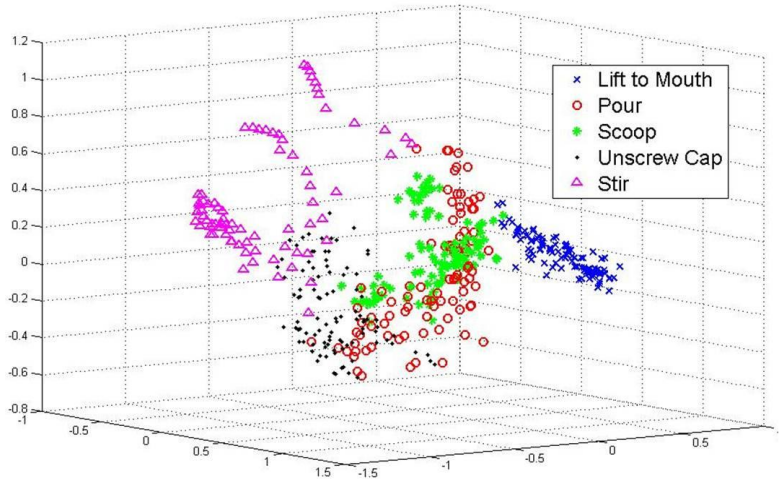



Figure 21: Principal Component Analysis of activity gesture samples from all the subjects

These features were further analyzed using the AdaBoost framework. AdaBoost works by iteratively calling a certain weak learning algorithm known as the base learner to arrive at a classifier that gives better than 'by chance' accuracy, and constantly updates the distribution of weights of the samples after every iteration. If the base learner is a decision stump, then the process of boosting results in a number of weighted decision stumps, which when combined linearly gives the final class label. These weights associated with the decision stumps can further be used to derive weights for each of the feature dimensions. Feature dimensions with relatively high weights can then be considered to characterize the corresponding gestures more effectively. This process was adopted to study the importance of the different features extracted with respect to the activity gestures.

In addition to this methodology, a technique for computing feature weights using SVM, as discussed by Zhang et al [91] was implemented. Multiple SVMs were trained with only one feature dimension considered at a time, and the area under the Receiver Operating Characteristics (ROC) curve, was used to derive the weight associated with the dimension. This was performed for every feature dimension and for every activity gesture. The objective was to compare and contrast as well as, validate the results obtained through AdaBoost.

The contribution of the individual features towards the recognition of the different gestures was calculated using the weights computed by the AdaBoost algorithm and the SVM ROC method. The subfigures in Figure 22 present the weights for each of the 39 features computed using both the approaches for the gestures lift to mouth, pour, scoop unscrew cap and stir respectively. The distribution of weights as computed by AdaBoost shows the relative significance of certain features as indicated by peaks in the plots. The most evident observation from the plots is the uniformity in the distribution of weights as computed using the SVM ROC method. While the distribution is uniform, the trend observed for the SVM ROC method shows small peaks that correspond to the features having higher weights computed using AdaBoost.

Based on the distribution of weights computed through AdaBoost, for every gesture, features with weights, twice or more than that of the uniform weights were selected. Table 9 presents the features that were selected through this process. It is interesting to note that with the exception of the gesture unscrew cap, all other gestures have nearly 10 dominant features (out of a total of 39) contributing to nearly $60\%$ of the total weight. The significance of these features was further measured, by observing the classification performance by training AdaBoost only on these features. The AdaBoost trained on this reduced feature set resulted in an accuracy of 89.12%, which was approximately the same accuracy obtained using all the features. This clearly validates the use of AdaBoost as a feature selection mechanism.

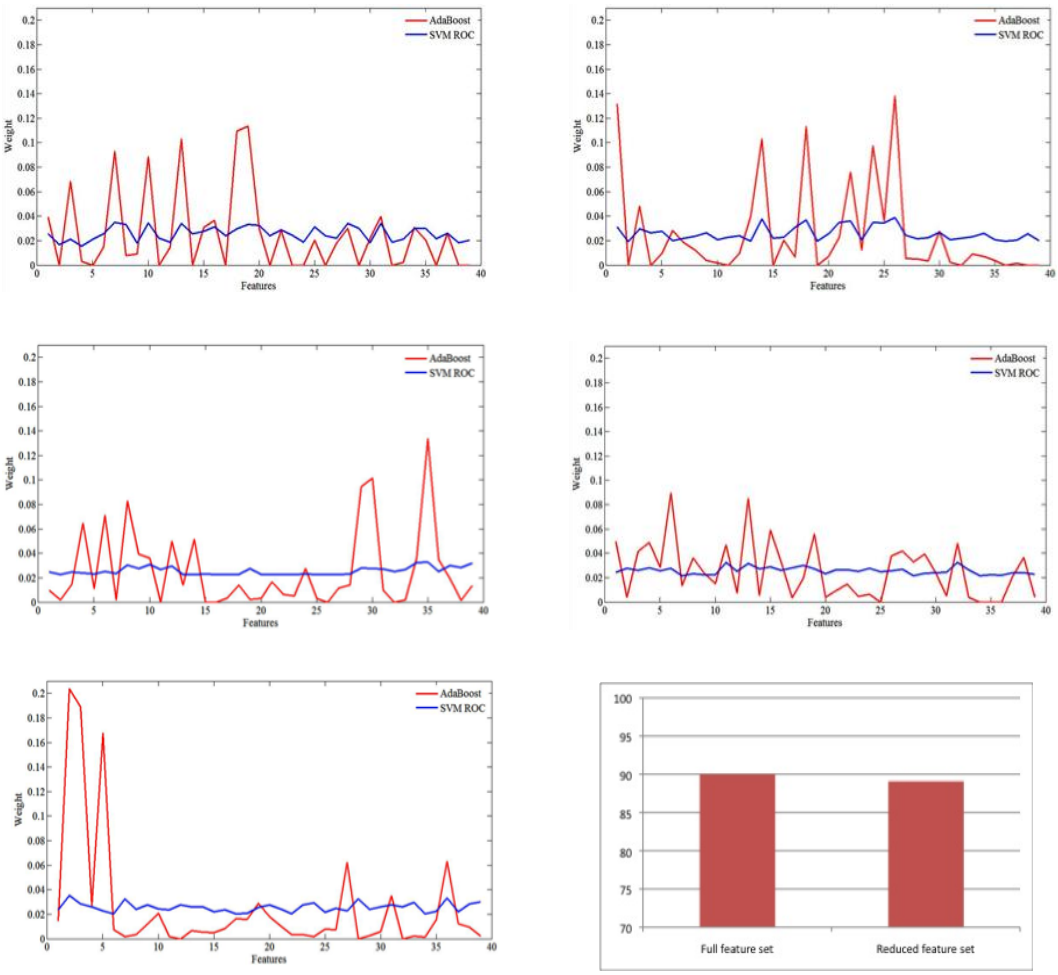



Figure 22: The weights for the different features as computed by AdaBoost and SVM ROC for each of the activity gesture. The final plot is the comparison of the performance of the reduced feature set as derived through AdaBoost and the original feature set.

It was also noted that the features selected were intuitive in characterizing the gestures. As an example, for the activity gesture lift to mouth that is represented by the variations in the accelerometer data on the wrist, the majority of the features selected corresponded to this characterization. A similar observation was made with respect to all the other gestures as well, except for unscrew cap. While one would expect the energy and entropy features for the unscrew cap to have high significance due to the repetitive movement of the hand, low weight values were observed for these features. This probably also indicates a need to explore other features, for understanding some of these gestures.

Table 9: Dominant Features for each of the gestures as calculated using AdaBoost (X1, Y1, Z1 and X2, Y2, Z2 are the X, Y and Z axes of the accelerometers placed on the wrist and elbow respectively

| Activity Gesture | Dominant Feature | Aggregate Weight |
|---|---|---|
| **Lift to Mouth** | Mean (Z1), Variance (X1, X2), Correlation (X1 and Y1, Y1 and Z1, Y1 and X2) | 57.59 |
| **Pour** | Mean (X1), Correlation (X1 and Z1, Y1 and Z1, Z1 and X2, Z1 and Z2, X2 and Z2) | 65.88 |
| **Scoop** | Mean (Y1, Z1, Y2), Correlation (Y2 and Z2), Entropy (Z1) | 68.56 |
| **Unscrew cap** | Mean (Z2), Correlation (X1 and Y1, X1 and X2, Y1 and X2) | 28.99 |
| **Stir** | Mean (X2, Z2), Variance (Y1), Correlation (X1 and Z1), Energy (Y1, Z1), Entropy(Y1) | 59.94 |

### 4.5 Classification – Results and Discussion

The performance of k-NN a template matching model, HMM a generative model and AdaBoost a discriminative model was evaluated using three scenarios:

- Subject Independent Evaluation: This was performed using a leave one out strategy where data from 4 subjects was used as the training data to be tested on the 5$^{th}$ subject, in a round robin fashion. This evaluation provides the most difficult classification scenario for the classifiers.
- Subject Adaptive Evaluation: This was also performed using a leave one out strategy, with a modification. In addition to the data from 4 subjects, 25% the data from the 5$^{th}$ subject was also used during training and the resulting classifier was tested on the remaining 75% of the samples of the 5$^{th}$ subject. More details about subject adaptation have been discussed in Section 4.5.1.
- Subject Dependent Evaluation: In this scenario, part of the data (50%) from a subject was used for training and the remaining for testing. This can be correlated to a scenario where the system has been personalized for a single subject and is being evaluated for this subject alone.

AdaBoost had demonstrated excellent results in previous experiments with similar accelerometer data for ambulatory motion [52]. AdaBoost combines simple, weak learners like decision stumps into a powerful classifier. A linear combination of the weighted weak hypothesis learnt in each

iteration is used as the final classifier. A separate AdaBoost classifier was trained for each class, considering the data belonging to the class as positive samples and all other classes as negative samples. AdaBoost defines the boundary that separates the samples from each other. Each classifier passed through a maximum of 100 iterations. The training error reduced exponentially over the iterations and is illustrated in Figure 23. It can be seen that there was a uniform decrease in the training error across all the actions.

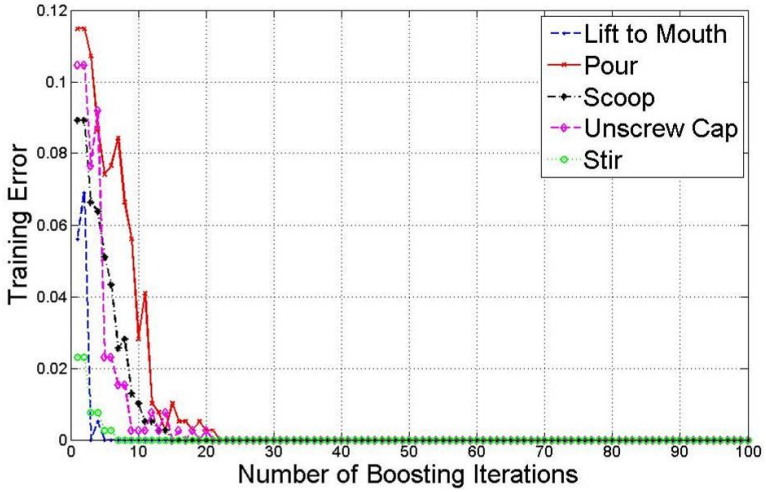



Figure 23: Training error using AdaBoost across iterations for the different activity gestures

HMMs are effective for modeling temporal data because the global features extracted from the data do not explicitly capture the temporal nature of the data. Another advantage of HMMs is these samples can be of varying length. As in the case of AdaBoost, a HMM was trained for each of the classes and the label with maximum likelihood estimated by the classifiers was chosen as the winner. The statistical and spectral features were extracted at a rate of 10Hz (determined through empirical evaluation) from the data. This stream of features was used to train the HMMs. Each state in the HMM was modeled using a Gaussian mixture model. The parameters for HMM namely the number of Gaussians at every state and the number of hidden states - was decided using a trial and error method of cross validation. The optimal value for the number of hidden states and the number of GMMs was found to be 3 and 3 respectively. The same HMM parameters were used for training all the classes.

Since k-NN was a common technique used for classifying ambulatory movements, we also experimented with it for classifying hand movements. Dynamic Time Warping based distance techniques that can handle data of varying length need extensive computation and so were not considered. Instead, we chose to use the fixed length features and a Euclidean distance based k-NN.

The subject independent, adaptive and dependent accuracies obtained for gesture set CS1 are summarized in Figure 24. Overall, recognition accuracy is highest for AdaBoost in the subject independent and adaptive evaluations with 90% and 95% accuracy, which is consistent with the past work where it was used for recognizing ambulatory movements [52] [53]. HMMs were the second most accurate algorithm. The lower recognition accuracy for AdaBoost for the subject dependent evaluation can be attributed to the reduction in the number of training samples. K-NN resulted in a mere 43% subject independent accuracy, which is surprising from that fact that prior work using k-NN for recognizing ambulatory movements was able to achieve significantly higher accuracies [11][43][44].

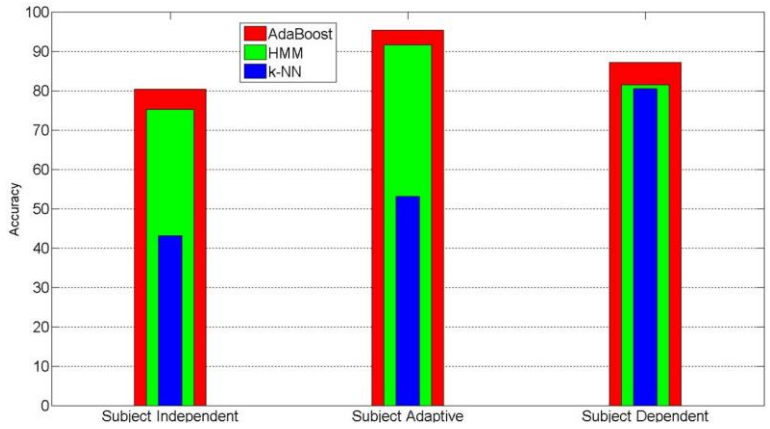



Figure 24: Subject independent, adaptive and dependent accuracies for AdaBoost, HMM and k-NN using the data from the accelerometer on right wrist

Table 10 shows the aggregate confusion matrix for AdaBoost classifier based on the leave one out subject independent evaluation. Recognition accuracies for the stir action were 68%, while lift to mouth, pour and scoop had significantly better accuracies of 95%, 94% and 92% respectively. There was a notable confusion between the actions stir and unscrew cap, both of which involve

repetitive patterns with only marginal differences in the rate at which the action is performed. Thus, there is a chance of miss-classification when stir is performed quickly or unscrew cap is performed slowly.

Table 10: Aggregate confusion matrix obtained for AdaBoost classification

| **Activity Gesture** | **Lift to mouth** | **Pour** | **Scoop** | **Unscrew Cap** | **Stir** |
|---|---|---|---|---|---|
| **Lift to mouth** | 95 | 0 | 5 | 0 | 0 |
| **Pour** | 3 | 94 | 3 | 0 | 0 |
| **Scoop** | 0 | 0 | 92 | 2 | 3 |
| **Unscrew Cap** | 0 | 0 | 8 | 85 | 4 |
| **Stir** | 0 | 0 | 5 | 5 | 65 |

Overall, the recognition accuracy was significantly higher for AdaBoost, probably indicating the need for a discriminative classifier that models the boundaries between the classes for recognition. The high accuracy obtained for the leave one out subject independent evaluation process indicates the presence of strong correlations in the action patterns of different subjects. In addition, since this evaluation used a large training data set (from 4 subjects), the training process would have resulted in a more generalized classifier.

### 4.5.1 Subject Adaptation

With AdaBoost providing the best results, additional experiments were conducted to see how many samples of an individual are required to train the algorithm to learn the individual's movements. This is similar to the subject adaptive evaluation, with varying number of samples from the test subject used for training. Starting with 5% of the test subject's data, more data was added until nearly 100% accuracy was obtained. A 5-fold evaluation process was used, where the classifier trained on the data from 4 subjects is adapted to the $5^{th}$ test subject.

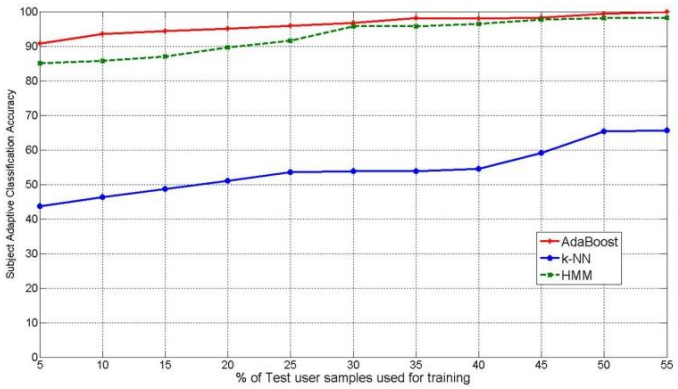



Figure 25: Subject adaptive accuracies obtained using AdaBoost and k-NN for varying number of test subject samples used for training

The objective of this experiment was to determine which techniques achieved maximum accuracy (highest) using minimum number of samples (fastest). Figure 25 illustrates the performance of the three algorithms when varying number of samples from the test subject were used for training. Note that there is a significant jump of about 10% in the performance after adding just 5% of the test samples for training uniformly across all the techniques. As expected, performance increases as the number of samples are increased and specifically AdaBoost and HMM reach close to 100% accuracy with 55% of the samples being used for training. With a very similar performance by AdaBoost and HMM, it is hard to conclude which of the techniques adapt to a subject faster. While k-NN also shows an increase in recognition accuracy, AdaBoost and HMM show the highest and fastest adaptation to a subject.

### 4.5.2 Accelerometer configuration

It is evident from the analysis of the ambulatory movements that placement of the accelerometers plays a crucial role in determining the performance of the system. This holds true for hand movements as well. The discriminatory power of the different accelerometer locations on the hand was evaluated using the AdaBoost classifier. As mentioned before, three locations for placing the accelerometers: right wrist, right elbow and left wrist were used. Since all the subjects were right-handed, the right wrist is the most intuitive location to place the accelerometer. While the palm would be the position that can discriminate the most between the different actions, placing the accelerometer on the palm would restrict the individual's movement and hence was not used.

Table 11 summarizes the recognition accuracies obtained using data from these three accelerometer positions. The best performance was achieved with a combination of accelerometers on the right wrist and the right elbow. This is intuitive, as the subjects used their right hand for all the actions, resulting in a strong correlation between the motion at the wrist and the elbow. Thus monitoring the joints that determine the movement of the hands is useful for recognizing the actions involved with mixing and drinking a powdered beverage. However, movements from the right elbow alone are not sufficient to classify the actions. While our initial hypothesis was that data from the accelerometer on the left wrist, would aid in classification, surprisingly adding acceleration information from the left wrist reduced the performance. A review of the video recordings highlighted one possible explanation for this result: not all subjects used their left hand, nor did subjects use their left hands in a similar manner during the recorded activities.

Table 11: Classification accuracies for different configurations of the accelerometers

| Accelerometer Configuration | Classification Accuracy |
|---|---|
| **Right wrist** | 89.31 |
| **Right elbow** | 74.90 |
| **Right elbow and wrist** | 90.01 |
| **Right and left wrist** | 65.45 |

### 4.5.3 Performance on activity gesture dataset CS2

The same experiment was repeated with the data collected in CS2 corresponding to the real-life scenario. Three training strategies were used to evaluate the performance in this scenario. In the first strategy, models learnt from data collected using CS1 were used to test samples from CS2. In the second strategy, samples from two trials of every subject obtained from CS2 were used for training. However since the number of training samples available from data collected under CS2 is very less, samples collected from CS1 were also included in training in the third strategy. For training scenarios that used data from CS2, the trained models were tested on the remaining trials of the subject. The two test trials was chosen in a round robin fashion, for determining the most generalized accuracy in the given situation. The isolated recognition accuracies for each of the models are shown in Table 12 As expected, there is a drop in performance of AdaBoost when trained on samples using CS1. Since there is an inherent difference in the movement patterns

between the scenarios as CS2 corresponds to an unconstrained environment. The accuracies obtained using models trained on data collected in real-life scenarios are also poor. This is probably because of the lack of sufficient number of training samples. Models were trained with approximately 10-20 samples per class. However, when the samples from both CS1 and CS2 were combined, the performance of AdaBoost improved drastically clearly indicating some similarity in the movement samples from both the scenarios that can assist in the learning process. We explore more on this aspect in Chapter 6 of the dissertation which deals with learning classifier models using auxiliary data samples.

Table 12: Isolated recognition performance on CS2 dataset through classifiers trained on CS1, CS2 and a combination both the datasets

| **Classifiers** | **Gesture data from only CS1** | **Gesture data from only CS2** | **Combination of CS1 and CS2 datasets** |
|---|---|---|---|
| **AdaBoost** | 0.74 | 0.68 | 0.86 |

### 4.5.2 Performance on gesture set CS3

The performance of the different classifiers was again performed on the CS3 dataset consisting of 11 gestures obtained from 5 subjects as described in Section 4.3. The subject independent precision and recall obtained for each of the classifier over 5-fold cross validation is presented in Figure 26. As noted previously, both SVM and AdaBoost result in superior performance over HMM and k-NN, validating the effectiveness of discriminative classifiers over generative classifiers. Radial Basis Function kernel SVM resulted in the best performance with a 2-3% better precision and recall values over AdaBoost trained on decision stumps. These results are consistent with the results obtained on the CS1 gesture dataset. The confusion matrices for the different classifiers are presented in Table 13, Table 14,Table 15and Table 16. The precision and recall values for the individual gestures are also shown in these tables.

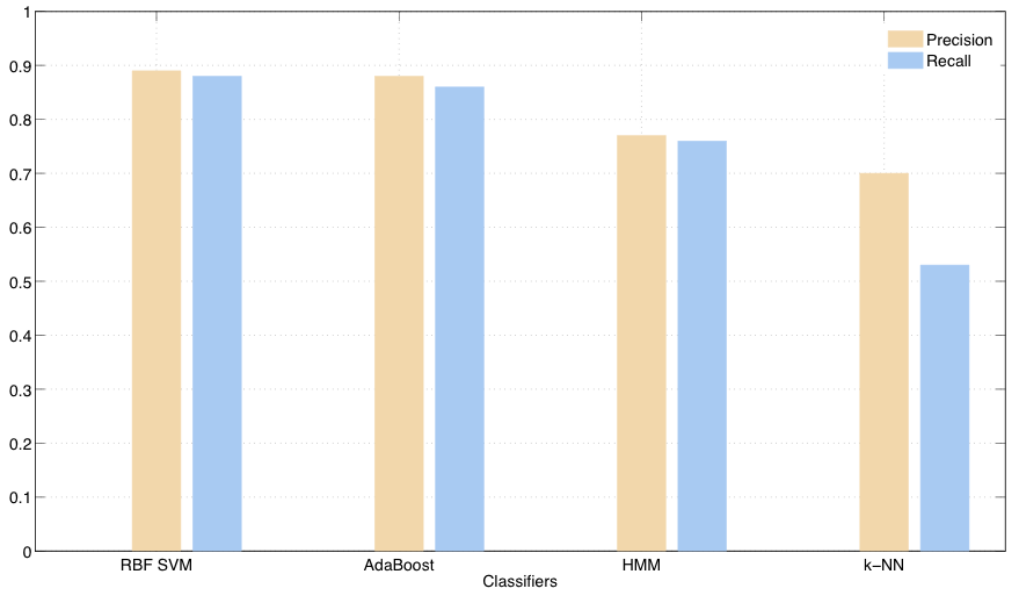



Figure 26: Precision and recall values for the different classifiers obtained on the data belonging to the case study 3 (CS3).

Table 13: Aggregate confusion matrix, precision and recall values for the individual gestures obtained from RBF kernel SVM

| Activity / Gesture | Comb | LE | LM1 | LM2 | Mix | Pour | Scoop | Spread | Sprinkle | Stir | Unscrew cap | **Recall** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Comb | 65 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | **0.89** |
| LE | 0 | 66 | 14 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | **0.78** |
| LM1 | 0 | 9 | 60 | 11 | 0 | 7 | 1 | 1 | 0 | 0 | 0 | **0.68** |
| LM2 | 0 | 0 | 3 | 80 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | **0.91** |
| Mix | 0 | 0 | 0 | 0 | 84 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| Pour | 0 | 0 | 0 | 0 | 0 | 84 | 3 | 0 | 0 | 0 | 0 | **0.97** |
| Scoop | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 1 | 0 | **0.99** |
| Spread | 0 | 0 | 1 | 0 | 0 | 9 | 8 | 73 | 0 | 0 | 0 | **0.80** |
| Sprinkle | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 84 | 0 | 1 | **0.96** |
| Stir | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 46 | 9 | **0.73** |
| Unscrew cap | 0 | 2 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 84 | **0.93** |
| **Precision** | **1** | **0.79** | **0.77** | **0.84** | **1** | **0.78** | **0.90** | **0.90** | **1** | **0.87** | **0.88** | |

Table 14: Aggregate confusion matrix, precision and recall values for the individual gestures obtained through AdaBoost with decision stump as the weak learner

| Activity<br>Gesture | Comb | LE | LM1 | LM2 | Mix | Pour | Scoop | Spread | Sprinkle | Stir | Unscrew cap | **Recall** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Comb | 60 | 12 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0.82** |
| LE | 1 | 65 | 9 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | **0.76** |
| LM1 | 0 | 6 | 72 | 2 | 0 | 8 | 0 | 1 | 0 | 0 | 0 | **0.80** |
| LM2 | 0 | 0 | 6 | 78 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | **0.88** |
| Mix | 0 | 0 | 0 | 0 | 84 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| Pour | 0 | 0 | 6 | 0 | 0 | 79 | 0 | 2 | 0 | 0 | 0 | **0.91** |
| Scoop | 0 | 0 | 0 | 1 | 0 | 0 | 98 | 1 | 0 | 1 | 0 | **0.97** |
| Spread | 0 | 2 | 0 | 2 | 0 | 6 | 15 | 63 | 0 | 2 | 1 | **0.69** |
| Sprinkle | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 88 | 0 | 0 | **1** |
| Stir | 0 | 1 | 0 | 0 | 0 | 0 | 3 | 5 | 5 | 43 | 6 | **0.69** |
| Unscrew cap | 0 | 1 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 86 | **0.94** |
| **Precision** | **0.98** | **0.75** | **0.77** | **0.83** | **0.98** | **0.84** | **0.81** | **0.88** | **0.95** | **0.93** | **0.92** | |

Table 15: Aggregate confusion matrix, precision and recall values for the individual gestures obtained through HMM

| Activity<br>Gesture | Comb | LE | LM1 | LM2 | Mix | Pour | Scoop | Spread | Sprinkle | Stir | Unscrew cap | **Recall** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Comb | 50 | 7 | 0 | 0 | 10 | 0 | 0 | 1 | 1 | 0 | 4 | **0.69** |
| LE | 4 | 65 | 9 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 4 | **0.75** |
| LM1 | 0 | 10 | 59 | 3 | 2 | 7 | 0 | 2 | 0 | 1 | 5 | **0.66** |
| LM2 | 0 | 6 | 3 | 62 | 0 | 1 | 7 | 0 | 0 | 0 | 9 | **0.71** |
| Mix | 0 | 0 | 0 | 0 | 84 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| Pour | 0 | 0 | 10 | 0 | 0 | 74 | 0 | 3 | 0 | 0 | 0 | **0.86** |
| Scoop | 0 | 0 | 0 | 0 | 0 | 17 | 74 | 10 | 0 | 0 | 0 | **0.73** |
| Spread | 1 | 0 | 0 | 0 | 12 | 3 | 20 | 48 | 1 | 6 | 0 | **0.53** |
| Sprinkle | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 81 | 0 | 0 | **0.92** |
| Stir | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 6 | 1 | 54 | 0 | **0.86** |
| Unscrew cap | 1 | 5 | 6 | 1 | 1 | 10 | 5 | 3 | 0 | 3 | 56 | **0.61** |
| **Precision** | **0.89** | **0.70** | **0.68** | **0.91** | **0.72** | **0.66** | **0.69** | **0.65** | **0.97** | **0.84** | **0.72** | |

Table 16: Aggregate confusion matrix, precision and recall values for the individual gestures obtained through k-NN

| Activity / Gesture | Comb | LE | LM1 | LM2 | Mix | Pour | Scoop | Spread | Sprinkle | Stir | Unscrew cap | **Recall** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Comb | 66 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | **0.90** |
| LE | 60 | 8 | 0 | 0 | 15 | 0 | 0 | 2 | 0 | 0 | 0 | **0.10** |
| LM1 | 38 | 0 | 9 | 0 | 18 | 0 | 0 | 18 | 6 | 0 | 0 | **0.10** |
| LM2 | 33 | 0 | 0 | 32 | 0 | 0 | 0 | 20 | 3 | 0 | 0 | **0.37** |
| Mix | 0 | 0 | 0 | 0 | 84 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| Pour | 0 | 3 | 5 | 0 | 9 | 70 | 0 | 0 | 0 | 0 | 0 | **0.81** |
| Scoop | 12 | 0 | 0 | 0 | 0 | 0 | 63 | 24 | 2 | 0 | 0 | **0.63** |
| Spread | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 52 | 0 | 0 | 0 | **0.57** |
| Sprinkle | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 70 | 0 | 0 | **0.80** |
| Stir | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 5 | 36 | 6 | **0.57** |
| Unscrew cap | 35 | 0 | 0 | 0 | 21 | 0 | 0 | 17 | 11 | 0 | 7 | **0.08** |
| **Precision** | **0.22** | **0.72** | **0.64** | **1** | **0.56** | **1** | **1** | **0.35** | **0.70** | **1** | **0.54** | |

One of the striking observations from these tables is that the gesture ‘Mix’ has a recall value of 1 for all the classifiers. This illustrates the distinctiveness of this gesture from the others when observed that the SVM classifier results in 100% precision in recognizing this gesture. The distinctiveness of this gesture can be attributed to the movement patterns captured by the sensor placed on the wrist and lower arm. The ‘Sprinkle’ gesture is the second best recognized gesture. The unique jerky up and down movements defined at the wrist location makes it easier to recognize.

A distinctive observation from the confusion matrices is the similarity between the ‘Lift to Ear’ and the two ‘Lift to Mouth’ gestures. All the three gestures have similar motion of the lower arm moving towards the facial region. This similarity in the movement can be attributed to the poor performance. This also indicates that having information about the object of interaction such a phone, spoon and a cup, can help in differentiating the different ‘lift lower arm’ gestures.

Another set of confused gestures are ‘Pour’, ‘Scoop’ and ‘Spread’. These gestures are defined by rotation of the wrist while interacting with a spoon, knife or a cup. It was observed that these rotational movements were very similar for ‘Pour’ and ‘Scoop’, along with certain movements in ‘Spread’ gesture. All the movements involved an abduction of the wrist and the lower arm. It was also observed that movement information captured from the wrist was most important in

distinguishing the different gestures, with the upper arm sensor being the least important. The overall accuracy for recognizing the gestures dipped from 85% for the wrist sensor to 79% for the lower arm sensor to 61% for the upper arm sensor.

It is worthwhile to note that these subjects performed these gestures interacting with objects of different size and shape. For example, certain subjects preferred to unscrew the small cap of the water bottle, while other subjects chose to unscrew the wide lid of powder drink jar. The accuracies obtained for these different gestures even with these variations certainly demonstrates the feasibility of using wearable sensors for recognizing activity gestures when no other sensor data is available.

### 4.6 Conclusion

The activity gesture datasets presented in this chapter, though limited to a finite number of activities performed in a constrained scenario, represent a wide range of different hand –object interactions occurring in complex activities of daily living. These gestures symbolize different spatio-temporal variations and pose a significant problem for automatic recognition. The performance of different discriminative, generative and template based classification approaches for recognizing these gestures was studied on different datasets. The results indicate a superior performance of discriminative classifiers trained on statistical features extracted from the accelerometer data over generative and template-based approaches. The case study 1 (CS1) dataset resulted in an average subject independent accuracy of 90% and the case study 3 (CS3) dataset consisting of a larger vocabulary of gestures also yielded an accuracy of 91%. This establishes the feasibility of accelerometer based activity gesture recognition.

This chapter also presented an alternate data capture methodology for collecting sufficient number of samples for training using the scenarios presented in case study 1 and 3 (CS1 and CS3). Classifiers that were trained on CS1 were used to evaluate the performance on data collected in CS2. While there was a drop in the recognition accuracy for algorithms trained only on CS1, classifiers trained on a mix of both CS1 and CS3 data showed better performance. This indicates that there is merit in using auxiliary samples for learning. However, better learning strategies have

to be devised to further improve the performance. These are discussed in Chapter 6, which presents a transfer learning approach for learning from auxiliary data samples. While this chapter only discusses the problem of isolated recognition of gestures, the next chapter presents techniques for continuous recognition of these gestures in the presence of out-of-vocabulary samples.

Chapter 5

# ACTIVITY GESTURE SPOTTING

## 5.1 Introduction

The previous chapter demonstrates the feasibility of using acceleration from body worn sensors to recognize activity gestures part of complex activities of daily living. The study assumes that the gestures were a priori segmented from the continuous stream. The gesture segments can be extracted using other sensors such as RFID tags or audio. However, in the absence of any other data one has to rely on the acceleration information alone for spotting the relevant gestures from the continuous stream. This challenging problem is often termed as gesture spotting: the task of detecting the start and end points of the gestures from continuous data and recognizing the detected gestures from a pre-defined vocabulary. The difficulty of spotting gestures stems from the fact that gestures defined in the vocabulary (valid gestures) occur sporadically in a continuous gesture stream, interspersed with other hand movements: gestures not defined in the vocabulary (invalid/out-of-vocabulary gestures). These invalid gestures cannot be explicitly learned, as there is no one set of samples that characterize them. Gesture spotting approaches typically rely on models learned from valid gesture samples to overcome this problem.

This chapter discusses the problem of detecting out-of-vocabulary label samples from models learnt only on in-vocabulary labeled examples. It proposes novel techniques based on boosting and one class support vector machine for detecting invalid or out-of-vocabulary samples. It demonstrates the effectiveness of these models in the context of activity gesture spotting. On a larger note, this chapter makes the following contributions:

- It presents novel discriminative techniques for detecting out-of-vocabulary samples, by learning the decision boundaries using only valid samples. While these techniques are used here in the context of gesture recognition, they are easily applicable to other domains as well as illustrated using other benchmark datasets.
- It discusses a gesture-spotting network that meets the needs and constraints of activity recognition in wearable and pervasive systems such as the presence of invalid gestures,

large variability in the way the gestures are performed and varying length of the gestures. The gesture-spotting network uses discriminative models, thereby saving on computationally intensive operations such as matching over large data sets, without compromising on the accuracy of spotting and recognition.

The chapter begins with a brief review of approaches in the literature for gesture spotting in Section 5.2. The next section (Section 5.3) describes the proposed approaches for modeling out-of-vocabulary samples. It discusses the multiple one class SVM and AdaBoost based threshold models, with briefly discussing the large margin Gaussian mixture models. It also presents a comparative study of the different approaches for detecting out-of-vocabulary sample on standard benchmarking datasets. Section 5.4 then describes the proposed gesture-spotting network. The results of the gesture-spotting network using AdaBoost, SVM and HMM as classification models are discussed in Section 5.5. Section 5.6 concludes the work described in this chapter.

**5.2 Related Work**

Pattern spotting refers to the task of detecting the occurrence of a specific pattern in a continuous stream of data without prior knowledge of its start or end point. In contrast to isolated activity/gesture recognition, spotting task is more challenging. The difficulty of spotting specific human motion events stems from a number of sources such as co-articulation, where consecutive activities or gestures influence each other, as well as intra- and inter-person variability. Another challenge, the system has to deal with, is the fact that the motion events to be spotted may only occur sporadically, in a continuous data stream, while at the same time being embedded into other, partly arbitrary movements. These movements, however, are inherently difficult to model, due to their complexity and unpredictability. As a consequence, conventional recognition schemes for continuous classification, such as hidden Markov models (HMMs), are not directly applicable to the spotting task. Hence, one cannot take advantage of the implicit data segmentation capabilities that HMMs provide.

Approaches for spotting activities and gestures can be divided into two categories based on the assumption of the presence of out-of-vocabulary activities or gestures (invalid activities or

gestures). These movements typically occur in between two valid movements and represent either the transition between the valid movements or are completely arbitrary movements. A review of the spotting approaches in the two categories is presented in the following two sub-sections:

### 5.2.1 Spotting without invalid samples

The approaches that fall into this category assume that there are no arbitrary movements in between two valid patterns. However, the transition between the two patterns is unique and can be detected using some technique, thus providing the start and end points of the patterns. Kahol et al. [72] propose a gesture segmentation algorithm which employs a hierarchical layered structure to represent the human anatomy. The algorithm used low-level motion primitives to characterize motion captured through a Vicon camera system in the various layers of this hierarchy and was able to predict the segmentation boundaries. Wang et al [69] present an approach for automatically segmenting sequences of natural activities into atomic sections and clustering them. The segmentation was based on finding the local minimum of velocity and local maximum change in direction. The minimum below and the maximum above the certain threshold were selected as segment points. Liang et al [92] propose a temporal segmentation technique based on the discontinuity of the movements according to four gesture parameters and use HMMs to perform real-time continuous gesture recognition of sign language.

Morguet et al [93] discuss a two-step approach for continuous recognition of gestures in video sequences. The first step consists of a simple segmentation algorithm to identify the start and end points of potentially meaningful segments using a fixed threshold on a specific motion parameter, in conjunction with simple rules to obtain valid segments. The second step was used to classify the segments as one of the valid gestures. Cutler and Turk [94] use the characteristics of motion blobs estimated from optical flow for segmenting gestures from video streams. A rule-based technique is then employed to classify the segments as one of the relevant gestures. Stefanov et al [95] propose a novel approach for hand shape(gesture) recognition in continuous video streams using variable length Markov models to represent the high level structure and temporal ordering of gestures. The motion trajectories are calculated using an annealed particle filter tracking.

Nayak et al [96] propose a continuous space model for recognizing gestures. An unsupervised approach is used to extract and learn models for continuous basic units of gestures called as *gestemes*. Yang et al [97] present an enhanced Level Building algorithm for recognizing hand gestures from continuous data streams. A dynamic programming approach was used with the Level Building algorithms to spot the gestures. Gao et al [98]describe a technique for recognizing Chinese sign language gestures from continuous data obtained from a data glove. A DTW based dynamic programming technique was used to match the input data and templates of different gestures. It was assumed that the interval between two gestures is always similar and this boundary can be explicitly modeled. Jafari et al [15] discuss a technique for spotting ambulatory movements from continuous accelerometer data streams by explicitly modeling the transitions between the different movement patters using HMM. Morency et al [85] describe an algorithm for detecting and identifying head and eye gestures from continuous video streams using a latent-dynamic CRF model, thus combining the segmentation and recognition processes into one step.

### 5.2.2 Spotting with invalid samples

Spotting valid movements in a continuous data stream in the presence of invalid movements is a challenging task. The main problem lies in developing a model to characterize the invalid movements, through training samples available only for valid movements. The most common approach is to classify a test sample as a valid class if the probability of it belonging to that class is greater than a fixed threshold as demonstrated by Krishnan et al [53]. However a fixed threshold will not work at all times, as the highest probability of a sample belonging to class varies with samples. The challenge then lies in trying to determine an adaptive threshold for a sample to belong to the invalid class.

Deng and Tsui [99] proposed a method for spotting gestures in continuous data. Their approach makes use of a hidden Markov model to derive an accumulation score that is in turn used for endpoint detection of a particular gesture in a continuous data stream. The corresponding starting point for the end point is computed using the Viterbi algorithm. Lee and Kim [100] developed a HMM based gesture spotting network in video sequences. The internal segmentation property of

the HMM was used to derive an artificial HMM that acts as the threshold model. This model consists of states combined from all the individual HMMs with appropriate transitions. Since the number of the states in the threshold model can be very high depending on the number of individual HMMs, a method for reducing the number of states using mutual information metric for computing the similarity between the states was also proposed. These individual HMMs are then used to construct a circular gesture-spotting network, which is used in tandem with the Viterbi algorithm for determining the start and end points of the gestures.

A discriminative approach to sign language spotting in video streams using conditional random fields (CRF) is proposed by Yang et al[84]. This approach is similar to the HMM based threshold model proposed by Lee et al[100]. While multiple HMMs are learnt in [100] one for each of the valid class, [84] construct a single CRF, with each node corresponding to one of the valid class. Feature functions extracted from the trajectory information are used to train the CRFs. A CRF threshold model (T-CRF) is constructed by adding a label for the invalid sign patterns using the weights of the feature functions of the original CRF. Thus by retaining the structure of the CRF, [84] spot sign language gestures from continuous streams. In order to accommodate for gestures of varying length, an SVM model for the short sign detection is incorporated into the framework. Yang and Sarkar [101] propose an ASL spotting method on video streams using CRF by extracting key frames for training co-articulatory movements from sentences consisting of in-vocabulary signs and invalid sign patterns.

While the above described approaches make use of the implicit segmentation capabilities of HMM and CRF, using an explicit segmentation step to facilitate spotting is another approach proposed in the literature. Lee et al[102] developed a system for online gesture recognition using HMMs. Segmentation was used a pre-processing step to gesture recognition. Segmentation was carried out by selecting appropriate acceleration thresholds, on velocity data with the assumption that there exist short pauses between two consecutive gestures. Junker et al[32] propose a sliding window and bottom up algorithm (SWAB) for segmenting activity gestures from continuous accelerometer and gyroscope data. The continuous data streams pass through a pre-selection stage, which

localizes and preselects sections in the continuous signal stream, likely to contain relevant motion events. These candidate sections are then passed on to the classification stage and are classified in isolation using HMM based classifiers.

### 5.3 Threshold models for detecting out-of-vocabulary samples

In this section two threshold models based on Adaptive Boosting and One-Class SVM's are presented. Prior experiments have shown promising performance by AdaBoost and SVM. Having a uniform model for detecting both valid and invalid gestures is beneficial as it reduces the training time and is computationally efficient. The multiple one class SVM model is first presented followed by the AdaBoost based threshold model. The performance of these models is compared against large margin a Gaussian mixture model [181] that is described in subsection 5.3.2.

#### 5.3.1 Multiple One-Class SVM based threshold model

##### 5.3.1.1 One-Class SVM

One-class SVMs (OCSVM)[103] are commonly used in the area of anomaly detection [103], where only samples belonging to the normal class are available for training. The basic idea behind one-class SVM is to determine a function $f$ that characterizes a `small' region in the feature space that contains most of the data points by taking `+1' in that region and `-1' elsewhere. This is performed by assuming that the origin of the space falls in the `-1' region and by determining the boundary that maximizes the margin between the data points and the origin.

Formally, let $x_1, \cdots, x_n$ be a set of points all of which are considered to be a part of the `+1' region. The following quadratic program is solved for determining the required function $f$:

$$\min \frac{1}{2}\|w^2\| + \frac{1}{\nu l}\sum_{i=1}^{n}(\xi_i - \rho)$$

$$s.t.\ (w \cdot x_i) \geq \rho - \xi_i, \qquad \xi_i \geq 0$$

Here, the slack variables $\xi_i$ are introduced to allow some data points to lie outside the separating boundary, and the parameter $\nu$ controls the trade off between reducing the number of errors and

making the regularization term $\|w^2\|$ small. Using the dual representation of the Lagrangian, the objective function is equivalent to

$$\min \frac{1}{2} \sum_{i,j=1}^{n} \left(\alpha_i \alpha_j \langle x_i \cdot x_j \rangle\right),$$

$$s.t.\ 0 \leq \alpha_i, \alpha_j \leq \frac{1}{vl}, \sum_{i=1}^{n} \alpha_i = 1.$$

This quadratic programming (QP) problem can be solved using standard optimization techniques. The value of the function $f(x) = sign(w \cdot x - \rho)$ determines if the point $x$ is an outlier or a valid data sample.

$$f(x) = sign(w \cdot x - \rho)$$

$$x \in \begin{cases} +1 & if\ f(x) \geq 0 \\ -1 & if\ f(x) < 0 \end{cases}$$

Typically the training data are mapped into a feature space for obtaining a better data description through a kernel function and the optimization formulation is changed using the kernel trick, by replacing all the inner products $\langle \cdot, \cdot \rangle$ by the kernel function $k(\cdot, \cdot)$. In this work, due to the complex nature of the data, we use a Gaussian Radial Basis Function (RBF) kernel defined as follows:

$$k(x_i, x_j) = exp(-\lambda^2 \|x_i - x_j\|^2).$$

where $\lambda$ is a scaling factor that controls the width of the kernel function.

The magnitude of the decision function $f$ is typically associated with the margin of classification and can be converted to a probability value by fitting a logistic sigmoid to it. This value thus obtained indicates the probability by which a point can be considered as a normal point.

$$P(+1/x) \approx \frac{1}{1 + e^{-\phi f(x)}}$$

$$P(-1/x) \approx 1 - P(+1/x).$$

where $\phi$ is an empirically determined constant.

#### 5.3.1.2 Multiple One-Class SVM

An OCSVM may not suffice for characterizing the smallest region containing the data, especially when dealing with sparse data from multiple classes. To overcome this problem we propose to use multiple one-class SVMs, one for each class, and combine their outcomes to detect points that do

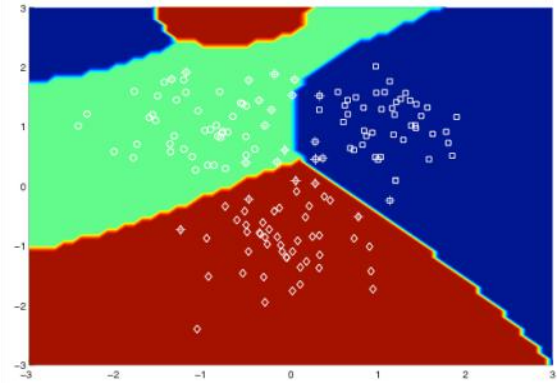


Figure 27: Decision contours for a resular SVM

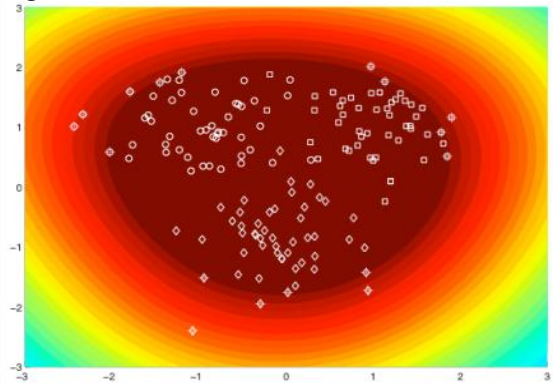


Figure 28: Decision contours for a one class SVM model (OCSVM)

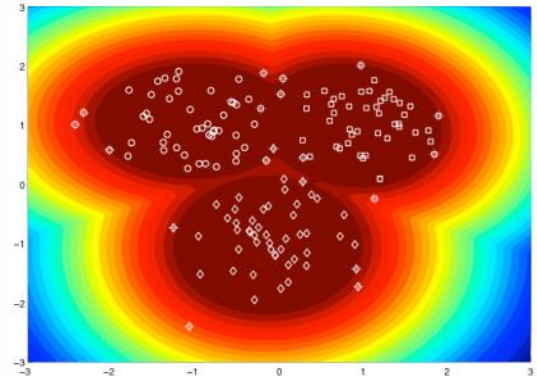


Figure 29: Multiple one class SVM (MOC-SVM) Decision surface obtained on synthetic data - three Gaussians centered at [1 1], [-1 1] and [0 -1] denoted by the circles, squares and diamonds respectively.

not belong to any of the classes. The probability of classifying a sample x as an outlier is obtained by fusing the probabilities from each of the OCSVM by taking maximum, minimum or average of these probabilities. The actual fusion technique can be determined based on the empirical performance.

To illustrate the effectiveness of MOCSVM, consider the decision contours depicted in Figure 27Figure 28Figure 29from SVM, OCSVM and MOCSVM for a three class scenario. The three classes are modeled as Gaussian distributions with mean [1 1], $[-1.1]$ and $[0, -1]$ respectively. The SVM package developed by Canu et al[104] was used for the obtaining the decision surfaces. Default parameters were used for SVM, while the parameters for OC-SVM and MOC-SVM was set ($\nu = 0.15$) to ensure the number of outliers detected while training was minimum.

It is evident from Figure 27 that the binary SVM based multi-label classifier partitions the entire space into regions corresponding to the different classes. It does not try to determine the smallest region that contains samples from each of the classes. This surface suffers from the problem of throwing up a high percentage of false positives and thus is not suitable for identifying samples as belonging to none of the classes. In contrast, both the OCSVM and MOCSVM offer better decision surfaces for detecting points that do not belong to any of the classes. However, the OCSVM model does not truly characterize the distribution of the data from the different classes as illustrated in Figure 28. The decision contours derived for OCSVM are relatively loose in nature when compared to MOCSVM depicted in Figure 29 resulting in a larger region that encompasses the samples from all the classes. The proposed MOCSVM yields a superior decision contours for separating the valid samples and invalid samples.

### 5.3.2 AdaBoost based threshold model

Before discussing the threshold model derived from AdaBoost, a brief overview of the boosting framework proposed by Freund and Schapire is presented as it forms the basis of the proposed work. AdaBoost, short for Adaptive Boosting is a meta algorithm that can be used in conjunction with any other learning algorithm to improve their performance. Freund and Schapire [105] originally proposed this algorithm for the online learning task. Subsequently, the effectiveness of

the algorithm made it hugely popular among the machine learning community by the work of Viola and Jones [183] that utilized this framework for face detection.

#### 5.3.2.1 Adaptive Boosting

Algorithm in Figure 30 illustrates the boosting algorithm as proposed in the seminal work of Freund et al. [105]. A weak hypothesis is learnt at every iteration that results in a weighted error less than 1/2. These weak hypotheses are then linearly combined in a weighted manner resulting in the final strong classifier. For a multi-class scenario, multiple binary AdaBoost classifiers are

**Algorithm 2** AdaBoost

**Input** Given a set of $N$ labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$,distribution $D$ over the $N$ examples, an integer $T$ specifying maximum number of iterations and a weak learner **Weak Learn**

**Initialize** the initial weight vector: $w_i^1 = D(i)$ for $i = 1, \ldots, N$.

**For** $t = 1, \ldots, T$

1. Set $p^t = \frac{w^t}{\left(\sum_{i=1}^{N} w_i^t\right)}$.
2. Call **Weak Learn**, providing it with the distribution $p^t$. Then, get back a hypothesis $h_t : X \to Y$(or $[0, 1]$).
3. Calculate the error of $h_t$:

$$\epsilon_t = \sum_{i=1}^{N} p_i^t \left| h_t(x_i) - y_i \right|. \tag{5.1}$$

4. Set $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
5. Update the new weight vector:

$$w_i^{t+1} = w_i^t \exp\left(-\alpha_t y_i h_t(x_i)\right) \tag{5.2}$$

**Output** the final classifier

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{5.3}$$

Figure 30: The AdaBoost algorithm as described by Freund et al [136]

learnt in a one class against all fashion. A test sample is passed through all the binary classifiers and the class associated with the maximum margin is chosen as the label of the sample.

Freund et al [105] demonstrate that AdaBoost maximizes the probability of classification and the probability values can be calculated by fitting a logistic sigmoid to the margin.

$$P(c/x) = \frac{1}{1 + e^{-\phi H_c(x)}}$$

where $H_c(x)$ is the margin of classification of sample $x$ as label $c$ and $\phi$ is an empirically determined constant.

**5.3.2.2 Threshold model**

Most of the existing gestures spotting techniques employ a fixed threshold for discriminating between a valid and an invalid gesture. However it is difficult to select a fixed threshold that is effective for all the samples. This problem is addressed by proposing an adaptive threshold model (ATM) based on the individual AdaBoost classifiers. The ATM is used to obtain a threshold for each test sample. This ATM is constructed without any additional learning, but by using some of the steps described in the AdaBoost algorithm described in Figure 30.

As described in the Section 5.3.2.1, AdaBoost learns weak hypothesis that is aimed towards classifying samples of a particular class. Some of the weak hypotheses that are learnt are quite general that they are applicable to samples from other classes as well. This principle is used for deriving all the hypotheses that satisfy a majority of samples from all the classes. These hypotheses can be considered to be the weakest among the ones that are learnt and model the most general characteristics of the space of the gestures (samples) which hold good even for the invalid gestures (samples). However, the probability of classification of a valid gesture (sample) by the correct class model will be higher than that of the ATM since there are more hypotheses trained specifically for that class.

Thus the objective is now to select the weakest among the weak hypotheses. To achieve this, samples belonging to all the classes are considered as positive samples and tested against each of the weak hypotheses contained in AdaBoost models for every class and retain those that give an error less than chance. Formally, if $\Gamma = \{h_1, h_2, \cdots, h_{CT}\}$ is the set of the hypotheses from the AdaBoost models learnt for every class ($C$ is the number of labels and $T$ is the maximum number of hypothesis per AdaBoost model) and $\{x_1, x_2, \cdots, x_N\}$ are samples from all the classes, the hypothesis contained in the ATM is defined as:

$$\Gamma_{ATM} = \left\{ h_k \epsilon\, \Gamma : \ \varepsilon_k < \frac{1}{2} \right\}$$

where $\varepsilon_k = \frac{1}{N}\sum_{i=1}^{N} |h_k(x_i) - 1|$. Each of the hypothesis $h_k$ in $\Gamma_{ATM}$ is given a weight $\alpha_k = \log\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right)$ as described in the original AdaBoost algorithm. The final hypothesis is defined as:

$$H(x) = sign\left( \sum_{k=1}^{T_{ATM}} \alpha_k \, h_k(x) \right)$$

where $T_{ATM}$ is the size of $\Gamma_{ATM}$ and the probability of classifying a sample as an invalid gesture is

$$P(IG/x) = \frac{1}{1 + e^{-\phi\, H_(ATM)(x)}}$$

where $H_{ATM}(x)$is the margin of classification derived out of the threshold model.

Figure 31 illustrates the decision surface obtained by decision stump based AdaBoost on the sample dataset presented in Figure 32. The blue regions in the image indicate regions that have low probability of containing valid data samples. The probability increases as it gets closer to the actual regions containing the valid data points from the three classes. The shape of the red regions can be attributed to the weak learner - the decision stump.

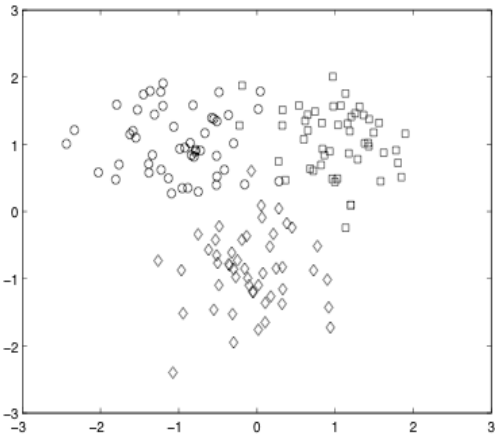



Figure 31: The distribution of samples belonging to the three classes

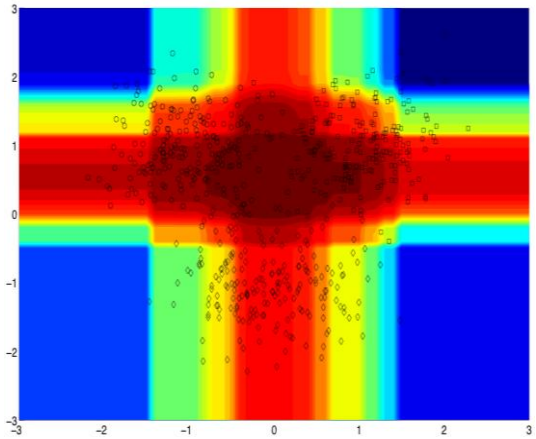



Figure 32: Illustration of the decision boundary for the valid sample as obtained by AdaBoost based threshold model on a three-label classification scenario.

### 5.3.3 Large Margin Gaussian Mixture Model

Gaussian mixtures that model the distribution of data are commonly used as a generative approach for classification. It assumes that the data can be defined in terms of a mixture of Gaussians and the learning task involves determining the parameters that define the Gaussians namely: the prior, centroid and the co-variance. Maximum-likelihood (ML) estimation is popularly used for estimating these parameters, given a training set. The output of these models i.e., the probability of a sample belonging to a particular class, can be directly used to detect out-of-vocabulary samples. However, off late it has been widely recognized that the ML estimation of the GMM does not result in optimal parameters for the purpose of classification. Many approaches have been proposed for a discriminative estimation of the GMM parameters such Sha et al [79][183][184].

Recently, Sha et al [79] proposed a framework for large margin classification by GMM. The approach is based on the idea of margin maximization similar to the SVM approach. The framework trains GMM that maximize the Mahalanobis distance of labeled examples from the decision boundaries that define competing classes. The parameters are determined through a convex optimization function. The basic difference between SVM and large margin GMM is the assumption on the type of separating surfaces. SVM assumes a hyper plane to separate samples belonging to different labels, while large margin GMM assumes hyper ellipsoids to contain the different labels. The next few paragraphs briefly describe the large margin GMM derived in this manner. For the detailed analysis of this approach, the reader is requested to refer to [79]

In the simplest scenario, one can assume each class to be represented as a single ellipsoid in the input space. The ellipsoid for each class $c$ is parameterized by a centroid vector $\mu_c \in \mathcal{R}^D$ and a positive semi definite matrix $\Sigma_c \in \mathcal{R}^{DxD}$ that determines the orientation. This refers to the mean and covariance parameters of the Gaussian model. Also associated with each class a non-negative scalar offset $\theta_c \geq 1$ that corresponds to the prior class probabilities. The decision rule labels an example $x \in \mathcal{R}^D$ by the class whose centroid yields the smallest Mahalanobis distance:

$$y = arg \min_c \{(x - \mu_c)^T \Sigma_c^{-1} (x - \mu_c) + \theta_c\}$$

Sha et al, reparameterize this equation yielding in a simpler expression in terms of a single enlarged matrix $\boldsymbol{\phi_c} \in \mathcal{R}^{D+1xD+1}$:

$$\boldsymbol{\phi_c} = \begin{bmatrix} \Sigma_c^{-1} & -\Sigma_c^{-1}\mu_c \\ -\mu_c^T\Sigma_c^{-1} & \mu_c^T\Sigma_c^{-1}\mu_c + \theta_c \end{bmatrix}$$

Thus the decision rule described in the previous equation can be re-written as

$$y = \arg\min_c \{\boldsymbol{z^t \phi_c z}\} \, where \, \boldsymbol{z} = \begin{bmatrix} x \\ 1 \end{bmatrix}$$

The argument on the right hand side of the decision rule in this equation is linear in the parameters $\boldsymbol{\phi_c}$. Thus in this transformed representation of the decision rule, the goal is to learn the elements of s single $\boldsymbol{\phi_c}$ for each class of labeled examples.

In the large margin GMM, the condition on matrices $\boldsymbol{\phi_c}$ is that all the samples in the training set are correctly classified by a large margin. In other words, samples belonging to one class are situated far from the boundaries defining other competing classes. Sha et al, express this condition in terms of the Mahalanobis distances as:

$$\forall c \neq y_n \qquad z_n^t \phi_c z_n \geq 1 + z_n^t \phi_{y_n} z_n$$

This means that for competing classes $c \neq y_n$, Mahalanobis distance (plus offset) to the $c^{th}$ centroid exceeds the Mahalanobis distance (plus offset) to the target centroid by a margin of at least one unit.

A convex optimization akin to the SVM optimization that selects the "smallest" parameters that satisfy the large margin constraints is proposed. In this case, the optimization is an instance of semi definite programming:

$$\min \sum_c trace(\Sigma_c^{-1})$$

$$s.t. \; 1 + z_n^t (\phi_{y_n} - \phi_c) z_n \leq 0 \quad \forall c \neq y_n, n = 1, 2, \cdots N$$

$$\phi_c \succ 0, \qquad c = 1,2, \cdots C$$

It can be noted that the trace of the inverse covariance matrix $\Sigma_c^{-1}$ alone is considered in the minimization function. Sha et al, choose this instead of the $\phi_c$ matrix to prevent the regularization of the mean vectors as well. The last constraint of the optimization problem $\phi_c \succ 0$ restricts the

matrices to be positive semi definite. As in the case of SVM, nonnegative slack variables are introduced to control the amount by which the margin constraints are allowed to be violated. The objective function in this setting balances the margin violations versus the scale regularization:

$$\min \sum_{nc} \xi_{nc} + \kappa \sum_{c} trace\left(\Sigma_c^{-1}\right)$$

$$s.t.\ 1 + z_n^t\ (\phi_{y_n} - \phi_c) z_n \leq \xi_{nc},$$

$$\xi_{nc} \geq 0 \quad \forall c \neq y_n, n = 1, 2, \cdots N$$

$$\phi_c \succ 0, \qquad c = 1,2, \cdots C$$

The balancing parameter $\kappa$ is set through cross-validation.

Sha et al, extend this formulation for multiple mixture components, where each class is represented by multiple ellipsoids. Let $\phi_{cm}$ denote the matrix for the $m^{t\acute{h}}$ ellipsoid or the mixture component in class $c$. Associated with each sample $x_n$ is not only a class label $y_n$, but also a mixture component label $m_n$ indicating the mixture component that is most likely to generate the sample $x_n$. Though these mixture component labels are not provided in the training data, Sha et al, suggest generating these labels by fitting GMMs to each class through maximum likelihood estimation.

In this setting where each class is represented by multiple ellipsoids, the goal of learning is to ensure that each example is closer to its target ellipsoid than the ellipsoids from all other classes. Thus the original constraint is replaced by a set of M (number of mixture components) constraints:

$$\forall c \neq y_n \qquad z_n^t \phi_{cm} z_n \ \geq 1 + \ z_n^t \phi_{y_n m_n} z_n$$

Sha et al, transform these multiple constraints into a single constraint using the 'softmax' inequality resulting in

$$\forall c \neq y_n \qquad -\log \textstyle\sum_m e^{-z_n^T \phi_{cm} z_n} - \ z_n^T \phi_{y_n m_n} z_n \ \geq 1$$

This constraint is used in the optimization problem to derive the $\phi_{cm}$ matrices. We do not delve into the numerical optimization of this minimization problem and request readers to refer to Fei Sha's thesis for the specific details. In their implementation, Sha assumes a full covariance matrix. However for simplicity and for providing numerical stability to the optimization problem, the

experiments conducted in this work assume only a diagonal covariance matrix. This change is reflected in our modification of Sha's implementation which was targeted towards speech recognition. The output of the large margin GMMs are nothing but the negative log likelihoods. These can be easily converted to likelihoods for determining the probability of a sample belonging to a particular class.

### 5.3.4 Comparison of Out-of Vocabulary sample detection models

Typical machine learning and pattern recognition algorithms assume that the test data points are sampled from the class labels presented while training. It is hard to find natural datasets that contain out-of-vocabulary samples in the test set. The efficiency of the different techniques for detecting out-of-vocabulary samples has been experimented with 6 simulated datasets. Five of these datasets were selected from the UCI machine-learning repository [186] that is commonly used for benchmarking the performance of different algorithms and the sixth dataset was the activity gesture dataset corresponding to CS3 described in the previous chapter.

Three partitions were created for each of the dataset corresponding to the training, validation and test sets. The training set consisted of only in-vocabulary samples. The validation set consisted of equal amounts of in and out-of vocabulary samples. This set was used for empirically estimating the best learning parameters for the different algorithms. The performance of the algorithms was compared on the output of the test set that consisted of equal number of samples from the in and out-of vocabulary. The description of the partitions for each of the dataset is presented in Table 17. The Yeast dataset [185] has a total of 10 classes. Nearly 60% of the samples of this dataset belong to categories CYT and NUC. Categories CYT, NUC and MIT represented the in-vocabulary set and rest of the labels were a part of out-of-vocabulary set. The Scene recognition dataset [187] consists of features extracted from images representing different scenes such as tall buildings, highway etc, with a total of 8 classes (2688 images). 200 images from each class were randomly selected for the experiment conducted in this work. The gist Gabor features described by Torralba et al [188] was extracted from each of the image resulting in 512 dimensional feature vector. Categories *tall buildings, inside city, street* and *highway* were part of the in-vocabulary set

and *coast*, *open country*, *mountain* and *forest* form the out-of-vocabulary set. The USPS hand written image data [186] consists of a total of 9298 images of the hand written digits from 1-9. This dataset is commonly used for testing machine learning algorithms. Each of the images was re-sampled to 16x16 resulting in a 256 dimensional feature vector. 400 randomly selected samples from each digit were employed for the experiment conducted in this work. Digits 1,2,3,4 and 5 formed the in-vocabulary set and 6, 7 8, 9 and 10 represented the out-of-vocabulary set. The SatImage [186] is another multiclass image dataset consisting of 6 classes with a total of 4435 images. A 36 dimensional feature vector represents each image. 400 randomly selected images from each of the class resulting in a total of 2400 samples were used for the study. The Iris dataset [186] is one of the best-known dataset found in pattern recognition literature. It is a three-class problem with the objective of predicting the class of the iris plant as Setosa, Versicolor or Virgnica. Each sample is represented as in terms of the characteristics of the sepal and petal of the flower. Each class consists of 50 samples resulting in a total of 150 samples. All the samples were considered for this study. Setosa and Versicolor were part of the in-vocabulary set and Virginica was the out-of-vocabulary label. The activity gesture dataset CS3 described in the previous chapter was the last dataset considered for this study. Subject information associated with the gesture samples were ignored while creating the split. 5 gestures namely 'Comb', 'Lift to Ear', 'Lift to mouth 1', Lift to Mouth 2' and 'Mix' formed the in-vocabulary labels and the remaining 6 gestures: 'Pour', 'Scoop', 'Spread', Sprinkle', 'Stir', 'Unscrew cap' correspond to the out-of-vocabulary labels. This dataset helps to understand the ability of the out-vocabulary label detection techniques to filter irrelevant samples during gesture spotting.

Table 17: Description of the in and out of vocabulary label split for the datasets used in the study

| **Dataset** | **Yeast** | **USPS** | **Scene Recognition (OSR)** | **SatImage** | **Iris** | **Activity Gesture (CS3)** |
|---|---|---|---|---|---|---|
| **Total number of samples (Feature dimension)** | 1484 (8) | 3000 (256) | 1200 (512) | 1800 (36) | 150(4) | 940(80) |
| **In vocabulary (+)** | 1, 2, 3 | 1, 2, 3, 4, 5 | 1, 2, 3, 4 | 1, 3, 6 | 1, 2 | 1, 2, 3, 4, 5 |
| **Out-of vocabulary(-)** | 4, 5, 6, 7, 8, 9, 10 | 6, 7, 8, 9, 10 | 5, 6, 7, 8 | 2, 4, 5 | 3 | 6, 7, 8, 9, 10, 11 |
| **Training** | 567(+) | 1000(+) | 400(+) | 600(+) | 50(+) | 208(+) |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Validation** | 284(+)<br>284(-) | 500(+)<br>500(-) | 200(+) 200(-) | 300(+)<br>300(-) | 26(+)<br>26(-) | 104(+)<br>104(-) |
| **Test** | 173(+)<br>174(-) | 500(+)<br>500(-) | 200(+) 200(-) | 300(+)<br>300(-) | 24(+)<br>24(-) | 104(+)<br>104(-) |

SVM, One Class – SVM and multiple One Class-SVM were experimented with different kernel functions. Linear, quadratic and cubic polynomials and radial basis functions (Gaussian) were the kernels experimented in this study. The parameters for SVM were empirically determined through experimentation on the validation dataset. A modified implementation of the large margin Gaussian mixture model by Sha et al [79] was used for experimented LM-GMM and GMM. The number of Gaussian mixtures was fixed at 5 empirically through experimentation on the validation set. The accuracies on in-vocabulary label samples were computed for SVM implementation with different kernels, GMM and LMGMM. These values are presented in Table 18. The accuracies on all the datasets except 'Yeast' and 'SatImage' are high. This indicates that the classifiers have indeed learnt the in-vocabulary samples. Since the accuracies obtained across multiple classifiers for 'Yeast' and 'SatImage' appear to be consistent, we can assume that classifiers have learnt the maximum possible from the labeled in-vocabulary training set. It can also be noticed that large margin version of GMM does not always guarantee an improvement in the performance. This marginal improvement in the performance of LMGMM can be attributed to the fact that the large margin optimization methodology is used for estimating the diagonal variance parameters of the Gaussians. The mean vector does not get modified in the process. Furthermore since the initial parameters for the Gaussians are estimated using maximum likelihood, one cannot expect significant deviation in the final parameters after the large margin optimization.

Table 18: Accuracies on in-vocabulary label samples

| **Dataset** | **SVM** | | | | **GMM** | **LMGMM** |
|---|---|---|---|---|---|---|
| | **Linear** | **Quadratic** | **Cubic** | **Gaussian** | | |
| **Iris** | 1 | 1 | 1 | 1 | 1 | 1 |
| **AG CS(3)** | 0.94 | 0.99 | 0.99 | 0.94 | 0.91 | 0.86 |
| **OSR** | 0.93 | 0.94 | 0.93 | 0.93 | 0.87 | 0.89 |
| **Yeast** | 0.6 | 0.6 | 0.61 | 0.61 | 0.61 | 0.61 |
| **SatImage** | 0.67 | 0.67 | 0.67 | 0.67 | 0.65 | 0.66 |
| **USPS** | 0.97 | 0.94 | 0.97 | 0.97 | 0.9 | 0.96 |

The accuracies on the activity gesture CS3 dataset are very high for the SVMs. This should not be confused with the results presented in the previous chapter where subject independent accuracies SVM accuracies were presented. The accuracies presented in Table 18 can at best be termed as subject dependent since the subject information is ignored while creating the in and out of vocabulary label split. The Gaussian kernel SVM appears to yield the best accuracy when compared against the other kernels. The results obtained on the wide variety (in terms of dimensionality, size and type) of datasets considered in this study can be used to conclude the generalized performance of the different approaches on other datasets.

The performance of the different techniques for detecting out-vocabulary label samples was measured by computing the area (AUC) under the Receiver Operating Characteristic (ROC) curve. Higher values of the value of AUC represent better performance of the algorithm. The ROC curve is an ideal choice for comparison as it considers both the true positives and false positives detected by the algorithm ([207]). A good way to quantify this feature is to compute the area under the ROC curve. The ROC curves for the different algorithms on each of the datasets are illustrated in Figure 33, Figure **34** and Figure **35**. The ROC curves are relatively smooth for all the datasets except 'Iris'. This can be explained by the small size of the test samples. The AUC values obtained for the respective runs are summarized in Table 19. The algorithm that resulted in the highest AUC for each of the datasets is bold faced. The following paragraphs discuss some of the observations made from the ROC curves and the AUC values.

The first observation that one can make is the poor performance of all methods except OCSVM on the 'Iris' dataset. This can be again attributed to the small sample size. The results on this dataset are excluded from the discussions that follow. Let us begin the discussion by first comparing the performance of OCSVM and MOCSVM. The results unanimously indicate the superiority of MOCSVM over OCSVM. It testifies the ability of the multiple one class SVM to handle samples from multiple classes over a single one class SVM. It shows that MOCSVM trained on in-vocabulary label samples leads to compact regions in the data space containing in-vocabulary label samples in comparison to a single one class SVM as hypothesized and experienced in the

discussion in Section 5.3.1. In fact for some of the data sets such as 'OSR' and 'SatImage' the performance of OCSVM is almost equivalent to a coin toss, where as MOCSVM results in significantly better performance.

Focusing on the comparison between the version of the regular SVM and MOCSVM, it is interesting to note that the proposed technique that combines the output of the individual one class SVM through the max function results in best performance on all datasets except the out door scene recognition (OSR) dataset. The improvement is significant for 'Yeast', 'SatImage and 'Activity gesture CS3' datasets, while a marginal improvement is noticed for the 'USPS' dataset. This shows that explicit modeling of out-vocabulary label samples through multiple one class SVM leads to a better characterization of the data space than regular SVM that assumes the exhaustive partition of the data space into hyper planes/spheres representing the different in-vocabulary label samples. This actually validates the strong assumption made by these classifiers that the entire data space consists of only in-vocabulary label samples. Thus for problems where the classifier can expect out-of-vocabulary label samples, finding the decision surface that maximizes the margin between the in-vocabulary label samples alone will not be sufficient. The anomalous behavior of the MOCSVM on OSR dataset is could be attributed to the sparseness of the high dimensional data space. Further experiments have to be carried out to understand the behavior of OCSVM in high dimensional spaces.

It can also be observed that even though the SVM models resulted in very high performance for the in-vocabulary label classification task, there is a significant decrease when it comes to the out-of-vocabulary label sample detection. This indicates that having higher in-vocabulary sample classification accuracies does not guarantee a good performance for the classifying out-of-vocabulary label samples. However, there appears to be some correlation between the performance of SVM for out-of-vocabulary label sample detection and the accuracies on in-vocabulary classification, when the performance on the latter task is poor. All the versions of the regular SVM that resulted poor in-vocabulary label sample recognition accuracies for 'SatImage' and 'Yeast' also show poor out-of-vocabulary label sample detection for these two datasets. However,

for the USPS dataset, even though the in-vocabulary label classification has high accuracies, the corresponding out-of-vocabulary label sample detection has low AUC values. Intuitively this means that models with poor generalization performance on the in-vocabulary label classification problem will result in significantly poorer performance for detecting invalid samples.

Another interesting observation that can be made from Table 19 is on the performance of the generative classifier model GMM. The performance of both maximum likelihood based GMM and large margin GMM are significantly bad in comparison to their discriminative counter parts. It is surprising that classifiers that directly model the probability of a sample being generated by a class result in inferior performance for detecting out-of-vocabulary label samples. One reason for this result could be the fundamental Gaussian distribution assumption made by these models on data samples. The very fact that there is not significant change in the performance of the GMM trained in a generative and a discriminative manner indicates the independence of the training mechanism on detecting out-of-vocabulary sample detection. Further experiments with techniques that model other types of data distributions have to be conducted to validate this hypothesis.

Experiments were also conducted to study effect of averaging the outputs of the individual OCSVM for training a MOCSVM instead of determining the maximum value. The AUC values for the averaging MOCSVM (‘OSR’ – 0.45, ‘SatImage’ – 0.53, ‘USPS’ – 0.84, ‘Yeast’ – 0.76) were significantly lower than the values obtained for the maximizing MOCSVM, which are listed in Table 19. This indicates that averaging the probability outputs of each of the OCSVM results in an expected value of classifying the sample as out-of-vocabulary is a poor measure in contrast to the maximum for detecting the out-of-vocabulary samples.

Finally it can be noted that for the gesture recognition task that is the focus of this dissertation, the proposed MOCSVM technique results in the best performance over other approaches. While this study was conducted only on the activity gesture CS3 dataset, the following sub section, discusses the performance of the MOCSVM technique in the context of detecting invalid gesture samples by learning only on the CS1 and CS2 datasets. The performance of the proposed Adaptive Boosting

based threshold model that was not considered for this study is also discussed in the next subsection.

Table 19: Area under the ROC curve (AUC) values for the different classifiers on the selected datasets. For each dataset, the classifier with the highest AUC value has been bold faced.

| **Dataset** | **SVM** | | | | **One Class SVM** | | | | **Multiple One Class SVM** | | | | **GMM** | **LMGMM** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Kernel Type** | Linear | Quadratic | Cubic | Gaussian | Linear | Quadratic | Cubic | Gaussian | Linear | Quadratic | Cubic | Gaussian | | |
| **Iris** | 0 | 0.01 | 0.01 | 0.18 | 0.78 | **1** | **1** | **1** | 0.65 | 0.71 | 0.71 | 0.86 | 0.31 | 0.31 |
| **AG CS(3)** | 0.83 | 0.76 | 0.8 | 0.94 | 0.62 | 0.69 | 0.7 | 0.9 | 0.83 | 0.86 | 0.87 | **0.98** | 0.62 | 0.62 |
| **OSR** | 0.72 | 0.74 | **0.76** | 0.74 | 0.40 | 0.43 | 0.43 | 0.52 | 0.5 | 0.6 | 0.58 | 0.60 | 0.53 | 0.61 |
| **Yeast** | 0.55 | 0.5 | 0.5 | 0.6 | 0.75 | 0.75 | 0.76 | 0.75 | 0.74 | 0.76 | 0.76 | **0.78** | 0.30 | 0.29 |
| **SatImage** | 0.61 | 0.67 | 0.69 | 0.66 | 0.51 | 0.46 | 0.46 | 0.53 | 0.71 | 0.76 | 0.77 | **0.88** | 0.57 | 0.57 |
| **USPS** | 0.83 | 0.84 | **0.92** | 0.82 | 0.63 | 0.69 | 0.76 | 0.85 | 0.8 | 0.85 | **0.92** | 0.90 | 0.6 | 0.62 |

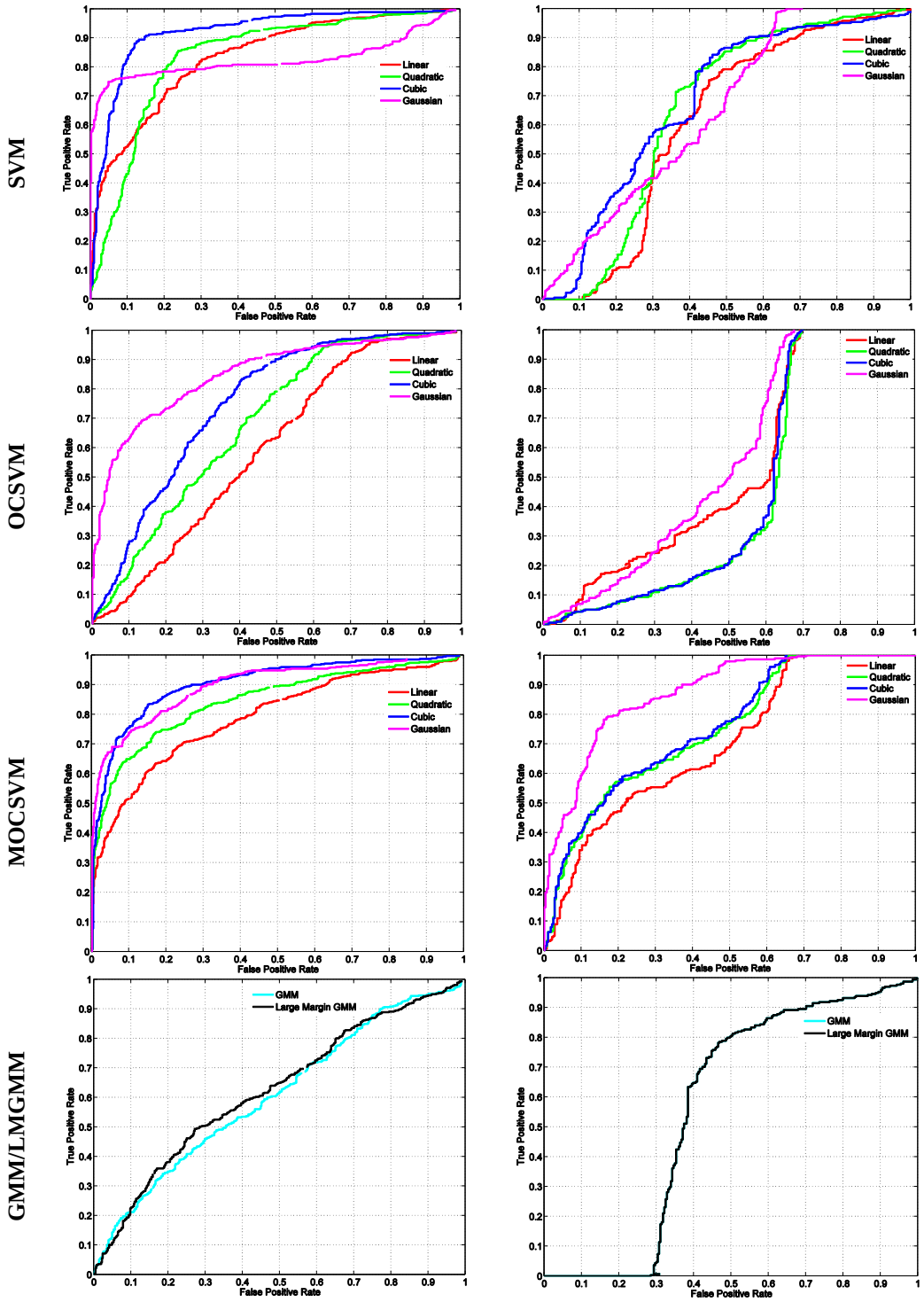



Figure 33: ROC curves for out-vocabulary label detection on USPS and SatImage datasets. Red- Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM

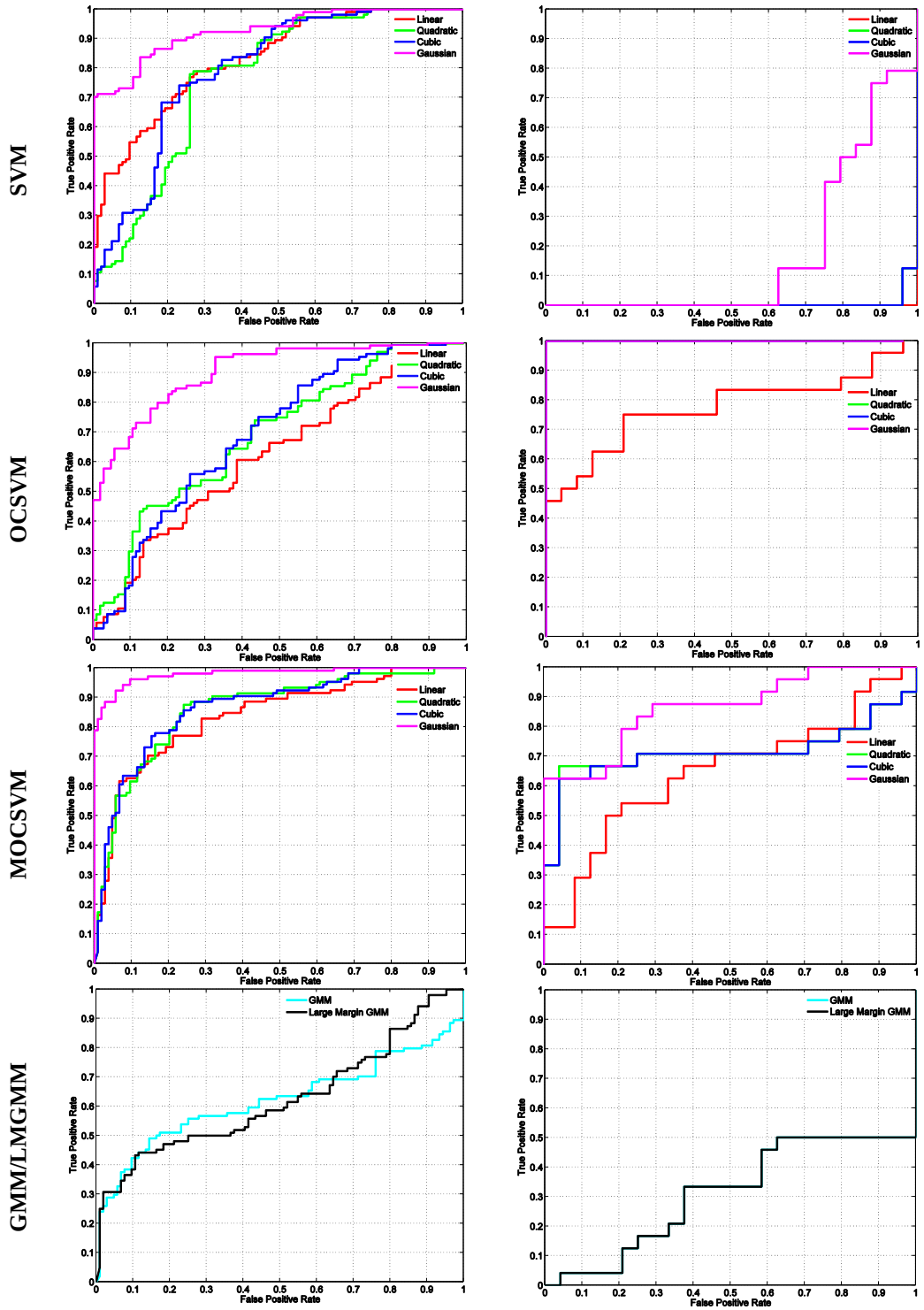



Figure 34: ROC curves for out-vocabulary label detection on AGCS3 and Iris datasets. Red-Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM

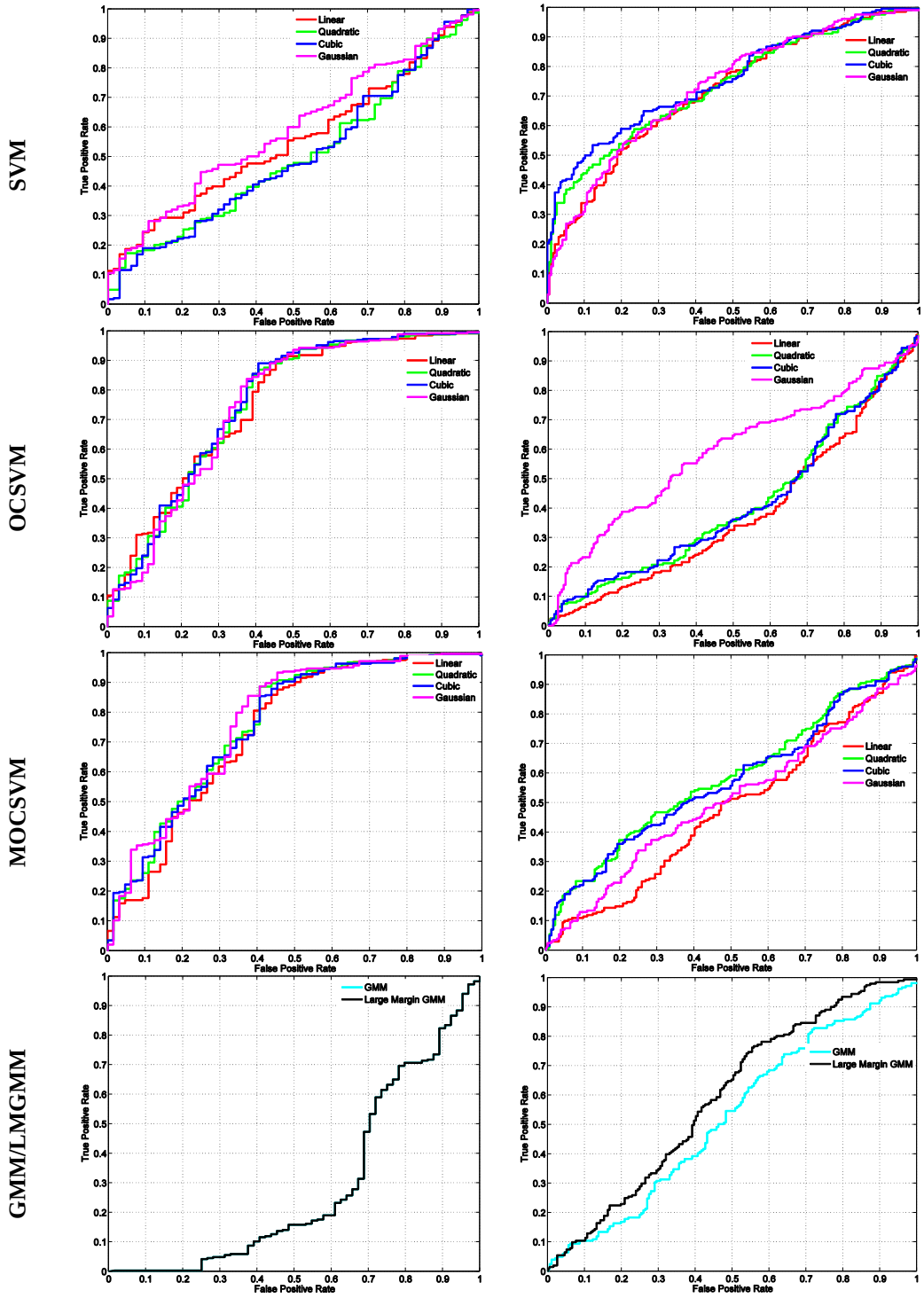



Figure 35: ROC curves for out-vocabulary label detection on Yeast and OSR datasets. Red- Linear SVM, Green - Quadratic kernel, Blue - Cubic kernel, Magenta - Gaussian Kernel, Cyan - GMM, Black - Large Margin GMM

### 5.3.5 Performance on the activity gesture set CS1

The adaptive threshold models based on boosting and MOCSVM as proposed in this chapter was evaluated with all the isolated valid gestures (from CS1). The objective was to see if the probability of classification by the actual valid gesture model is greater than the threshold model. Table 20 presents the results from this test. It can be seen that the threshold model performs as expected, by correctly classifying all the gestures. Similar results were obtained with the isolated gestures from CS2. These results demonstrate that the threshold model is good at discriminating valid gestures.

Table 20: Accuracy of the threshold models on Activity Gestures from CS1

| **Activity Gesture Label** | **Total number of Samples** | **AdaBoost based ATM** | **MOC-SVM based ATM** |
|---|---|---|---|
| **Lift to Mouth** | 95 | 0.92 | 0.94 |
| **Pour** | 95 | 0.95 | 0.96 |
| **Scoop** | 102 | 0.94 | 0.96 |
| **Unscrew cap** | 92 | 0.97 | 0.95 |
| **Stir** | 100 | 1 | 1 |

To further test the performance of the MOC-SVM based threshold model, gestures not belonging to none of the classes were used for testing. Figure 36 illustrates the results obtained from this experiment. The performance was measured in terms of the F score. It can be seen that the proposed MOC-SVM method performs better than the OC-SVM technique, for all the values of $\nu$.

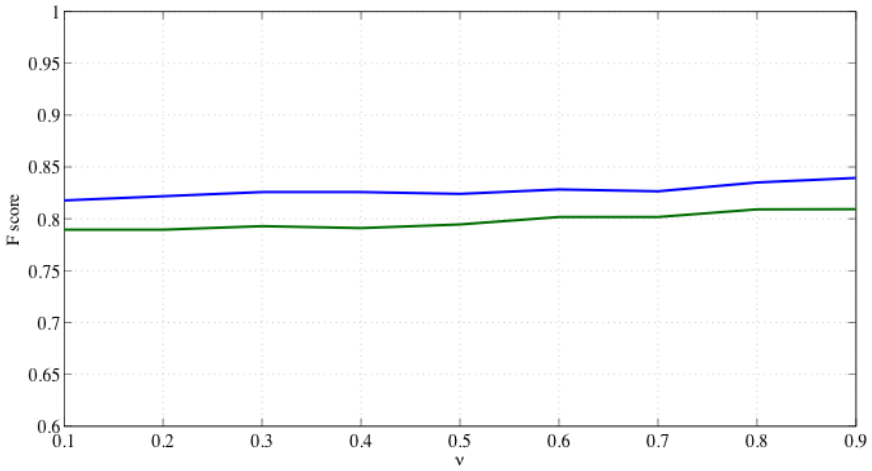



Figure 36: Illustration of the performance of OC-SVM (Blue) and MOC-SVM (Green) for detecting invalid gestures samples for varying values of $\nu$.

As discussed before, $\nu$ controls the number of outliers considered by the system. It can be observed that even for low values of $\nu$, the MOC-SVM technique performs better.

**5.4 Gesture-spotting network**

Having defined the two types of adaptive threshold models, we now focus our attention to the gesture-spotting network. For the sake of simplicity, the following notations are used. The valid gesture model for label $i$ learnt through either AdaBoost or SVM are denoted as $G_i$ and similarly the two adaptive threshold models are also denoted by $ATM$. When the gesture-spotting network consists of AdaBoost models, the AdaBoost based threshold model is used. The same holds good for the SVM based gesture spotting network.

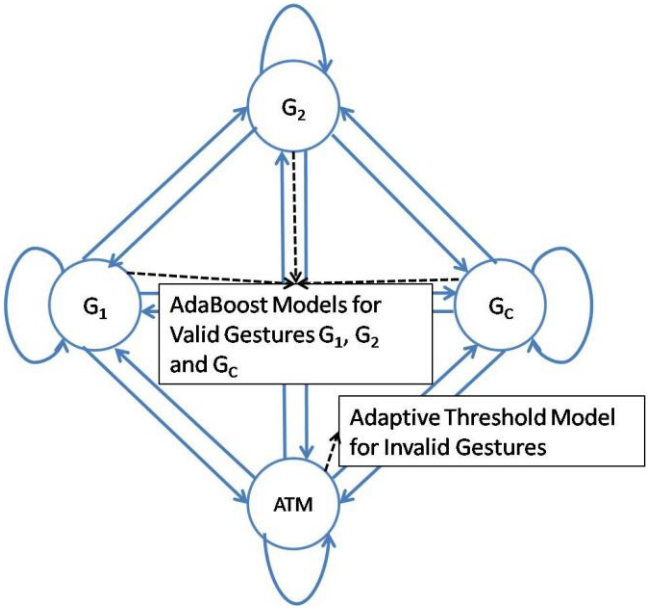



Figure 37: Gesture-spotting network

The ergodic graphical model $\{V, E\}$ proposed for gesture spotting is illustrated in Figure 37. Each of the vertices of the graph corresponds to either a valid class or the threshold model i.e. $V \in \{G_1, \cdots, G_C, ATM\}$. Given an observation, each vertex of the graph emits the probability of the observation belonging to the corresponding class derived from the individual gesture models. The objective of the gesture spotting network is to find the best label (vertex) sequence $Y_{1,t} = y_1, y_2, \cdots, y_t$ with $y_i \in V$ that generates a given observation $O_{1,t} = o_1, o_2, \cdots, o_t$; thereby finding

the start and end points of gestures embedded in the input stream. An observation at a time step is a feature vector extracted from a set of acceleration samples ending at that time step. The Viterbi algorithm [180][189] is used to uncover the state sequence, with the objective of maximizing the probability of a state sequence $Y_{1,t}$ that maximizes the observation sequence $O_{1,t}$.

We define the quantity:

$$\delta_t(i) \cong \max_{Y_{1,t}} P(Y_{1,t-1}, y_t = G_i, O_{1,t}/\lambda)$$

where $\lambda$ are the parameters of the network, with the highest probability along a single path arriving at $G_i$ at time $t$ and accounting for the first $t$ observations. Using induction it can be seen that:

$$\delta_1(j) = \pi_j b_j(O_1) \{1 \leq j \leq N\}$$

$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] \cdot b_j(O_t) \{2 \leq t \leq T, 1 \leq j \leq N\}$$

where $b_j(O_t)$ is the probability of classifying observation $O_t$ as gesture $G_j$ derived from the valid gesture model for $G_j$, $a_{ij}$ is the transition probability from gesture $G_i$ to $G_j$ and $\pi_j$ is the probability of starting from gesture $G_j$. For backtracking information, variable $\psi_t(j)$ is defined, that stores the argument that maximizes $\delta_t(j)$ for each $t$ and $j$:

$$\psi_t(j) = arg \max_i [\delta_{t-1}(i)\, a_{ij}] \{2 \leq t \leq T, 1 \leq j \leq N.\}$$

Once an end point of a gesture is determined, the most likely state sequence ($Y^* = y_1^*, y_2^*, \cdots, y_T^*$) is uncovered through tracing back to the initial state by following the Viterbi path:

$$y_T^* = G_N,$$

$$y_t^* = \psi_{t+1}(y_{t+1}^*), \{t = T-1, T-2, \cdots, 1.\}$$

Typically initially, the likelihood of invalid gesture model is the highest as is illustrated in Figure 38. However, as a forward pass comes closer to the end of a valid sign, the likelihood of the valid sign increases and crosses the value for the invalid gesture. In Figure 38, the likelihood for the `LM' gesture increases slowly at around 175ms and becomes higher than the threshold model at 275ms. All points between 275 and 430 are candidate end points (CEPs) for the `LM' gesture. For each CEP, the corresponding starting point is estimated by backtracking the Viterbi path.

The end-point detection is the process of selecting the best point to distinguish each gesture. The process is initiated when the last CEP of the current gesture is found after the preceding gesture to

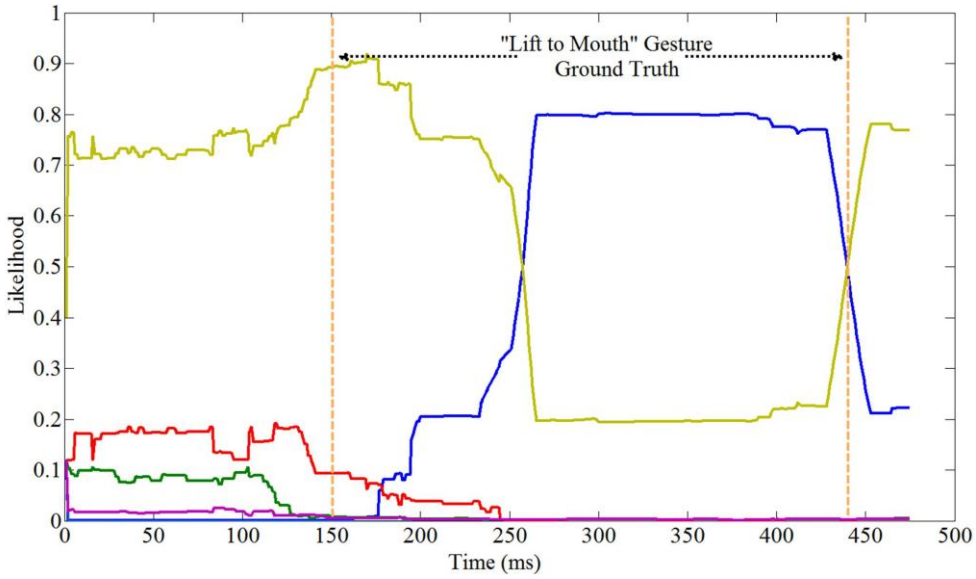



Figure 38: Evolution of the likelihood of the different gesture labels and the adaptive threshold model over time

fire or when the elapsed time steps since the last gesture are greater than a given length (ensuring duration constraint). Then the detection criterion is defined as follows:

- When the immediately following pattern is not a gesture, then the last CEP of the preceding gesture is determined as the end point.
- When the immediately following pattern is by itself a valid gesture, the CEP immediately preceding the starting point of the pattern is chosen the end point of the gesture. In this work, it is assumed that none of the gesture in the vocabulary subsumes another gesture in the vocabulary.

**5.5 Results and Discussion**

The proposed gesture spotting technique was evaluated in the context of recognizing activity gestures, from accelerometer data. Five hand gestures namely; *lift to mouth*, *pour*, *scoop*, *unscrew cap* and *stir*; that constitute the activities of making a drink and drinking were considered. Two different scenarios were set up for collecting the data. In the first scenario (CS1), subjects repeated the same gesture 20 times, with a gap of 1-3 seconds between successive repetitions. In the second

scenario (CS2), the subjects performed 4 trials of the activities: making a drink and drinking, without any explicit instructions. Data was collected from 5 subjects, with the accelerometers placed near the wrist and elbow. More details about the activity gesture data can be found in Section 4.3

Statistical features such as mean, variance, correlation, minimum, maximum and zero crossing rate were extracted from sequence of data samples. The observation for every time step is described in the form of a 45 dimensional feature vector. The features extracted such that they can be iteratively updated a new data samples are made available. Duration constraints were imposed to ensure that samples that are very old do not have an influence in the computation of the feature vector for the current time step. An AdaBoost model for every gesture label was learnt using decision stumps as the weak learner. A maximum of 100 iterations were performed to learn the models. Individual SVM models using Gaussian kernels were also learnt for building the SVM based gesture spotting network. The transition matrix for the gesture spotting network was manually designed. In future, we plan to learn this directly from the data using an expectation maximization procedure.

HMM based threshold model and gesture spotting network as described by Lee et [100] was implemented to compare its performance against the proposed techniques. The number of states for each individual HMM was determined empirically. The HMMs were trained on the sequence of feature vectors as computed using a sliding window of constant width. Left-to-right and ergodic HMM's were experimented. However, for the lack of space the results only from the best HMM model (ergodic) is presented here.

The performance of the gesture spotting network was evaluated using the precision and recall metrics:

$$precision = \frac{Recognized\ Gestures}{Retrieved\ Gestures}$$

$$recall = \frac{Recognized\ Gestures}{Relevant\ Gestures}$$

Relevant gestures are instances of the valid gestures that were actually performed by the subjects. Retrieved gestures are instances of the valid gestures that are retrieved by the gesture-spotting network. Recognized gestures are those instances of the relevant gestures that have been correctly retrieved. In addition, number of Insertions - retrieved instances that do not correspond to a relevant gesture and Deletions - relevant instances that were not retrieved, were also computed. These metrics were used to evaluate the continuous recognition.

The gesture spotting task was performed by the network based on AdaBoost and SVM as described above and using the HMM based threshold model. We used the continuous data collected from CS1 for performing this test. Each subject performed five long sequences each consisting of 15-20 gestures interspersed with other natural hand movements. A five-fold leave one out strategy was employed, to obtain the subject independent performance. Table 21, Table 22 and Table 23 present the results obtained from this test by the proposed AdaBoost, SVM and the HMM based gesture-spotting network respectively. The proposed gesture spotting networks outperform the HMM based network, both in terms of overall precision and recall. Both the AdaBoost and SVM based networks are able to spot a large number of relevant gestures from the continuous stream as indicated by the high recall rate. However, there is only a marginal difference in the performance between these two approaches. The precision and recall values for gesture ‘scoop’ was low compared to other gestures. It was observed that the network was often confusing ’scoop’ with ‘pour’, suggesting a close similarity between the gestures. The gesture ``unscrew cap’ has a very high recall but relatively low precision. The current framework performs segmentation and recognition simultaneously and there is no additional step for classifying the detected valid gesture segments. This might be a reason for the low precision values. As part of future work, we plan to implement an additional classification layer through a strong classifier such as SVM and improve the precision.

Table 21: Gesture spotting test using the AdaBoost based threshold model

| Activity Gesture | Lift to Mouth | Pour | Scoop | Unscrew cap | Stir | Total |
|---|---|---|---|---|---|---|
| **Relevant** | 95 | 95 | 102 | 92 | 100 | **484** |
| **Retrieved** | 105 | 117 | 115 | 142 | 107 | **569** |
| **Recognized** | 95 | 86 | 79 | 90 | 100 | **450** |
| **Insertions** | 10 | 21 | 36 | 52 | 7 | **126** |
| **Deletion** | 0 | 9 | 23 | 2 | 0 | **34** |
| **Precision** | **0.91** | **0.81** | **0.69** | **0.63** | **0.94** | **0.781** |
| **Recall** | **1** | **0.91** | **0.78** | **0.98** | **1** | **0.93** |

Table 22: Gesture spotting test using MOC-SVM based threshold model

| Activity Gesture | Lift to Mouth | Pour | Scoop | Unscrew cap | Stir | Total |
|---|---|---|---|---|---|---|
| **Relevant** | 95 | 95 | 102 | 92 | 100 | **484** |
| **Retrieved** | 105 | 117 | 115 | 142 | 107 | **586** |
| **Recognized** | 95 | 89 | 88 | 87 | 100 | **459** |
| **Insertions** | 10 | 7 | 49 | 61 | 0 | **127** |
| **Deletion** | 0 | 6 | 14 | 5 | 0 | **25** |
| **Precision** | **0.91** | **0.93** | **0.64** | **0.59** | **1** | **0.79** |
| **Recall** | **1** | **0.94** | **0.88** | **0.95** | **1** | **0.95** |

Table 23: Gesture spotting test using HMM based threshold model

| Activity Gesture | Lift to Mouth | Pour | Scoop | Unscrew cap | Stir | Total |
|---|---|---|---|---|---|---|
| **Relevant** | 95 | 95 | 102 | 92 | 100 | **484** |
| **Retrieved** | 176 | 193 | 186 | 181 | 250 | **986** |
| **Recognized** | 87 | 64 | 68 | 92 | 80 | **391** |
| **Insertions** | 89 | 129 | 118 | 89 | 170 | **595** |
| **Deletion** | 8 | 31 | 34 | 0 | 20 | **93** |
| **Precision** | **0.5** | **0.33** | **0.37** | **0.51** | **0.32** | **0.4** |
| **Recall** | **0.92** | **0.68** | **0.67** | **1** | **0.8** | **0.81** |

Further these models were used in the gesture spotting experiments as described for CS1. The individual gesture models were trained using data from both CS1 and CS2 (the third strategy described previously). The precision and recall values for both AdaBoost and SVM based gesture spotting network are illustrated in Figure 39. As expected, it can be noticed that the performance of both AdaBoost and SVM based gesture spotting networks are lower than their isolated recognition performance. It is interesting to note that in this `real-life' scenario data, SVM seems to be performing better than AdaBoost, consistently in all the cases. Training using data combined from both the scenarios, leads to increase in the recall values. On further analysis of the results, it was noticed that one of the gestures, `stir' was being falsely detected a number of times. This is

probably due to the movement pattern of this gesture, which is very different from the other gestures. A second level of classification either through movements or using data from other source is required to remove the false positives.

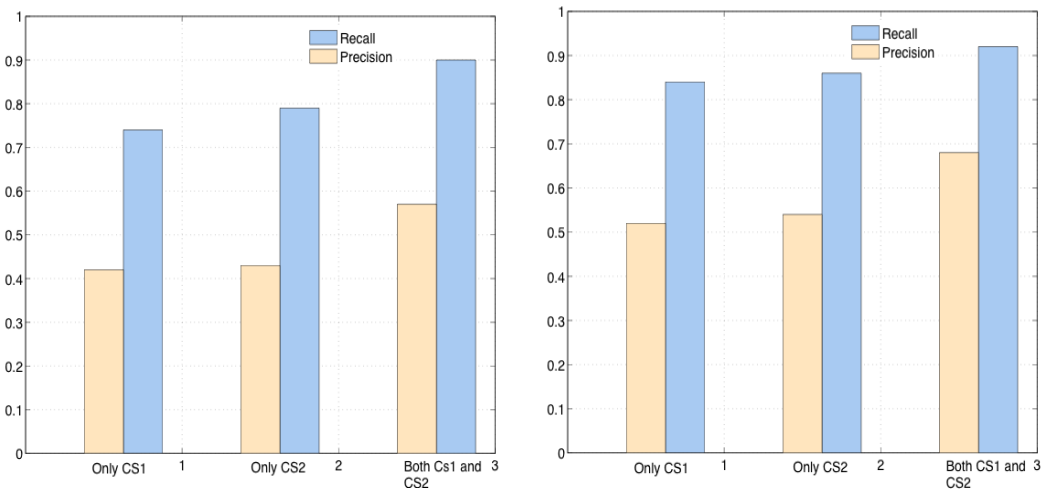



Figure 39: Gesture spotting on continuous data collected in CS2. The figure on the left side is obtained from Adaboost and the right side figure is for SVM

### 5.6 Conclusion

This chapter discussed in detail the problem of detecting out-of-vocabulary label samples when only samples belonging to in-vocabulary labels are available for training. It proposes two novel methodologies based on boosting and one class support vector machines to solve the problem. These methodologies are evaluated using the activity gesture datasets described in the previous chapter. It also proposed a novel ergodic graphical model based gesture spotting network building on top of the threshold models, gesture models and Viterbi algorithm. The superior performance of this gesture-spotting network is demonstrated against a popular spotting network based on HMM.

On a more general note, the chapter also discusses and demonstrates the effectiveness of the proposed multiple one class SVM approach for detecting out-of-vocabulary labeled samples. The results from the experiments conducted in the study illustrate the superiority of the proposed technique not only over it is single class counterpart, but also on models with very high accuracies on in-vocabulary samples. Considering that for most of the real-world applications, the

assumption of test samples belonging to in-vocabulary labels does not hold good, the body of work presented in this chapter has immense merit for the research community in dealing with such applications.

Chapter 6

# LEARNING FROM AUXILIARY DATA SAMPLES

## 6.1 Introduction

Many machine learning algorithms, by default, tend to assume that the training and test data will be drawn either from the same feature space or data distributions that are independent and identical (IID). However, this is seldom true in real-life, where one is more likely to train on data obtained from a specific set of sources and test on data occurring in certain targets. In the context of activity gesture recognition this can occur, when the training set consists of samples from a particular subject or performed in a certain environment, while the testing samples might be from a different subject or performed in a different context. In particular, in this dissertation, the training set of activity gestures were performed in a control setting under the supervision of a researcher, while subjects in a more realistic setting performed the test gestures. This variation in the distribution necessitates recollection of training data and rebuilding models from scratch to accommodate and learn the difference. In such cases, effort can be minimized by using algorithms, which are able to cater to the target domain by way of a knowledge transfer from the source to the target domain. This alleviates the need for labeling large amounts of data from the target domain. This problem of transferring knowledge from one domain into another for either accounting for the change in learning tasks or change in the data distribution is typically termed Transfer Learning.

This chapter is focused on developing algorithms that can learn from auxiliary or source domain sources in conjunction with a small amount of data from the target domain to improve the performance of the predictive models on the target domain. It begins by providing a review of different approaches for transfer learning in Section 6.2. Then it describes the proposed approach for transfer learning using cost sensitive boosting in Section 6.3. The theoretical analysis of the proposed framework in terms of the training and generalization error bounds is presented in Section 6.4. The details of the experimental evaluation of the proposed framework are discussed in Section 6.5. While the objective of developing the framework was to support activity gesture

recognition across contexts, the effectiveness of the framework as a generic transfer learning approach is experimented on other benchmarking datasets. Section 6.6 concludes the work presented in this chapter.

### 6.2 Related Work

This section presents a discussion on the different approaches in the literature for transfer learning. Before beginning this discussion we present an over view of the notations and symbols used in this chapter. We define a domain D to consist of two components, the feature/data space $\chi$and the marginal probability distribution $P(X)$, where $X = \{x_1, \dots x_n\} \in \chi$ .Associated with every domain is a set of learning tasks, typically classification denoted by $T(.)$ In the context of gesture recognition, all the dimensions of the feature vector together define the feature space $\chi$ and $X$ correspond to a certain set of training samples drawn from the distribution$P(X)$. Given a specific domain, $D = (X, P(X))$ a task consists of two components: a label space $Y$ and an objective function $f(.)$ that is not observed but learnt from the training data consisting of pairs $(x_i, y_i)$, where $x_i \in X$ and $y_i \in Y$. The function $f(.)$ is used to predict the label of a data point.

**Definition:** *Given a source domain $T_d$, a target domain $T_s$ and the corresponding set of learning tasks$T_{td}$ and $T_{ts}$ where, $T_d \neq T_s$or $T_{td} \neq T_{ts}$, Transfer Learning can be defined as a learning technique that works at improving the performance of a predictive function $f_t(.)$ in the given target domain $T_s$ using the knowledge gained from $T_d$ and $T_s$.*

The condition $T_d \neq T_s$ implies $X_d \neq X_s$ or $P(X_d) \neq P(X_s)$ i.e. when the source and target domains are different, either (1) the feature spaces between the domains may be different or (2) the marginal probability distributions of the data from the domains may be different. Similarly, the condition$T_{td} \neq T_{ts}$ implies $Y_s \neq Y_t$ or $P(Y_s \mid X_s) \neq P(Y_t \mid X_t)$ meaning when the learning tasks are different, either (1) the label spaces between the domains may be different or (2) the conditional probability distribution of the labels may be different.

There are three main components of transfer learning as defined by Pan et al. [106]: *what to transfer*, *how to transfer* and *when to transfer*. The first research question of *what to transfer* deals with which part of the knowledge needs to be transferred. In certain scenarios, some instances

from the source domain might still be relevant for learning models for the target domain. In this case, instance transfer might be the best approach. After what part of knowledge to be transferred has been identified, the research question of *how to transfer* comes into play. This basically deals with developing learning algorithms for transferring the identified information. This can be achieved through inductive or transductive learning as commonly used in traditional machine learning frameworks. Finally, *when to transfer* looks at recognizing scenarios in which knowledge transfer should and should not be done. In some cases, when the source domain and target domain are not related to each other, brute-force transfer may prove to be unsuccessful and even hurt the performance in the target domain. This phenomenon is known as *negative transfer*. This happens typically when the source domain data and task contribute to the reduced performance of learning in the target domain. There exists little work in this area of study for identifying scenarios where negative transfer can happen.

TL approaches can be categorized into 4 groups based on the question *what to transfer*, namely instance transfer, feature transfer, parameter transfer and relational transfer. In the following subsections, we provide an extensive review of instance transfer techniques, while briefly touching upon the other modes of transfer.

### 6.2.1 Instance Transfer

The instance-transfer approach to the inductive transfer learning setting is intuitively appealing: although the source domain data cannot be reused directly, there are certain parts of the data that can still be reused together with a few labeled data in the target domain.

Dai et al. [107] propose a boosting based transfer learning algorithm, TrAdaBoost as described in Figure 40 which is an extension of the Freund and Schapire's [105] AdaBoost algorithm, to address the inductive transfer learning problems. The primary assumption made by the algorithm is that due to the difference only in the data distributions between the source and target domains, some of the source domain data may be useful in learning for the target domain, but some of them may not and even might be harmful. It attempts to iteratively re-weight the source domain data to reduce the effect of the `bad' source data while encourage the `good' source data to contribute

more for the target domain. For each round of iteration, TrAdaBoost trains the base classifier on the weighted source and target data. The error is only calculated on the target data. Furthermore, TrAdaBoost uses the same strategy as AdaBoost to update the incorrectly classified examples in the target domain while using a different strategy from AdaBoost to update the incorrectly classified source examples in the source domain.

Jiang et al[108] linearly combine several adaptation heuristics using instance-level and global coefficients, into a unified objective function. Their suggested framework aims at tackling both

**Algorithm** TrAdaBoost

**Input**: the two labeled datasets $T_d$ and $T_s$, the unlabelled dataset $S$, a base learning algorithm Weak Learner, the maximum number of iterations, C, a vector consisting of cost factors associated with every sample in $T_d$.

**Initialize**: the initial weight vector $w^1$=$\{w_1^1, w_2^1, \cdots, w_{n+m}^1\}$.

For $t = 1, \ldots, N$

1. Set $p^t = {w^t}/{\sum_{i=1}^{m+n} w^t(i)}$
2. Call the weak learner, providing it with $p^t$ and the combined training set of $T_d$ and $T_s$ along with the cost factors for $T_d$ C. Get back the hypothesis $h_t : X \to Y; Y \in \{-1, 1\}$.
3. Calculate the weighted error of $h_t$ on $T_s$

$$\varepsilon_t = \sum_{i=m+1}^{m+n} \frac{w^t(i) \cdot |h_t(x_i) - y_i|}{\sum_{j=m+1}^{m+n} w^t(j)}$$

4. Set $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ and $\beta = 1/(1 + \sqrt{2 \log n / N})$. Note that $\varepsilon_t$has to be less than $1/2$
5. Update the new weight vector

$$w^{t+1}(i) = \begin{cases} w_i^t \beta^{|h^t(x_i) - y_i|}, & 1 \le i \le n \\ w_i^t \beta_t^{-|h^t(x_i) - y_i|}, & n+1 \le i \le m+n \end{cases}$$

Output: The final hypothesis

$$h_f(x) = \begin{cases} 1, & \prod_{t=N/2}^{N} \beta_t^{-h_t(x)} \ge \prod_{t=N/2}^{N} \beta_t^{-1/2} \\ 0, \text{Otherwise} \end{cases}$$

Figure 40: TrAdaBoost algorithm of Dai et al [107]

areas of *domain adaptation* namely *labeling adaptation* and *instance adaptation*, by (1) removing misleading training instances in the source domain, (2) assigning more weights to labeled target instances than labeled source instances and (3) augmenting training instances using target

instances with predicted labels. Liao et al[109] propose an active learning approach for selecting unlabeled samples from the target domain that needs to be labeled using samples from the source domain. Shi et al[110] provide a framework that actively transfers knowledge across domains in unsupervised and semi-supervised settings. Their method, *AcTraK* (Actively Transfer Knowledge), uses transfer learning to complement active labeling of unlabeled data. While active learning helps mitigate the risk of domain difference that comes with transfer learning, they also minimize the cost of having to assign labels with the help of a domain expert. They construct a classifier with labeled examples from both the source and target domains that selectively requests for an expert's label only when the confidence of knowledge transfer for a particular example is low.

Inductive transfer learning assumes that some amount of labeled data from the target domain is available for training. However transductive transfer learning only assumes that there is plenty of unlabeled target domain is available for training. The change in the data distribution between the source and target domain is attributed to sample selection bias. Correcting the sample selection bias has its inspirations drawn from the Nobel prize winning work of Heckmann in economics [111]. Huang et al[112] reformulate the problem of empirical risk minimization using concepts of importance sampling and *kernel mean matching*. The objective of *kernel mean matching* is to ensure that distance between the mean of the data samples from both the domains is minimum in some reproducing kernel Hilbert space. With the objective of learning a parameter, $\theta$ for minimizing the risk estimated on $P_t = P(X_t, Y_t)$, given just the source data $(X_s, Y_s)$ drawn from $P_s = P(X_s, Y_s)$ and an intent of solving for the *selection bias* ($P_s \neq P_t$) problem, they use importance sampling to estimate risk with respect to $P_s$ using $P_t$. The approach centers around an risk minimization framework for estimating the values of 'weights' $\beta$ that correspond to the importance of samples in the source domain with respect to the target domain.

There have been different approaches proposed for estimating the values of the 'weights' $\beta$. Sugiyama et al[113] propose an algorithm named *Kullback-Leibler Importance Estimation Procedure* (KLIEP) to estimate $\frac{P(T_s)}{P(T_d)}$ directly, based on the minimization of the Kullback-Leibler

divergence measure. KLIEP is integrated with cross validation to perform model selection automatically in two steps: (1) estimating the weights of the source domain data; (2) training models on the reweighted data. Zadrozny et al[114] propose to estimate the terms $P(T_s)$ and $P(T_d)$ independently by constructing simple classification problems. Bickel et al[115]combine the two steps of empirical risk minimization and weight estimation by deriving a kernel-logistic regression classifier.

### 6.2.2 Feature Transfer

This approach for transfer learning, aims at finding `good' feature representations that minimize the divergence between the source and target domain data, while reducing the classification or regression error. As is the case with instance transfer, if there is sufficient amount of labeled data available in the source domain and the target domain, inductive approaches are proposed, else, unsupervised transductive techniques are considered.

Argyriou et al[116] propose a method for learning a low-dimensional feature representation which is shared across a set of multiple related tasks. Building upon the 1-norm regularization problem, they use a new (2,1)-norm regularizer to come up with a non-convex optimization problem which attempts to simultaneously select a low dimensional feature representation and learn them. Ruckert et al[117] propose a kernel based approach for feature transfer. The approach proposed by them aims at determining a suitable kernel for the target data that reduces the classification error. These approaches for feature transfer fall under the umbrella of multi-task learning, where the objective is to find a common feature representation across the tasks.

Transductive approaches for feature transfer are typically unsupervised in nature. Blitzer et al[118] propose a technique called structural correspondence learning that focuses on using unlabeled data from both the source and target domains to learn a common feature representation that is meaningful across both the domains. Pan et al[119] exploit a dimensionality reduction method named, the *Maximum Mean Discrepancy Embedding* (MMDE) to learn a shared low dimensional latent feature space, such that the distributions between the source and target domain data are the same or close to each other. The theory of Maximum Mean Discrepancy described by Bogwart et

al [120] states that the distance between distributions of two samples is equivalent to the distance between the means of the two samples mapped into a Reproducible Kernel Hilbert Space(RKHS). By capitalizing on this theory, MMDE converts the problem of minimizing a distance function in feature space into a semi definite program in RKHS to identify a latent set of features. Post this, supervised and semi-supervised learning approaches are used to train a model for a mapping between the tasks and data across both the domains. However this method is computationally inexpensive. Torralba et al[121] present a multi-class boosting procedure called *joint boosting* to solve the problem of detecting large number of different object classes in cluttered scenes, by finding common features that can be shared across classes, in an effort to reduce computational and sample complexity.

### 6.2.3 Parameter Transfer

Parametric knowledge transfers have been observed to fall under two basic types of approaches as described in the seminal work of Thrun [122]: (1) approaches that partition the parameter space of a conventional learning algorithm into task-specific parameters and general(cross-task) parameters and (2) approaches that learn shape constraints, which are superimposed when learning a new function. Evgeniou et al[123] propose an SVM based parameter transfer approach, where the parameter for the SVMs for the source and target domain, $w_s$ and $w_t$ share a common parameter $w_o$. Thus $w_s = w_o + v_s$ and $w_t = w_o + v_t$. They then formulate an optimization framework for determining the parameters $w_o, v_t, v_s$.

Lawrence et al[124] proposed an efficient algorithm known as MT-IVM (Multi-task, Informative vector machine), which is based on Gaussian Processes (GP), to handle the multi-task learning case. MT-IVM tries to learn parameters of a Gaussian Process over multiple tasks by sharing the same GP prior. Most of the approaches on parameter transfer are devoted towards multi-task learning. There has not been much work in trying to learn parameters for the target task, when the source and the target task is the same, but the distribution of the data in both the domains is different.

### 6.2.4 Relational Transfer

Different from the other three contexts, the relational knowledge transfer approach deals with TL problems in relational domains, where the data are non-i.i.d and can be represented by multiple relations, such as bioinformatics network data and social network data. This approach does not assume that the data drawn from each domain be independent and identically distributed as traditionally assumed. It tries to transfer the relationship among data from a source domain to target domain. Mihalkoval et al[125], perform transfer between Markov Logic Networks(MLN). Richardson et al [126] define an MLN to consist of a set of first-order logic formulae; each with a weight attached, and provides a model for the joint distribution of a set of variables. Given a learned MLN for a source task they learn an MLN for a related target task by starting with the source-task one and diagnosing each formula, adjusting ones that are too general or too specific in the target domain. The hypothesis space for the target task is therefore defined in relation to the source task MLN by the operators that generalize or specify formulas.

In the domain of activity recognition, Kasteren et al[37] present a framework that allows to transfer knowledge of activity recognition from one context to the next. They use wireless binary sensing nodes that can be used to capture activities anywhere in a household, such as measuring a door being opened, a toilet being flushed or the temperature of a stove rising. In this work, they describe a method which uses unlabeled data captured from house A together with labeled data from house B, to learn the parameters of model for activity recognition in house A. The difference in the domains appears in the form of the difference in the layout of the houses and thereby difference in the location of the sensors and the properties they measure. To solve this problem, the authors use a set of manual mapping operations namely Intersect, Duplicate and Union to get the final feature set over which a semi-supervised learning algorithm is used.

## 6.3 Cost sensitive boosting for transfer learning

We will use the notations in the TrAdaBoost algorithm described by Dai et al [107] to discuss the proposed cost sensitive boosting approach for transfer learning. $T_d = \{x_1, \cdots, x_m\}$ refers to the different distribution labeled training data, $T_s = \{x_1, \cdots, x_n\}$ refers to a small amount of same

distribution labeled training data; $S$ refers to the unlabeled same distribution test data. The points in $T_d$, $T_s$ and $S$ are sampled from a d-dimensional instance space $X$. It is assumed that the size of $T_d$ is significantly greater than the size of $T_s$ ($m \gg n$). The set $\{x_1, \cdots x_m, x_{m+1}, \cdots, x_{m+n}\}$ corresponds to the combined training set consisting of samples from $T_d$ and $T_s$. The first $m$ samples belong to $T_d$ and the last $n$ samples from $T_s$. The objective here is to learn a model from many labeled samples from different distribution data $T_d$ and a few labeled samples from same distribution data $T_s$, to minimize the prediction error on the unlabeled same distribution data $S$.

We now present the cost sensitive boosting framework for transfer learning CTrAdaBoost that extends the original AdaBoost framework proposed by Freund and Schapire [105]. AdaBoost was originally developed as an online learning framework that aims to build a strong classifier by linearly combining several weak classifiers. However, it assumes that the training and test data are sampled from the same instance space according to identical sampling distributions, which does not hold good for the problem being addressed here. In the proposed extension, the boosting framework is applied separately to same and different distribution data. Furthermore, the boosting updates for the different distribution data is slightly modified to take into account the cost factors that represent the relevance of the samples with respect to same distribution data. It ensures that weights of instances in different distribution data that are not relevant to same distribution data are slowly decreased to reduce its impact on learning, while maintaining the weights of the relevant samples.

A formal description of the framework is presented in Algorithm 1. The framework is pivoted around two hinges: 1) applying separate boosting schemes for $T_d$ and $T_s$, 2) attaching costs to samples in $T_d$ that encode the relevance of the samples to $T_s$ estimated in an offline process. These two steps help the framework to minimize the training error over $T_s$, while at the same time reducing the net misclassification costs over $T_d$. This dual objective helps in reducing the chances of over fitting the final classifier to a few samples in $T_s$. Retaining the original boosting algorithm for minimizing the training error over $T_s$, the weight update factor $\alpha_t^{T_s}$, is determined so as to focus on learning the harder same distribution samples. However, since the number of samples in $T_s$, is

small, focusing on learning a few samples in $T_s$ could easily lead to over fitting, especially with increasing number of iterations. This can be overcome by adding sufficient number of relevant samples from $T_d$ to the training process without biasing the outcome on the weighted $T_s$ dataset. This is achieved by adding a cost sensitive boosting update for samples in $T_d$. Applying the cost factors ensures the weight updates to $T_d$ are made relative to its importance to $T_s$, and at the same reducing the training error on $T_d$.

As can be seen from the algorithm, for a given iteration, weights different distribution samples that are incorrectly classified are increased according to the cost factors associated with them. So if a sample is irrelevant and is misclassified, the weight for the sample decreases significantly compared to that of a misclassified relevant different distribution instance for the next round of iteration. Similarly, for different distribution samples that are correctly classified, the weights are changed according to the cost factors. The weight of a correctly classified different distribution sample with low relevance is made lower than a correctly classified different distribution sample with a higher cost. The addition of the cost factors provides a tool for reducing the impact of the weak classifier during the iterative process. This is important because, as we reach higher rounds of iteration, the weak hypothesis are learnt primarily from a subset of the same distribution data (harder examples from $T_s$), which could lead to over fitting. Thus even if a different distribution sample is relevant, there is a high tendency for the weak classifier to predict the wrong outcome. In the absence of the cost factors for such scenarios, the different distribution sample is interpreted as a conflicting sample and the weight is significantly reduced for the successive iteration. Thus an important aspect of the proposed framework is the cost factors associated with the different distribution samples. It has to be noted that the terminology of cost sensitivity used in this work is different from the one typically used in the literature for associating costs for correct and incorrect classification. The next subsection describes an approach adopted in this work for estimating the costs.

Algorithm 1: cost sensitive boosting for transfer learning

**Algorithm** CTrAdaBoost

**Input**: the two labeled datasets $T_d$ and $T_s$, the unlabelled dataset $S$, a base learning algorithm Weak Learner, the maximum number of iterations, C, a vector consisting of cost factors associated with every sample in $T_d$.

**Initialize**: Set the initial weights for all samples in $T_d$ and $T_s$. This is represented by the weight vector $w^1$={$w^1(1), w^1(2), \cdots, w^1(m+n)$}. If the user does not specify an initial weight vector $w^1(i) = \frac{1}{(m+n)}$ $\forall i$.

For $t = 1, \ldots, T$

6. Set $w^t = {w^t}/{\sum_{i=1}^{m+n} w^t(i)}$
7. Call the weak learner, providing it with $w^t$ and the combined training set of $T_d$ and $T_s$ along with the cost factors for $T_d$ C. Get back the hypothesis $h_t : X \to Y; Y \in \{-1, 1\}$.
8. Calculate the weighted error of $h_t$ on $T_s$

$$\varepsilon_t^{T_s} = \sum_{i=m+1}^{m+n} \frac{w^t(i) \cdot |h_t(x_i) - y_i|}{\sum_{j=m+1}^{m+n} w^t(j)}$$

9. Calculate the cost factored weighted error of $h_t$ on $T_d$

$$\varepsilon_t^{T_d} = \sum_{i=1}^{m} \frac{w^t(i) \cdot c_i \cdot |h_t(x_i) - y_i|}{\sum_{j=1}^{m} w^t(j) \cdot c_j}$$

10. Set

$$\alpha_t^{T_s} = \frac{1}{2}\log\left(\frac{1-\varepsilon_t^{T_s}}{\varepsilon_t^{T_s}}\right) \text{ and}$$

$$\alpha_t^{T_d} = \frac{1}{2}\log\left(\frac{\sum_{i=1, y_i = h_t(x_i)}^{m} c_i w^t(i)}{\sum_{i=1, y_i \neq h_t(x_i)}^{m} c_i w^t(i)}\right)$$

11. Update the new weight vector

$$w^{t+1}(i) = \begin{cases} {c_i \cdot w^t(i) \cdot \exp\left(-\alpha_t^{T_d} h_t(x_i) y_i\right)}/{Z_t^{T_d}}, & 1 \le i \le m \\ {w^t(i) \cdot \exp\left(-\alpha_t^{T_s} h_t(x_i) y_i\right)}/{Z_t^{T_s}}, & m+1 \le i \le m+n \end{cases}$$

Where $Z_t^{T_d}$ and $Z_t^{T_s}$ are the normalization constants for $T_d$ and $T_s$ respectiely

Output: The final hypothesis

$$H(x) = \text{sign}\left(\sum_{t=i}^{T} \alpha_t^{T_s} h_t(x)\right)$$

### 6.3.1 Cost estimation:

The role of the cost items is primarily to associate the relevance of different distribution samples $T_d$ with respect to the same distribution samples $T_s$. This can be computed in a supervised or unsupervised manner. For the experiments conducted in this study, these weights were computed in a supervised manner using an approach proposed by Jiang et al [190] for pruning misleading different domain instances. Their approach involves learning a classifier model using the few labeled same distribution sample set $T_s$ and using this model select instances in different distribution set that are correctly classified. Instead of eliminating all the instances that are incorrectly classified, we use the probability of correct classification associated with each sample as the cost factor. Thus in the process, samples in $T_d$ with high probability of correct classification have higher cost items compared to the samples in $T_d$ with low probability of correct classification. Since the estimated values are probabilities, the cost thus computed is already normalized between [0, 1].

The cost factors can be computed in an unsupervised manner taking into account the entire unlabeled same distribution set $S$. These approaches are similar to the ones discussed in the transductive approaches for transfer learning such as importance sampling [112], Kullback-Liebler importance estimation procedure [113] etc. The importance of the different distribution samples estimated through these procedures can also be substituted for the cost factors. Supervised approaches for cost estimation includes approaches that computes some type of similarity between the different and same distribution samples belonging to a particular class.

### 6.3.2 Relation to other boosting based transfer learning approaches

While this is not the first boosting approach for transfer learning, a discussion centered on the similarity between the proposed approach and other boosting based transfer learning approaches is presented in this section. The most commonly cited boosting based transfer learning approach is the TrAdaBoost algorithm of Dai et al [107] as described in the related work section. The CTrAdaBoost is similar to the TrAdaBoost in the manner in which the weights of same distributions samples are updated. Both of the techniques employ the original boosting based

approach for updating the weights of same distribution samples. The main difference between the two algorithms is in the manner in which the different distribution samples are handled. TrAdaBoost uses the weighted majority algorithm to adjust the weights, repeatedly decreasing the weight of incorrectly predicted different distribution sample by a constant factor $\beta$. It also notes that since the error on the same distribution samples converges to 0 only after half of the total number of iterations, the TrAdaBoost algorithm considers only the weak hypothesis learnt in the second half of the boosting iterations to arrive at the final strong classifier. Intuitively the weak hypothesis learnt during the initial rounds of boosting fit a majority of data, with the focus on the harder examples during the later rounds. If the harder examples represent outliers in the $T_s$ data, then TrAdaBoost has a tendency to over fit the labeled same distribution training data.

Another important difference between TrAdaBoost and CTrAdaBoost is the manner in which weight updates are performed on the different distribution training data. In TrAdaBoost, the weights of $T_d$ samples either decrease or remain constant between successive iterations. There is no way in which the weight of a relevant sample can be increased, once decreased during the previous rounds of boosting. When the weights of these relevant samples become very low, their influence on learning a good weak hypothesis becomes negligible. In contrast, the CTrAdaBoost algorithm allows for increase in weights for the different distribution samples. However it ensures that weight increase is proportional to the relevance of the sample with respect to the majority of the same distribution samples. Thus during the later rounds of boosting, CTrAdaboost has a higher potential to retain relevant different distribution samples for learning the weak hypothesis compared to TrAdaBoost.

The set based boosting for instance level transfer (TransferBoost) proposed by Eaton et al [191] is another algorithm that uses boosting for transfer learning. Transferboost uses a set based weight updating scheme. It breaks the different distribution instances into task based sets. Instead of updating the weights of individual instances, it updates the weights of instances in a set in a similar manner. The scheme adopted for updating the weights of the same distribution data is similar to the CTrAdaBoost algorithm. Furthermore, it can be noted that the weight update in

TransferBoost for the different distribution instances is a special case of the CTrAdaBoost algorithm. When the parameter $\alpha_t$ , in the TransferBoost algorithm that represents the transferability of set is made a constant at the individual instance level, then TransferBoost boils down to CTrAdaBoost.

### 6.4 Theoretical Analysis of the algorithm

The previous section discussed the cost sensitive boosting framework for transfer learning and in particular presented the CTrAdaBoost algorithm. This section presents a theoretical analysis of the framework in terms of its convergence property and shows why the framework is able to learn even when the distributions of the data are not identical. This analysis builds on the theory of AdaBoost presented by Freund et al. [192] [193], with contributions drawn from the cost sensitive framework proposed by [194]. Continuing with the terminology introduced in Section 6.3, $T_d$ and $T_s$ represents labeled training data from the different and same distributions and S represents the test data. Since the weight -update equations for data in $T_d$ and $T_s$ are separate, the theoretical analysis also decouples the convergence of the algorithm for these two datasets. Theorem 1 discusses the convergence property of the CTrAdaBoost for the different distribution data $T_d$. Theorem 2 presents an upper bound for the error on the different distribution training data $T_d$. Theorem 3 and 4 discuss the same analysis for the same distribution training set $T_s$. Finally theorem 5 gives an alternative upper-bound of the generalization error on the same distribution data $T_s$.

**Theorem1:** Considering the CTrAdaBoost algorithm described in Algorithm 1 the training error of the final classifier $H$ on the different distribution training set $T_d$ is bounded as follows

$$\frac{1}{m}|\{i : H(x_i) \neq y_i\}| \leq \prod_{t=1}^{T} Z_t$$

**Proof:** Consider the weight update equation for samples belonging to the different distribution training data $T_d$

$$w^{t+1}(i) = \frac{w^t(i) c_i e^{-\left(\alpha_t y_i h^t(x_i)\right)}}{Z_t}$$

Where $Z_t = \sum_{i=1}^{m} c_i\, w^t(i)\ exp(-\alpha_t\, y_i\ h^t(x_i))$ is the normalization factor for the iteration $t$. Using this equation the weight update at the $(T+1)^{th}$ iteration can be recursively unraveled as

$$w^{T+1}(i) = \frac{c_i^T e^{-(\sum_{t=1}^{T} \alpha_t\, y_i\, h^t(x_i))}}{m\ \prod_{t=1}^{T} Z_t}$$

For an individual sample $x_i \in T_d$ if $H(x_i) \neq y_i$, then

$$\sum_{t=1}^{T} y_i \alpha_t h^t(x_i) \ \leq 0$$

$$\exp\left(-\sum_{t=1}^{T} y_i \alpha_t h^t(x_i)\right) \ \geq 1$$

Thus,

$$\frac{1}{m}|\{i : H(x_i) \neq y_i\}| \leq \frac{1}{m} \sum_{\mathrm{i}=1}^{\mathrm{m}} \exp\left(-\sum_{t=1}^{T} y_i \alpha_t h^t(x_i)\right)$$

Substituting for $\exp(-\sum_{t=1}^{T} y_i \alpha_t h^t(x_i))$ from the above equation, we have

$$\frac{1}{m}|\{i : H(x_i) \neq y_i\}| \leq \sum_{\mathrm{i}=1}^{\mathrm{m}} \left(\frac{\prod_{t=1}^{T} Z_t}{c_i^T}\right) w^{T+1}(i)$$

$$= \prod_{t=1}^{T} Z_t \left[\sum_{i=1}^{m} \frac{w^{T+1}(i)}{c_i^T}\right]$$

$$\leq \nu \prod_{t=1}^{T} Z_t$$

where $\upsilon \in [0, 1]$ is a constant and can be set as $\nu = \min\{c_1, \cdots, c_m\}$ and $\sum_{i=1}^{m} w^{T+1}(i) = 1$.

Thus it can be seen that

$$\frac{1}{m}|\{i : H(x_i) \neq y_i\}| \leq \prod_{t=1}^{T} Z_t$$

■

Theorem 1 shows that in order to minimize the training error on $T_d$, a reasonable approach might be to greedily minimize $Z_t$ on each round of the boosting. The next theorem provides a mechanism to apply this principle for minimizing the error on $T_d$.

**Theorem 2:** Define $r_t = \sum_{i=1}^{m} c_i\, w^t(i) y_i\, h^t(x_i)$. Assuming that each of the weak hypothesis $h^t \in [-1, 1]$, if we choose

$$\alpha_t = \frac{1}{2} \log\left[\frac{\sum_{i=1}^{m} c_i w^t(i) + r_t}{\sum_{i=1}^{m} c_i w^t(i) - r_t}\right]$$

Then the training error of $H$ on $T_d$ samples is at most $\prod_{t=1}^{T} \sqrt{1 - r^{t\,2}}$

**Proof:** Consider the following equation

$$Z_t = \sum_{i=1}^{m} c_i\, w^t(i)\, \exp(-\alpha_t\, y_i\, h^t(x_i))$$

$$\leq \sum_{i=1}^{m} \left[ c_i w^t(i) \left\{ \frac{1 + y_i\, h^t(x_i)}{2} e^{-\alpha_t} + \frac{1 - y_i\, h^t(x_i)}{2}\, e^{\alpha_t} \right\} \right]$$

This derivation is adopted from Freund et al, work in [193]. The upper bound is valid since $y_i\, h^t(x_i) \in [-1, 1]$ and $e^{-\alpha x}$ is convex for any constant $\alpha \in \mathbb{R}$..

From Theorem 1, it can be noted that the overall training error on $T_d$ can be minimized by greedily minimizing each of $Z_t$. Thus the value of $\alpha_t$ that minimizes $Z_t$ can be determined by differentiating equation (2) w.r.t $\alpha_t$ and equating it to 0 will result in

$$\sum_{i=1}^{m} \left[ c_i w^t(i) \left\{ -\frac{1 + y_i\, h^t(x_i)}{2} e^{-\alpha_t} + \frac{1 - y_i\, h^t(x_i)}{2}\, e^{\alpha_t} \right\} \right] = 0$$

$$\sum_{i=1}^{m} \left[ c_i w^t(i) \left\{ \frac{1 + y_i\, h^t(x_i)}{2} e^{-\alpha_t} \right\} \right] = \sum_{i=1}^{m} \left[ c_i w^t(i) \left\{ \frac{1 - y_i\, h^t(x_i)}{2} e^{\alpha_t} \right\} \right]$$

$$e^{2\alpha_t} = \frac{\sum_{i=1}^{m} \left[ c_i w^t(i) \left(1 + y_i h^t(x_i)\right) \right]}{\sum_{i=1}^{m} \left[ c_i w^t(i) \left(1 - y_i h^t(x_i)\right) \right]}$$

$$\alpha_t = \frac{1}{2} \log\left[\frac{\sum_{i=1}^{m} c_i w^t(i) + \sum_{i=1}^{m} c_i w^t(i)\, y_i h^t(x_i)}{\sum_{i=1}^{m} c_i w^t(i) - \sum_{i=1}^{m} c_i w^t(i)\, y_i h^t(x_i)}\right]$$

$$= \frac{1}{2} \log\left[\frac{\sum_{i=1}^{m} c_i w^t(i) + r_t}{\sum_{i=1}^{m} c_i w^t(i) - r_t}\right] \text{ (definition of } r_t)$$

Define $a_t = \sum_{i=1}^{m} c_i\, w^t(i)$, then

$$\alpha_t = \frac{1}{2} \log\left[\frac{a_t + r_t}{a_t - r_t}\right]$$

Substituting for $\alpha_t$ in equation (2) we have

$$Z_t \leq \left[\frac{a_t + r_t}{2} e^{-\alpha_t} + \frac{a_t - r_t}{2} e^{\alpha_t}\right]$$

$$\leq \left[\frac{a_t + r_t}{2} \frac{\sqrt{a_t - r_t}}{\sqrt{a_t + r_t}} + \frac{a_t - r_t}{2} \frac{\sqrt{a_t + r_t}}{\sqrt{a_t - r_t}}\right]$$

$$= \left[\frac{(a_t + r_t)\sqrt{a_t - r_t}\sqrt{a_t - r_t} + (a_t - r_t)\sqrt{a_t + r_t}\sqrt{a_t + r_t}}{2\sqrt{a_t - r_t}\sqrt{a_t + r_t}}\right]$$

$$= \frac{(a_t + r_t)(a_t - r_t) + (a_t + r_t)(a_t - r_t)}{2\sqrt{a_t^2 - r_t^2}}$$

$$= \frac{2(a_t^2 - r_t^2)}{2\sqrt{a_t^2 - r_t^2}}$$

$$= \sqrt{a_t^2 - r_t^2}$$

Since $a_t \in [0, 1]$ and $a_t \geq r_t$ $\sqrt{{a_t}^2 - {r_t}^2} \leq \sqrt{1 - {r_t}^2}$. Thus we have $Z_t \leq \sqrt{1 - {r_t}^2}$ and therefore it can be seen that

$$\prod_{t=1}^{T} Z_t \leq \prod_{t=1}^{T} \sqrt{1 - r_t^2}$$

■

Thus, with this setting of $\alpha_t$, it is reasonable to find a weak hypothesis $h_t$ that maximizes $|r_t|$ on $T_d$ on each round of boosting. In particular when $h_t \in \{-1, 1\}$, then

$$\Pr_{i \sim c\, w_t}[h_t(x_i) \neq y_i] = \frac{1 - r_t}{2}$$

This means that the average error on the $T_d$ according to the distribution that contains both the cost factors and the weights, can be minimized by maximizing the value of $r_t$.

Furthermore it is interesting to note that actual bound represented as $\sqrt{{a_t}^2 - {r_t}^2}$ is stricter than the bound observed in the case classic AdaBoost algorithm. Thus the CTrAdaBoost algorithm provides a stricter training error bound for the different distribution training data over Adaboost.

It can be further verified that $\frac{d^2 Z_t}{d\alpha_t^2}$ is strictly positive for all $\alpha_t \in \mathbb{R}$

$$\frac{d^2 Z_t}{d\alpha_t^2} = \sum_{i=1}^{m}\left[c_i w^t(i)\left\{\frac{1+y_i\, h^t(x_i)}{2} e^{-\alpha_t} + \frac{1-y_i\, h^t(x_i)}{2}\, e^{\alpha_t}\right\}\right]$$

$$\geq 0 \qquad \forall\, \alpha_t \in \mathbb{R}$$

and therefore $Z_t'$ can have at most one zero.

Let us now focus on the training error bounds for samples in $T_s$. Since the weight update equation is the same as that of regular boosting, the bounds derived for boosting by Schapire et al, [195] holds good for $T_s$. However for the sake of completeness these bounds are presented in the following theorems. The reader is requested to refer to [195] for the proofs.

**Theorem 3**: Consider the CTrAdaBoost algorithm described in Algorithm 1 the training error of the final classifier on $T_s$ is bounded as follows

$$\frac{1}{n}|\{i : H(x_i) \neq y_i\}| \leq \prod_{t=1}^{T} Z_t$$

■

**Theorem 4**: Define $s_t = \sum_{i=m+1}^{n} w^t(i) y_i\, h^t(x_i)$. Assuming that each of the weak hypothesis $h^t \in [-1, 1]$, if we choose

$$\alpha_t = \frac{1}{2}\log\left[\frac{1+s_t}{1-s_t}\right]$$

Then the training error of $H$ on $T_s$ samples is at most $\prod_{t=1}^{T}\sqrt{1-{s_t}^2}$

■

One constraint that is assumed here is that both $r_T$ and $s_t$ for every iteration is less than $1/2$. Theorems 1 to 4 together indicate that if the weak hypothesis learnt at every iteration minimizes the corresponding weighted error on $T_d$ and $T_s$, then the error of final strong classifier progressively decreases with increasing number of iterations. Therefore, CTrAdaBoost minimizes both the error on the same distribution data and the weighted average error on different distribution data simultaneously. Intuitively, it can be understood that CTrAdaBoost learns on the same distribution data with the help of relevant different distribution data, on condition that they do not result in more average training loss as additional training data. Furthermore, since the

weights of the relevant different distribution data does not rapidly become 0 as in the case of TrAdaBoost, the chances of over fitting on $T_s$ also reduces with increasing number of iterations.

All the above analysis was focused on the error on the training data and did not address the generalization issues. The next theorem presents an alternative upper-bound of the generalization error on the same distribution data. Freund and Schapire [193][196] showed how to bound the generalization error of the final hypothesis in terms of its training error, the sample size, the VC-dimension of the weak hypothesis space and the number of boosting iterations. Borrowing from the techniques proposed by Baum and Haussler [199], they derive the generalization bounds. Dai et al., utilize this work to derive the generalization bounds for TrAdaBoost. The following theorem is a straight forward modification of their derivation with a change in one of the parameters (sample size)

**Theorem 5**: Let $d_{vc}$ be the $V_c$ dimension of the hypothesis space, the generalization error on the same distribution data, with high probability is at most

$$\varepsilon + O\left(\sqrt{\frac{Td_{vc}}{(m+n)}}\right)$$

where $T$ is the number of boosting iterations and $m$ and $n$ are the sizes of the different and same distribution training data $T_d$ and $T_s$ and $\varepsilon$ is the training error on $T_d \cup T_s$ for the final strong classifier.

■

Let us look at the two components of the equation given in Theorem 5. The first part concerns with the training error. From the previous analysis on the training error bounds (Theorems 2 and 4) it can be seen that the error on the training data is lesser in the case of CTrAdaBoost when compared with that of AdaBoost and TrAdaBoost. The second component differs from the TrAdaBoost in the denominator in that includes the size of different distribution training data. It is however similar to that of AdaBoost. Thus adding both the components results in a stricter generalization error bound for CTrAdaBoost algorithm.

### 6.5 Experimental Evaluation

#### 6.5.1 Datasets

In order to evaluate the effectiveness of the proposed algorithm, experiments were performed on two different datasets (Activity Gesture (CS1 and CS2) and 20 Newsgroup [200]). The text data set were split to fit the transfer learning scenario, to make the different distribution data $T_d$ and same distribution data $T_s$ different. Newsgroup dataset represents a binary classification scenario, while the activity gesture dataset presents multi-label classification problem. The activity gesture dataset consists of gestures captured from two different settings. In the first set named as CS1, the subjects perform 5 activity gestures in a controlled setting under the supervision of the researcher. In the second set called as CS2, a different set of subjects perform the same activity gestures in a realistic setting without any researcher supervision interacting with a completely different set of objects. The CS2 dataset consists of activity gestures from 4 subjects each performing the real-activity 4 times. The data from one random selected trial was used as $T_s$ data, while the data from CS1 represented the $T_d$ data. For each of the gesture statistical and spectral features described in Section 4.3 was extracted. The Newsgroup dataset has a hierarchical structure with seven top categories. Under the top category there are multiple sub categories leading to a total of 20 labels. The task of identifying the top-level category is defined as the classification problem. The data is split into different and same distribution data based on the subcategory information, ensuring that the two datasets contain data from different subcategories. This also leads to a difference in the distribution of the dataset. Due to computational time complexity, the number of samples for splits was restricted to 1000.

Table 24 shows the description for each data. The first 4 datasets correspond to the 4 subjects from CS2 activity gesture data. The three datasets correspond to the splits from the Newsgroup data. The name of the data set 'rec vs talk' indicates that all positive instances are from the category 'rec' and the negative ones are from 'talk'. The other two 20 Newsgroup splits have been named in the same way. The Kullback-Leibler (KL) [197] divergence distance between $T_d$ and $T_s \cup S$ was computed on the features for each of the dataset and is presented in the table. It can be seen

that the KL-divergence measures for all the datasets are much larger than the same distribution case in which the measure will be close to 0.

Table 24: Description of the datasets used for the experiments

| **Dataset** | **KL-Divergence** | **Size** | |
|---|---|---|---|
| | | **T_d** | **T_s U S** |
| User 1 | 5.46 | 479 | 34 |
| User 2 | 7.55 | 479 | 32 |
| User 3 | 7.28 | 479 | 28 |
| User 4 | 5.80 | 479 | 34 |
| Rec vs talk | 1.102 | 1000 | 1000 |
| Rec vs sci | 1.021 | 1000 | 1000 |
| Sci vs talk | 0.854 | 1000 | 1000 |
| Comp vs rec | 1.126 | 1000 | 1000 |
| Comp vs sci | 1.048 | 1000 | 1000 |
| Comp vs talk | 1.303 | 1000 | 1000 |

### 6.5.2 Comparison Methods

For the experiments conducted in the work, SVM was used as the base learner on the Newsgroup dataset. Decision stumps was used the weak learner for the activity gesture dataset. The SVM implementation from LIBSVM was used for the experiments. Three baseline methods were implemented using LIBSVM. The different combinations of these baseline methods are presented in Table 25. The default parameters for a linear kernel implementation of the SVM were used to conduct the study. A threefold cross validation scheme was adopted for testing the performance on the each of the Newsgroup dataset splits. The $T_s$ samples were randomly selected for every fold. The maximum number of boosting iterations was set to 100.

Table 25: Description of the baseline methods. The methods using decision stumps were used only for activity gesture dataset

| **Training Data** | **20 newsgroup (SVM)** | **Activity Gesture (Decision Stumps)** |
|---|---|---|
| | **Baseline method** | **Baseline Method** |
| $T_d$ | SVMT_d | AdaTd |
| $T_s$ | SVMT_s | AdaTs |
| $T_d U\ T_s$ | SVMT_ds | AdaTds |
| $T_d U\ T_s$ | TrAda | TrAda |
| $T_d\ U\ T_s$ | CTrAda | CTrAda |
| $T_d$ | AdaT_d | |
| $T_s$ | AdaT_s | |
| $T_d U\ T_s$ | AdaT_sd | |

Certain constraints were added while training the weak base learners to ensure that the weights of the positive and negative samples were balanced. Furthermore, the difference in the weighted error on the $T_d$ dataset between two successive boosting iterations was used as early termination criteria. This was added for the following reason: assume that for a certain iteration, the training error on the same distribution set has become 0. This means that the weak hypothesis thus learnt is able to correctly classify all samples in $T_s$ and as a result these samples undergo no weight change for the subsequent iterations. However the training process continues on the weighted different distribution dataset. Since the weights of the same distribution samples remain constant, over large number of iterations, the weights of the different distribution samples tend to 0, resulting in a tendency for the base learners to over fit on the small same distribution dataset. This can be prevented by stopping the training process when the error on the different distribution data set does not change significantly between two successive iterations. We observed in our experiments that the boosting iterations converged in significantly less number (on an average 15) of iterations

**6.5.3 Results and Discussion**

The results for the newsgroup dataset are presented in Table 26. The accuracies presented here correspond to the situation when $T_s$ has a sample size 1% of $T_d$. It can be observed that the proposed CTrAdaBoost algorithm consistently provides better results over the other baselined techniques. CTrAdaBoost results in an average improvement of 3% over the second best baseline technique that varies with each dataset. It has to be noted that for all the three datasets, the performance of SVM trained in $T_s$ alone is better than its counterparts trained on $T_d$ and $T_s \cup T_d$. This shows that the traditional SVM framework does not support the transfer of knowledge when there is a change in the distribution of the data. In fact, adding the different distribution data to the same distribution data reduced the performance of SVM. Though the phenomenon of negative transfer was observed by Caruana et al [199] that resulted in the lowering of the generalization accuracies for the transfer-learning algorithm, it was not evident with the results obtained on the newsgroup dataset.

Table 26: Classification accuracies of the different techniques on Newsgroup data splits

| Method | RecVsTalk | RecVsSci | SciVsTalk | CompVsRec | CompVsSci | CompVsTalk |
|---|---|---|---|---|---|---|
| **SVMT_d** | 0.67 | 0.62 | 0.6 | 0.8 | 0.62 | 0.86 |
| **SVMT_s** | 0.75 | 0.75 | 0.71 | 0.83 | 0.64 | 0.75 |
| **SVMT_ds** | 0.73 | 0.68 | 0.67 | 0.63 | 0.68 | 0.86 |
| **AdaT_d** | 0.7 | 0.71 | 0.73 | 0.79 | 0.63 | 0.86 |
| **AdaT_s** | 0.7 | 0.71 | 0.73 | 0.7 | 0.62 | 0.72 |
| **AdaT_ds** | 0.71 | 0.68 | 0.64 | 0.81 | 0.69 | 0.86 |
| **TrAda** | 0.74 | 0.65 | 0.69 | 0.79 | 0.68 | 0.83 |
| **CTrAda** | **0.79** | **0.76** | **0.74** | **0.84** | **0.74** | 0.82 |

An interesting anomaly in the results presented in the Table 26 is the performance on the Comp Vs Talk split. The accuracy obtained by CTrAdaBoost is lower than that of the SVMT_ds model and the SVMT_d model. This can be understood by comparing the performance of the three SVM models. Note that the accuracy of SVMT_s is significantly lower than its two counterparts. This seems to suggest that the samples in the same distribution training set $T_s$ are of poor quality and potentially outliers of the same distribution set. Since SVMT_d and SVMT_ds do not distinguish between same and different distribution datasets, their performance does not change drastically. However in CTrAdaBoost, the samples in $T_s$ are used to determine the cost factors associated with the different distribution dataset. Thus noisy samples in $T_s$ lead to improper characterization of the relevance of samples in $T_d$, which in turn affects the performance. This indicates the need for a better technique for estimating the relevance of the different distribution samples.

The results of the transfer learning approaches with the baselined techniques on the activity gesture datasets are presented in Table 27. It has to be noted that decisions stumps were used as the weak learns for these experiments. Even on this dataset the CTrAdaBoost algorithm results in superior performance over other techniques. Again CTrAdaBoost appears to be consistent across the different user trials, an observation made on the Newsgroup dataset results. On an average CTrAdaBoost improves the performance of the next best algorithm by around 3%.

Table 27: Accuracies of the different baseline techniques on the Activity Gesture AG CS1 and CS2 datasets

| Method | User1 | User2 | User3 | User4 | Average |
|---|---|---|---|---|---|
| **AdaT_d** | 0.77 | 0.88 | 0.79 | 0.53 | 0.74 |
| **AdaT_s** | 0.70 | 0.60 | 0.65 | 0.76 | 0.68 |
| **AdaTds** | 0.91 | 0.92 | 0.83 | 0.77 | 0.86 |
| **TrAda** | 0.66 | 0.77 | 0.79 | 0.67 | 0.72 |
| **CTrAda** | **0.92** | **0.94** | **0.85** | **0.84** | **0.89** |

A common observation across all the results is the poor performance of the TrAdaBoost algorithm. Dai et al [107] note in their paper the superior performance of TrAdaBoost over the other baseline approaches. However we were unable to reproduce the exact results in their paper. In our analysis of the algorithm we noticed that beyond certain iteration, TrAdaBoost does not learn anything new. We observed that the total weight of the same distribution and different distribution training samples remain constant beyond certain iterations. This is discussed in more detail in the following sections.

To further understand the performance of the proposed cost sensitive boosting based transfer-learning algorithm, we varied the number of samples $T_s$ and observed the performance of the different algorithms. The percentage of samples in $T_s$ with respect to $T_d$ was varied as 0.3, 0.5, 1, 2, 3, 4, 5, 10, 20 and 50%. At one extreme of having only 0.3%, the total number of samples in $T_s$ is only 3, while for 50% it is around 500 samples. The size of the test set S was maintained at 1000, by adding more samples from the original pool (around 3000), but ensuring that there were no overlapping samples between $T_s$ and $S$. Again a threefold cross validation was performed and the error rates for the different sample sizes across the six Newsgroup splits are summarized in Figure 41Figure 42Figure 43Figure 44Figure 45Figure 46. The x-axis corresponds to the size of $T_s$ samples with respect to $T_d$ and the y-axis represents the classification error. The actual error value obtained for each percentage is denoted by the square overlapping the curve. The difference between the accuracy obtained by CTrAdaboost and the next best technique for each of the Newsgroup splits is presented in Figure 47,Figure **48**Figure 49Figure 50Figure 51Figure 52Figure

53Figure 54. Each figure corresponds to a specific ratio between number of samples in $T_s$ and $T_d$ starting from 1% till 50%.

It can be noticed that CTrAdaBoost consistently improves the performance on all the datasets up till the samples size of 5% except for the 'Rec Vs Sci' split, in which case it is on par to an SVM model trained only on the same distribution dataset. Beyond this point, the performance of CTrAdaBoost lags behind the SVM model trained on the same distribution data. This indicates that CTrAdaBoost is able to utilize the relevant information from the different distribution data to learn the same distribution data. From these figures we can see that the main contribution of CTrAdaBoost is in situations when the difference in the sample size of the same and different distribution data is less than 10%. When this percentage becomes higher, the samples in $T_s$ alone are sufficient as indicated by the high accuracies obtained by SVMTs. Another observation is the marginal increase in the error for CTrAdaBoost on 'Rec Vs Talk', 'Sci Vs Talk', 'comp Vs Rec' and 'Comp Vs Sci' datasets. This increase in error is consistently noted when the percentage increases from 5% to 10%. There was no clear evidence that explains this anomalous behavior. One can at best guess this as a result of noisy samples in the same distribution training dataset.

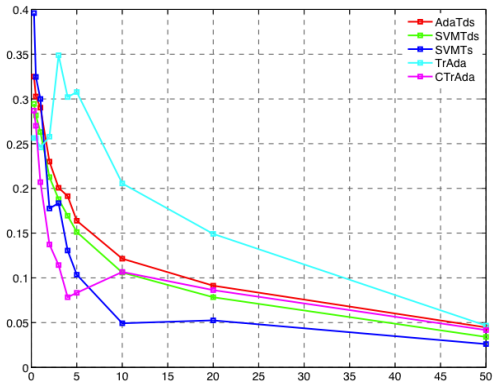


Figure 41: RecVsTalk error curves

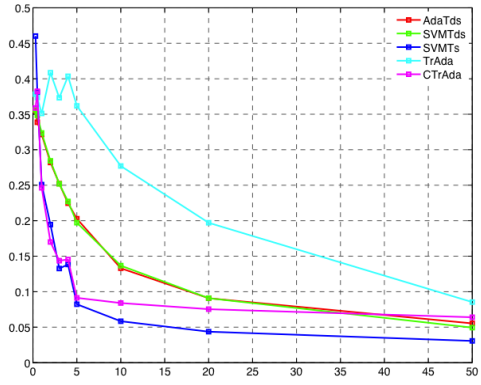


Figure 42: RecVvsSci error curves

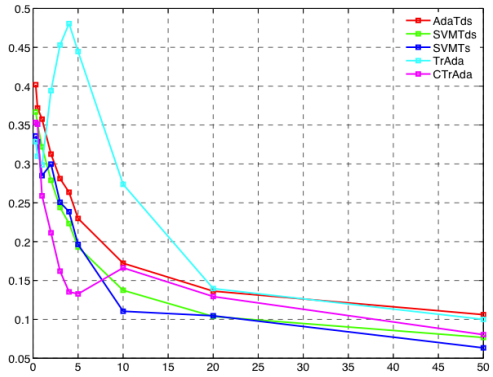



Figure 43: SciVsTalk error curves

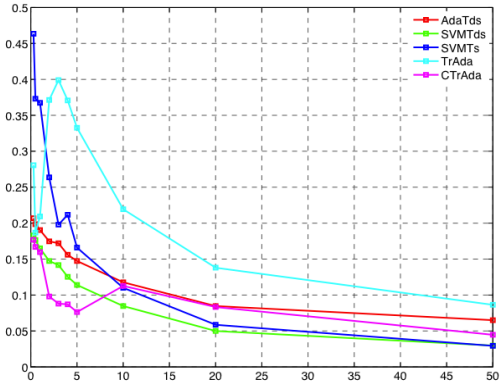



Figure 44: CompVsSci error curves

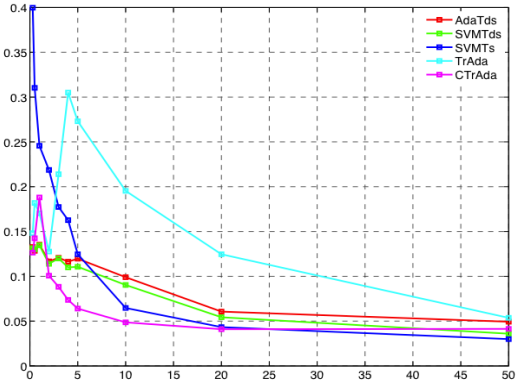



Figure 45: CompVsTalk error curves

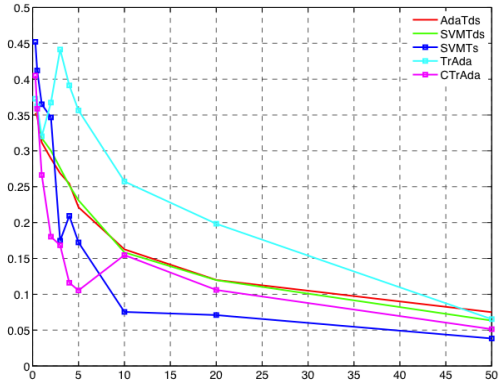



Figure 46: RecVsComp error curves

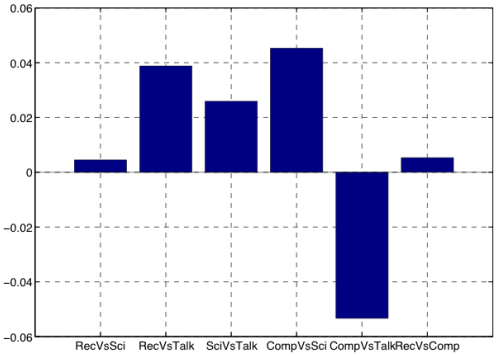



Figure 47: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 1%

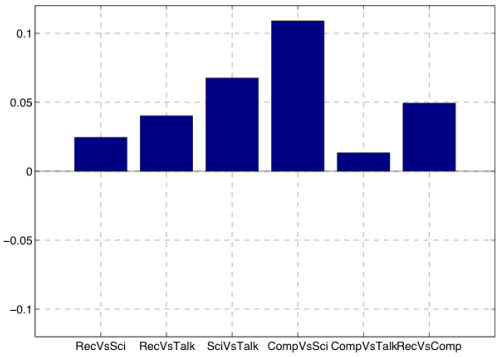



Figure 48: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$d and $\boldsymbol{T_s}$ at 2%

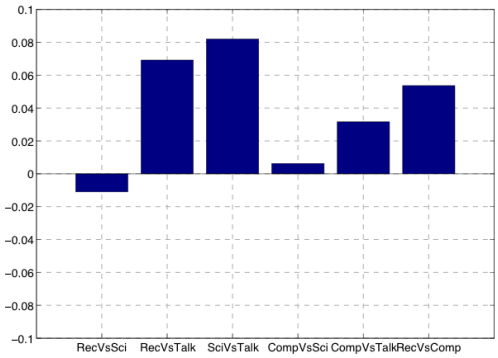



Figure 49: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 3%

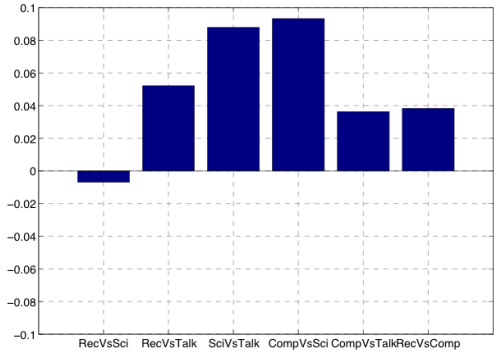



Figure 50: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 4%

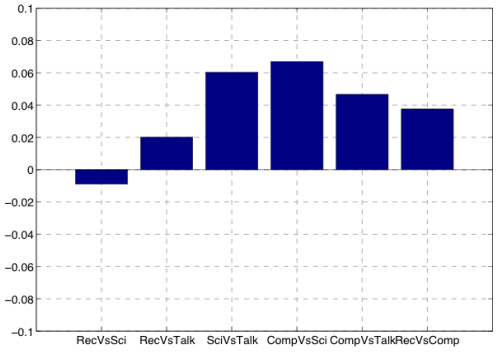



Figure 51: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 5%

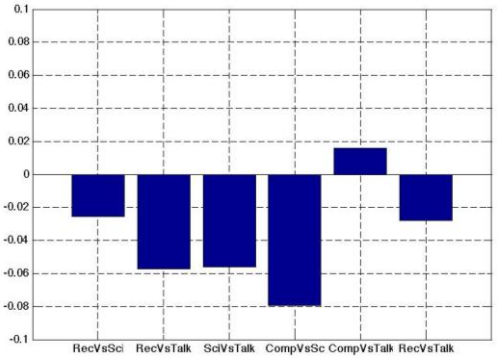



Figure 52: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 10%

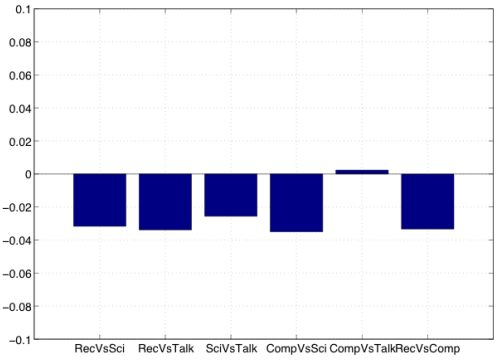



Figure 53: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 20%

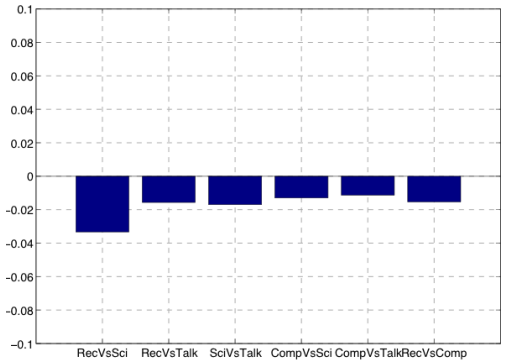



Figure 54: Difference in the accuracies for the ratio between $\boldsymbol{T_d}$ and $\boldsymbol{T_s}$ at 50%

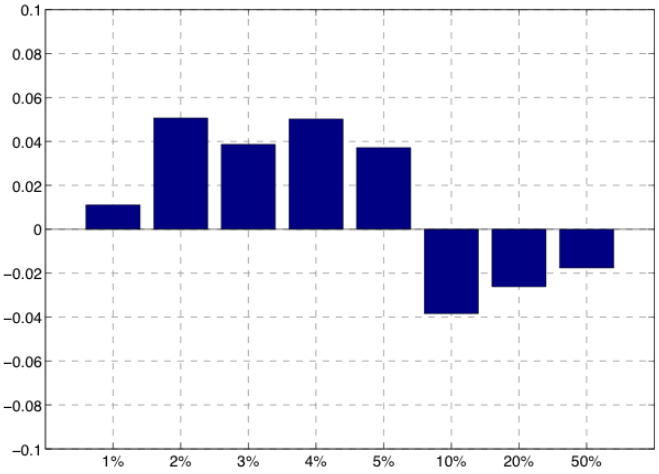



Figure 55: Difference in the accuracies between CTrAdaBoost and the second best technique as an average over all the datasets for different ratios between $\boldsymbol{T_s}$ and $\boldsymbol{T_d}$

One of the theoretical foundations of CTrAdaBoost is the ability to increase the weights of different distribution samples. Figure 56 illustrates a manifestation of this property by plotting the cumulative weights of samples in $T_d$ for different iterations for a particular trial of the Rec Vs Talk classification problem. The corresponding cumulative weights for $T_s$ samples are presented in Figure 57. It is very evident that during the initial iterations of CTrAdaBoost, there is an increase in the weights of $T_d$ samples. It is also interesting that during latter iterations, the total weight of $T_s$ samples tends towards1, while those of the $T_d$ samples tend towards 0. This indicates that the base learner focuses more and more on learning the same distribution samples as the number of iterations increases.

In contrast, the weights of $T_d$ samples never increase in TrAdaBoost, they either decrease and remain at a constant value. Furthermore, cumulative weights of the different distribution samples are significantly higher than those of $T_s$ samples. This forces the base learner to concentrate on learning the $T_d$ data samples, rather than the relevant $T_d$ and $T_s$ samples. We believe that this 'plateau' like behavior of TrAdaboost explains its poor performance. This type of trend was also noticed for the AdaBoost. Since there is no clear distinction between $T_d$ and $T_s$ samples, the former tends to dominate over the latter as indicated by the high cumulative weights for $T_d$ samples.

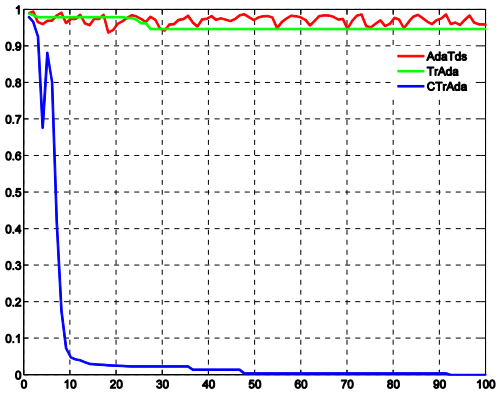



Figure 56: Trace of sum of weights for samples in $\boldsymbol{T_d}$ across boosting iterations

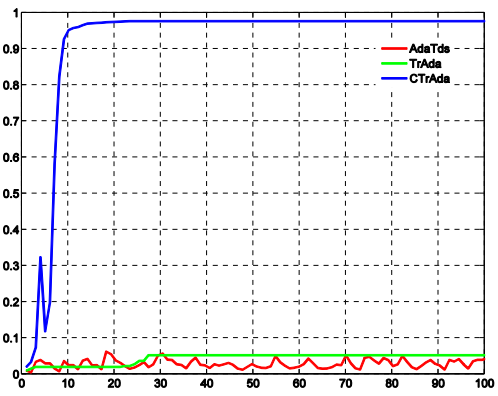



Figure 57: Trace of sum of weights for samples in $\boldsymbol{T_s}$ across boosting iterations

### 6.5.4 Limitations

One of the main limitations of the CTrAdaBoost algorithm is its dependence on reliable cost factors attached to the different distribution samples. In this work, the cost factors were estimated through instance pruning through a model learnt on the same distribution training samples. When the same distribution training samples contain outliers or noisy samples, the cost estimation process gets affected. This cost might not be suitable for training.

While the weights of the same distribution samples converging towards one (Figure 57) assists in the learning process, this can also lead to over fitting. In the current work an early termination criteria based on the change in the training error of the different distribution dataset was used to prevent over fitting. This is only a heuristic and has to be manually tailored through cross validation and hence does not guarantee consistent performance across different datasets.

## 6.6 Conclusion

This chapter introduced the concept of transfer learning and discussed how it was being used in the context of activity gesture recognition. It proposed a cost sensitive based boosting technique for transfer learning. The effectiveness of the algorithm on small sample datasets was demonstrated using the activity gesture datasets collected in this work and on the standard machine learning bench marking datasets. The results obtained on the activity gesture dataset indicate the merit of using transfer learning algorithms for activity recognition to learn activity

models from other data sources, when training data is scarce. The theoretical analysis of the algorithm presented in this work demonstrates the convergence and error bounds of the algorithm. The similarity of the proposed algorithm with other recently proposed techniques for transfer learning through boosting has also been discussed.

The performance of the proposed algorithm is better than other approaches when the number of training samples in the same distribution set is very small (around 1-5%) when compared with different distribution training set. The sensitivity of the algorithm towards training samples in the distribution dataset is one of the limitations of the algorithm. In future, we intend to design and develop unsupervised means of estimating the cost factors for different distribution data by considering the entire set of unlabeled same distribution data.

The framework can be further improved by adding a more robust cost estimation algorithm that probably also takes into account the large number of unlabeled target domain data. In addition, an adaptive cost estimation framework can be designed that changes the costs for the source domain data after every boosting iteration. The impact of this modification on the theoretical properties of the framework have to be studied.

Chapter 7

COMPLEX ACTIVITY TASK PREDICTION

## 7.1 Introduction

Complex activities such as cooking activities typically contain several sub-activities (which we call tasks) within themselves. When a person starts a cooking activity, he or she accomplishes it by following a particular order of doing its constituent tasks. For example consider the activity of making brownies as illustrated in Figure 58 that involves nearly 30 different tasks. Every individual has his/her own unique way of performing these tasks. This results in lot of variability in the order of occurrence of these tasks across individuals and contexts. Learning the activity models for these complex activities from such data posed a challenging problem. While the learning of the sequence of the tasks might not be important for recognizing activities, it forms an essential aspect for applications that require tracking of the activities. For example, in applications such as a prompting system for individuals with Alzheimer's disease (AD) or dementia, the activity models must also be capable of predicting the task the individual will perform even before its occurrence.

The problem of activity task prediction can be defined as follows: given that a person has started a cooking activity and has completed a set of tasks, what is the next most probable task that he or she is likely to do. In contrast, the activity recognition problem focuses on identifying the task

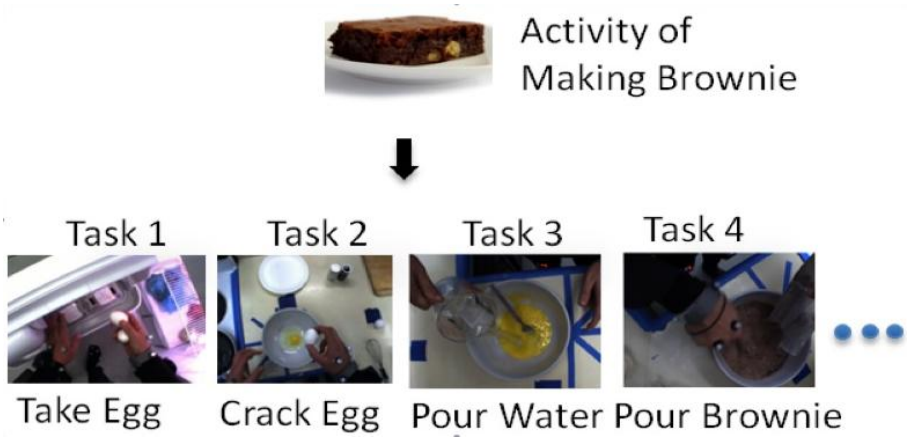



Figure 58: Tasks in the complex activity of making brownies

based on some sensor data. The focus of this chapter is only on activity task prediction and not on activity recognition.

Learning an activity model that performs reliable activity task prediction is beneficial in many ways. Firstly, certain applications require activity task prediction such as prompting systems for individuals with AD or dementia. These applications track the progress of an activity and present cues to individuals depending on the context for assisting in the completion of the activity. Secondly, reliable task prediction can in turn be used to reduce the search space for the recognition engine. For instance, if it can be reliably predicted that the following task is stirring with a spoon, the accelerometer based gesture recognition system can narrow down to gestures using a spoon or similar to stir. This helps in improving the recognition accuracies of these systems. Finally, task prediction also helps in designing reconfigurable systems that can save energy by switching on only the necessary sensors.

Any task prediction framework depends on the underlying activity model. In this chapter we present an activity model that has been developed using a novel methodology called the Hierarchical state space Markov Chain (HMC) that models the temporal relations between tasks. The models proposed have been evaluated on complex cooking activities viz. making brownie and making eggs. The work presented in this chapter is first of its kind that focuses on activity task prediction. Within this context, the work makes the following contributions:

- It proposes a methodology for reliably learning activity models from user data. While the methodology is evaluated with two complex activities, it can be adapted to any complex activity.
- It presents a task prediction framework that can reliably predict tasks in complex activities.

Section 7.2 describes some of the related work in the area of activity modeling. The proposed hierarchical state space Markov chain is described in Section 7.3. It also discusses the algorithm for predicting tasks in complex activities. The details of the experimental evaluation of the proposed models, the description of the datasets used to the conduct the studies along with a

discussion on the results are presented in Section 7.4.. Section 7.5 concludes the work presented in this chapter.

### 7.2 Related Work

Activity recognition is a well-researched area and there are a number of approaches in the literature that map the sensor data streams to either tasks or directly to the complex activity. However predicting activity tasks even before it has occurred is still an under explored research area. Rule based reasoning and probabilistic models are the commonly used approaches for activity task prediction. Hoey et al. [1] describe a probabilistic decision theoretic framework for prompting tasks involved in the activity of washing hands using partially observable Markov decision process. Jakkula et al [2] discuss an approach for event prediction by developing rules using temporal relations between different events. The events discussed in their work are low-level sensor triggers such as oven on, lamp off and not the activity tasks. Gopalratnam and Cook [3] introduce the Active LeZi prediction algorithm based on information theory for predicting a sequence of events that can be modeled as a stochastic process. All these approaches have primarily concentrated on the activity of washing hands that consists of a maximum of 10 tasks. The variations in the sequence of execution of the tasks increased with increasing number of tasks questions the scalability of these approaches to bigger activities. In such scenarios, a single layered task prediction model is insufficient.

Beaudoin et al [201] discuss extraction of common patterns from Inertial Measurement Units (IMU) data and development of motion motif graphs that represent the various actions in an activity along with their transitions and durations. Singla et al [164] have used data from activation, water and heat sensors in a smart home and have constructed activity models using Markov models that include the time duration of events. Yang et al. [202] in their paper have used examples of plans, which have been developed in the context of AI, as their activity data. Using these plans they have constructed action model for each action within an activity. Spriggs et al [127] have extracted the frame wise video features from a first person camera and used PCA to project the features into a low dimensional space and have used GMMs to cluster related frames

and HMMs to classify 2 cooking activities of making brownie and omelet. In their AutoReminder application [203], Pollack et al have modeled activities using Quantitative Dynamic Bayesian Networks considering the activity of each client as a new plan. Activity models that consider objects involved have been detailed in [163] where they have divided each activity into stages i.e. an activity like making tea was divided into boil water, add tea, add flavor. Activity models were developed using dynamic Bayesian networks by considering stages as the hidden variables and objects as observed variables.

Activity models thus learnt have commonly been used for automated activity recognition. Activity models can be used to validate new plans in the context of planning and scheduling in AI as presented in [203]. Isbell et al. [204] in their paper have developed activity models to predict the most probable task in the perspective of related commands while operating devices in a ubiquitous environment. In [204] they have grouped related commands into a task using KNN and used Markov chains to predict next probable task. The work presented in this chapter is a first of its kind, which attempts at developing the activity models based on user data for the purpose of reliable task prediction.

### 7.3 Hierarchical Markov Chain based prediction

A Markov model is a stochastic model where the state that a system occupies during a given time step is dependent on the states it has occupied in the past time steps. The Markov chain is a specific case where the number of time steps that affect the next state of the system is 1. The individual tasks in the activity can be considered as the state of the activity in the context of a Markov model. The sequence of past tasks defines the next task in the sequence. However if we proceed with a Markov chain assumption, then the next task only depends on the immediately preceding task. This does not necessarily hold good for human activity.

We observe that in human activity there is a dependence of the next task on the immediately preceding task as well as a group of past tasks. Thus one solution to capture this temporal relationship would be to use higher order Markov models. Higher order Markov models are typically defined by a large parameter space, which is difficult to learn in the absence of sufficient

number of training samples. The solution to this problem can be approached in two ways. In this first approach, an alternate learning strategy as described in [205] can be employed that reduces the parameters to be learned for defining the higher order Markov model. In this approach a model called as Mixture Transition Distribution, first introduced by Raftery et al [205], is proposed that can parsimoniously represent a Markov chain and whose parameters could be learnt more efficiently than a usual representation of a simple Markov chain. Let us consider a second order Markov chain consisting of 3 states. This represents the scenario where the current state depends on two of its predecessors. The number of elements of the transition matrix for this model is 81. Estimating all the 81 parameters using training data is infeasible. Raferty et al., [205] overcome this problem by defining a new set of parameters called *lag variables* by $\lambda_g$where $g$ defines the time lag between current state and a state occurring in previous time steps. For example if the order of the Markov chain is 3 then there will be 3 lag variables $\lambda_1, \lambda_3$ and $\lambda_3$ defining the dependence of the state at current time $t$ on the states at time steps $t-1, t-2$ and $t-3$ respectively. For a Markov chain of order $l$, the likelihood of occurrence of a state $X_t$ at a time $t$ conditioned on the states occurring during the previous $l$steps is given by the formula*:*

$$P(X_t = s_n / X_{t-l} = s_l, \cdots, X_{t-1} = s_1) = \sum_{k=1}^{l} \lambda_k q_{s_k s_n}$$

where $q_{ij}$'s are just the transition probabilities of a first order markov chain. Thus the number of parameters has decreased from $O(m^l)$ to $O(m^2 + l)$ which is a predominant decrease in the number of parameters to be learnt. For details regarding estimation of the lag parameters, the reader is requested to refer [206].

In the second approach, the state space of the Markov model can be increased to reflect the higher order relationships. This approach is adopted in the proposed hierarchical state space Markov model. The state space of the new Markov model has been transformed such that the state captures not only the immediate preceding task information but also the information about other tasks that have already occurred. The tasks of the activity form the original state space. Each task is transformed into a hierarchical state and these new states form the hierarchical state space. The

next sub section describes in detail this process of transforming a sequence of tasks into the new hierarchical state space.

### 7.3.1 Clustering of tasks

The activity is initially modeled using a Markov chain with the tasks as the states. A transition matrix for the Markov chain, $TM_{MC}$ is constructed from the training data which is a set of sequences of tasks as done by various individuals.

$$TM_{MC}(i,j) = \mathrm{P}\left(s_{n+1} = T_j / s_n = T_i\right)$$

where $TM_{MC}(i,j)$ is the probability of transiting from task $T_i$ to task $T_j$ and $s_n$ is the state at time $n$. Among all the tasks, there are some tasks that transit mostly within themselves i.e these tasks occur in close intervals of time and are related to each other. In order to extract such groups of tasks, the $TM_{MC}$ is plotted as a graph with the tasks as the nodes and the transitions as edges. Since the data consists of activity data from different people there will be many possible transitions while most of them being weak transitions. A threshold probability is applied on the graph and all the edges whose probabilities are less than the threshold are removed. From the new graph all the weakly connected components are identified and each weakly connected component is considered as a cluster and the tasks within each connected component are assigned to that cluster. These task clusters are represented as $\{C_1, C_2, \cdots C_M\}$. Figure 59 shows the clusters in the graph for the activity Making Eggs after applying the threshold on the graph. The colored tasks belong to clusters with more than one task. It can be observed from the graph that tasks like take-spatula, switch-on-stove etc. can occur very randomly at any time during the activity and so they belong to no cluster and their prediction is more uncertain than other tasks.

Each of the clusters can be considered as a super task, where the constituent tasks together accomplish a larger task called as super task. For example, in the clusters illustrated in the Figure 59, the green cluster corresponds to the super task of ‘pouring oil into the pan’; the blue cluster represents the super task of ‘placing an egg on the counter’, the dark yellow cluster symbolizes the super task of ‘cracking an egg’. It has to be noted that it is only the set of tasks that defines the

super task, not necessarily the order in which they occur. The next sub-section describes how to transform a sequence of tasks into a hierarchical state space representation using the task clusters.




**Figure 59:** The clusters in the activity graph for the complex activity 'making eggs' after applying a threshold. Connected components of the graph that are color-coded belong to a single cluster. Tasks with no color belong to no cluster and indicate the uncertainly of their occurrence during the sequence of the activity tasks.

### 7.3.2 Hierarchical state space

Every task in the sequence corresponding to a trail of a subject, $T_i$ where $i$ is the time step representing a state in the original state space is mapped to a new state □ □□□ which is a triplet defined as

$$HSS_i = \left(T_i, C(T_i), Pos(T_i)\right)$$

where $C(T_i)$ is the cluster to which $T_i$ belongs to and $Pos(T_i)$ is the count of the number of tasks $T_j \ni j \leq i$ and $C(T_j) = C(T_i)$.

As an example consider the tasks **take oil**, **twist-on cap**, **pour oil into bowl** and **stir bowl** that are a part of the activity ***Making Brownie***. Let these tasks belong to a cluster $C = 5$. The transformation from the original state space to hierarchical state space is shown in Table 28 using these tasks as example.

Table 28: Transformation of the original state space to hierarchical state space. (TO- Take Oil, POiB - Pour Oil into the bowl, ToC - Twist on Cap, SB - Stir Bowl

| **Time Step** | **Original State** | **Hierarchical State** |
|---|---|---|
| 1 | TO | (TO, 5, 1) |
| … | … | … |
| 6 | POiB | (POiB, 5, 2) |
| 7 | ToC | (ToC, 5, 3) |
| 8 | SB | (SB, 5, 4) |

Using the above transformation, all the tasks in the training data are converted into hierarchical states. Considering the activity as a Markov chain with the hierarchical state space (HMC), a new transition matrix $TM_{HSS}$ is constructed using the new states. If there are $M$ clusters with an average size of $P$ tasks each, then the maximum number of states will be $MP^2$ when compared to $MP$ states in a simple Markov chain. But in reality this is not the case because within each cluster the tasks occur with a certain amount of order so that the tasks may not occur at more than 4 different positions within a cluster and thus reducing the state space to a maximum of $4MP$.

### 7.3.3 Task prediction algorithm

Once the transition matrix $TM_{(HSS)}$ is obtained from the HMC model, it is used by the task prediction algorithm that is described in Algorithm 2. At each time step $n$, the algorithm returns the next most probable task $T_{n+1}$ and its probability of occurrence $P(T_{n+1})$. The task at current time step $T_n$ is transformed into a hierarchical state using the procedure *TransformCurrentTask* that was described in the previous section. Once the hierarchical state is obtained, all the states to

which the system can transit from this state can be obtained from $TM_{HSS}$ and they become the future states represented by the variable $FutStates$.

Algorithm 2: Task Prediction Algorithm

Algorithm: Task Prediction Algorithm

**Input**: Given the task clusters $\{C_1, C_2, \cdots C_M\}$, the hierarchical states from the training data: $\{HSS_1, HSS_2, \cdots, HSS_P\}$, the transition matrix $TM_{HSS}$ and past tasks $\{T_1, T_2, \cdots T_n\}$ with $n$ being the current time step.

**Procedure**: PREDICT NEXT TASK

1. $HSS_n \leftarrow$ TransformCurrent Task( ) // Transforms the task into hierarchical state space
2. Let $FutStates = \{HSS_1^*, HSS_2^*, \cdots, HSS_{F1}^*\} \ni TM_{HSS}(HSS_n, HSS_i^*) > 0, \forall i = 1, \cdots F1$
3. For $t = 1, \cdots, F1$
   a. $\left(T_t, C(T_t), Pos(T_t)\right) \leftarrow HSS_t^*$
   b. If $T_t \in [\![\{T\}]\!]_1, T_2, \cdots, T_n\}$ then
      i. Remove $HSS_t^*$ from $FutStates$
4. Let the reduced $FutStates = \{HSS_1^*, HSS_2^*, \cdots, HSS_{F2}^*\}$
5. Normalize $\left(P\left(\frac{HSS_1^*}{HSS_n}\right), \cdots P\left(\frac{HSS_{F2}^*}{HSS_n}\right)\right)$
6. $\left(T_{n+1}, C(T_{n+1}), Pos(T_{n+1})\right) \leftarrow HSS_N^*$ , where $N = \arg\max_{i} P\left(\frac{HSS_i^*}{HSS_n}\right)$
7. $P(T_{n+1}) \leftarrow P\left(\frac{HSS_N^*}{HSS_n}\right)$

At this point the elements in the $FutStates$ set also includes tasks that have already occurred in the sequence. So the first level of filtering removes all such tasks that have made their presence in the sequence. It is assumed that the tasks can occur only once during the entire sequence. It has to be noted that the labeling of the tasks ensures that irrespective of how similar the tasks are, if they constitute different set of objects or a different activity gestures, they are given unique labels. In some sense, this process of removing tasks that have already occurred, transforms the undirected graphical model into a unique directed acyclic graph that represents a particular instance of execution of the activity.

Finally, using the $TPM_{HSS}$ the probabilities of the remaining future states from the current state are normalized and the one with maximum probability is selected. Using the task information in the hierarchical state using the equation given above, the most probable task $T_{n+1}$ is obtained and the normalized probabilities from $TPM_HSS$ give $P(T_{n+1})$.

### 7.4 Experimental Evaluation

#### 7.4.1 Dataset description

The CMU Quality of Life Technology Center's human activity data [127] has been used to validate the methods proposed in this paper. The CMU human activity database is a collection of cooking activity data from different subjects and consists of data for activities Making Brownie, Making Eggs, Making Sandwich, Making Pizza and Making Salad. A set of multimodal (video, inertial sensors, RFID tags, Motion Capture, Audio) sensors captures the execution of the activity by a subject. The video data thus collected was used for annotating the tasks in the activity. A task was defined as a combination of action-object-preposition-object such as pour-oil-into-big bowl. The annotated data was available for Making Brownie and we have manually annotated for the activity Making Eggs using the same procedure. The task prediction results presented in this chapter have been obtained by evaluating on these 2 activities.

Table 29: Description of the tasks for each of the two activities

| **Making Brownie:** | **Making Eggs:** |
|---|---|
| close-fridge | clean-with-paper_towel |
| crack-egg | close-egg_box |
| open-brownie_bag | close-fridge |
| open-brownie_box | crack-egg |
| open-fridge | open-egg_box |
| pour-big_bowl-into-baking_pan | open-fridge |
| pour-brownie_bag-into-big_bowl | pour-oil-into-frying_pan |
| pour-oil-into-big_bowl | pour-pepper-into-small_bowl |
| pour-oil-into-measuring_cup_small | pour-salt-into-small_bowl |
| pour-water-into-big_bowl | pour-small_bowl-into-frying_pan |
| pour-water-into-measuring_cup_big | put-egg_shell-into-sink |
| put-baking_pan-into-oven | put-egg-on-plate |
| spray-pam | put-frying_pan-into-sink |
| stir-big_bowl | put-frying_pan-on-stove |
| stir-brownie | put-small_bowl-into-sink |
| stir-egg | stir-small_bowl |
| stir-oil | switch_off-stove |
| stir-water | switch_on-stove |
| switch_on-oven | take-egg |
| take-baking_pan | take-fork |
| take-big_bowl | take-frying_pan |

| | |
|---|---|
| take-brownie_box | take-oil |
| take-egg | take-paper_towel |
| take-fork | take-pepper |
| take-measuring_cup_big | take-plate |
| take-measuring_cup_small | take-salt |
| take-oil | take-small_bowl |
| take-pam | take-spatula |
| twist_off-cap | take-whisk |
| twist_on-cap | twist_off-cap |
| walk-to-counter | twist_on-cap |
| walk-to-fridge | walk-to-counter |
| | walk-to-fridge |

Out of that many tasks that belong to an activity, few tasks like put-oil-into-cupboard are not relevant in achieving the goal of an activity as such. Therefore after ignoring such tasks, the activity Making Brownie has 32 tasks and Making Eggs has 33 tasks, which is significantly higher than the number of tasks considered in other related work. The list of the tasks for each of the activity is listed in Table 29. Abnormal trials of the execution of the activity have been ignored resulting a total of 14 and 11 subject trials for the activities Making Brownie and Making Eggs respectively.

**7.4.2. Comparison methods and measures**

In order to evaluate the properties of the proposed framework two different baseline methods were implemented. The first method was the standard Markov chain model. This was modified slightly to account to ensure that tasks that have already occurred in the sequence are not considered for future predictions. The second method was the mixture transition distribution (MTD) version of a second order Markov model as described by [205], again with the modification to ensure that completed tasks are not considered for future predictions. The implementation of the higher order Markov model was adopted from [206]. The performance of these techniques was compared against the proposed hierarchical state space Markov model. A multi-fold cross validation strategy has been adopted for the evaluation of these techniques. Data from all but one subject is considered for training and is tested against the data of the left out subject. This is done in a round

robin fashion resulting in 14 fold for the activity 'Making Brownie' and 11 fold for the activity 'Making Eggs'.

Two different measures were used to compare the performance of the three different approaches namely: matching accuracy and DTW (Dynamic Time Warping) distance measure. While the matching accuracy measure compares the overall prediction accuracies of the techniques, the DTW distance measure compares the step-by-step performance of the approaches. The test sequence is provided to the task prediction algorithm one task at a time in the order of occurrence. The task prediction algorithm predicts the next probable task with some probability. If the predicted task is the same as the actual next task in the test sequence the matching accuracy is increased. Thus the final matching accuracy is the ratio of the total number of matches to the total tasks predicted.

While calculating matching accuracy, the current task from the original test sequence is provided and using all the tasks of the original test sequence till then, only the next task is predicted. But in order to test how realistically the model can predict and model rest of the entire activity itself the DTW distance measure has been used. Let $\{TT_1, TT_2, \cdots TT_{32}\}\}$ be the actual test sequence of length 32 where $TT$ denotes Test task. A partial sequence $\{TT_1, TT_2, TT_3, TT_4\}$, say of length 4 is considered and the proposed algorithm is used to predict the rest of the sequence and let $\{PT_5, PT_6, \cdots, PT_{32}\}$ be the predicted sequence where $PT$ denotes Predicted task. Using this strategy, by considering only the actual first task, the rest of the activity can be generated and thereby the recipe of the activity can automatically be generated. Now the actual sequence $\{T_5, T_6, \cdots, T_{32}\}$ and the predicted sequence $\{PT_5, PT_6, \cdots, PT_{32}\}$ are compared and the distance between these sequences is calculated using Dynamic Time Warping (DTW). The distance between any matched tasks is considered to be 0 and unmatched task as 1. For the above-mentioned test sequence since 32 partial sequences are possible 32 DTW distances are calculated and the average has been considered.

### 7.4.3 Results and Discussion

The test sequence is provided to the task prediction algorithm one task at a time in the order of occurrence. The task prediction algorithm predicts the next probable task with some probability. If the predicted task is the same as the actual next task in the test sequence the matching accuracy is increased. Thus the final matching accuracy is the ratio of the total number of matches to the total tasks predicted. The final accuracies have been calculated by taking an average of accuracies obtained from 14 test sequences for *Making Brownie* and 11 sequences for *Making Eggs*. The optimal matching accuracies achieved on the 2 activities *Making Brownie* and *Making Eggs* using the proposed HMC model are 0.68 and 0.65 respectively. Figure 60 compares this performance with the accuracies achieved using the other two baseline approaches. As shown in the figure there has been an improvement of around 5% in accuracies for each of the activities. As discussed in Section 7.3.1, to cluster the tasks in HMC model, a threshold is applied on the transition matrix. For the activities *Making Brownie* and *Making Eggs* the thresholds that produced these optimal accuracies are 0.012 and 0.02 respectively. It can be observed that the MTD version of the second order Markov model yielded the lowest performance. The reason for the poor performance of the MTD model can be attributed to the limited training data and not much dependence of the current state on states that occur before more than 1 time step. It was observed that the weights for the first and second time steps i.e., the lag variables were estimated as 1 and 0 which implies that only one time step is effecting the current state and for some instances the coefficients where 0 for last time step and 1 for the last but one time step. These inconsistencies are primarily due to lack of training data and since maximization is done using gradient descent etc, MTD model needs more data to come up with valid coefficients.

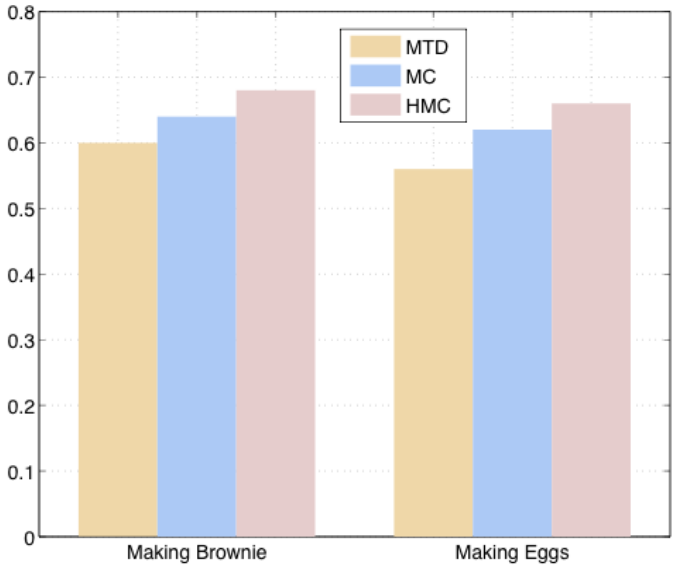



Figure 60: Matching accuracies for the task prediction for Markov Chain (MC), Mixture Transition Distribution (MTD) and Hierarchical state space Markov Chain (HMC)

Figure 61 shows the DTW distances computed on an average for the different execution trials of the two activities using Markov chain (MC), MTD and proposed algorithm (HMC). For each trial the number of partial sequences is equal to the number of tasks involved in it. Algorithms with better performance tend to have lower DTW distance measures. The results presented in Figure 61 further verify the effectiveness of the proposed HMC model over the other two baseline approaches. The distance measure on the Y axis is an average over all the partial sequences. To further understand the behavior of the algorithms, there are 28 partial sequences and the 28 corresponding DTW distances are normalized by the lengths of the partial sequences. It can be observed that the DTW distance of the sequences generated by the HMC models are consistently lesser when compared to MC or the MTD model. In fact the DTW distances for MTD increase as the sequence lengths increase whereas the DTW distances for HMC models decrease because, as the lengths of partial sequences increase there is more knowledge about the tasks that are completed.. We also observe that for HMC, the DTW distances do not decrease monotonically i.e. there are spikes in the distances. The reason is that even though the difference between 2 immediate partial sequences is only 1 task, that single task gives a completely different path to the completion of the activity, which results in larger distances.

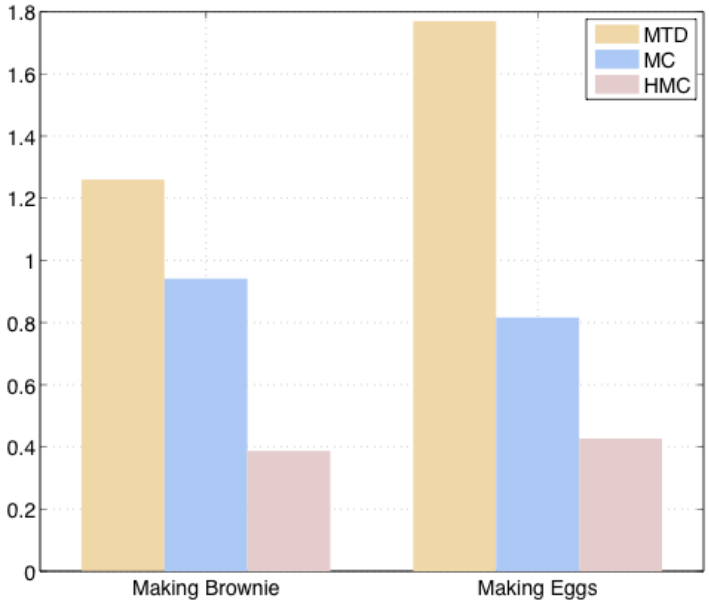



Figure 61: Average DTW distance measures across all the folds for the complete task sequence prediction for the three models

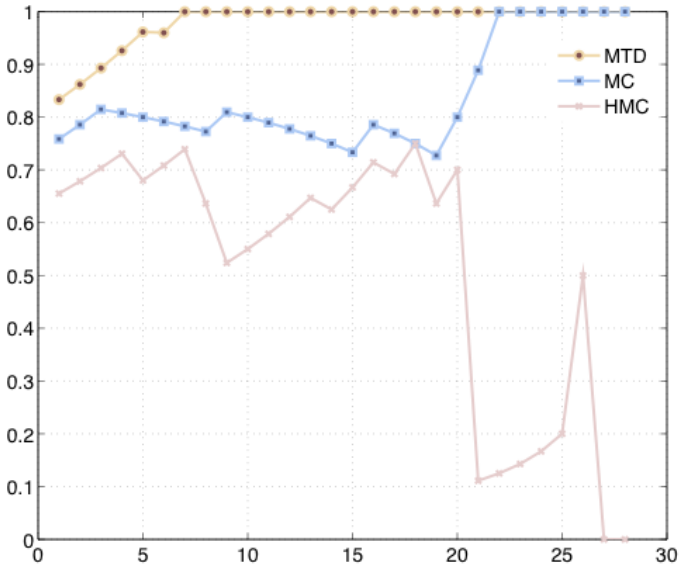



Figure 62: Normalized DTW distance measure for each of the partial sequences for a particular subject for the activity Make Eggs

### 7.5 Conclusion

This chapter focuses on the problem of task prediction for complex cooking activities that consist of multiple steps or tasks. The temporal relations between the different tasks have been modeled

using the proposed hierarchical state space Markov chain (HMC). The hierarchical state space takes into account the higher order temporal relationship between the individual tasks by expanding the state space to encode these relationships. An algorithm for predicting that next probable task using the HMC model and a given sequence of tasks has also been proposed. The performance of the proposed algorithms have been compared against two other baseline techniques namely the Markov chain and Mixture transition distribution model . The models and algorithms have been evaluated on two complex cooking activities that are a part of the CMU kitchen dataset resulting in an average subject independent task prediction accuracy of 0.67, which is better than the traditional Markov chain model by over 4% and is far superior over a random guess that gives a prediction accuracy of 7% only. The work presented in the chapter is the first of its kind focusing on activity models for task prediction. The models proposed in this work can further be used for automatically creating recipes from past cooking data customized to individuals for assisting in activity completion.

While the proposed algorithms show promising results, there is still a vast scope for improvement as can be noted by the distances computed for the partial sequences. The HMC model captures the temporal relations between the tasks, but ignores other associations such as shared object usage. Models that capture these relations can be integrated into the proposed framework for improving the performance of task prediction algorithms. Furthermore, the time duration of each task was considered to be constant in this work. Future work could also focus on determining appropriate distributions that model the varying time intervals for each task.

Chapter 8

# CONCLUSION

This dissertation presented a computational framework for wearable accelerometer based activity and gesture recognition. It addressed each of the five objectives related to activity and gesture recognition using accelerometers discussed in Chapter 1. It makes the following conclusions:

- ✓ This dissertation evaluated the effectiveness of different discriminative classifiers for simple activity recognition from low-resolution accelerometer data. Table 6 in chapter 3 summarizes the results obtained by different researchers for continuous recognition for a similar set of activities considered in this work. The last two rows of this table are the accuracies that were obtained using the methodology proposed, in this dissertation.. The results clearly illustrate the superior performance of discriminative classifiers such as AdaBoost and SVM over other classifiers. The proposed methodology achieves better performance in-terms of the accuracy, with a minimum number of accelerometers.
- ✓ Based on these results, a real-time simple activity recognition system was designed and implemented. The proposed real-time system is able to accurately recognize the simple activities in real-time achieving a performance superior to existing approaches in the literature. The complexity of the system was discussed both in terms of the computational complexity and normalized time for computation for classifying a single sample.
- ✓ The activity gesture datasets presented in this dissertation, though limited to a finite number of activities performed in a constrained scenario, represent a wide range of different hand –object interactions occurring in complex activities of daily living. The work presented in the dissertation establishes the feasibility of accelerometer based activity gesture recognition. The results obtained from discriminative classifiers as discussed in Chapter 4 again illustrates the effectiveness of discriminative classifiers for accelerometer based gesture recognition.
- ✓ Chapter 5 discussed the problem of detecting out-of-vocabulary samples in the context of activity gesture recognition. It proposes the AdaBoost and multiple one class SVM based

adaptive threshold models trained only on in-vocabulary label samples for detecting out-vocabulary label samples. Both the models yield better performance over their respective traditional classifier models. Though the proposed models targeted towards the problem of activity gestures spotting, the algorithms are applicable in other problem domains. The reliability of the multiple one-class SVM over traditional models (such as SVM, GMM and Large margin GMM) was established by experimenting over a wide variety of datasets. This particular work validates the fundamental assumption of current pattern recognition and machine learning techniques for modeling the data space and illustrates its inability to handle out-of-vocabulary label samples.

- ✓ A gesture spotting network that combines the adaptive threshold models and in-vocabulary label classifiers is proposed for spotting gestures from continuous accelerometer data streams. The proposed approach resulted in better performance over the traditional HMM based approach. This study seems to suggest that given the context of an activity and in the absence of other sensors; accelerometers can used independently to reliably detect the activity gestures, for tracking the progress of the activity.
- ✓ Gathering sufficient amount of real-world training data is difficult and at times infeasible It proposes a unique framework for compensating for lack of real-world data for training activity models. The framework hinges on collecting data from auxiliary sources and employs transfer learning algorithms to assist in the supervised learning process. The theoretical analysis of the framework illustrates the convergence properties and training and generalization error bounds of the algorithm. The effectiveness of the framework has also been shown on other bench mark datasets.
- ✓ Most of the research on activity recognition (including a major part of this dissertation) has focused on the problem of recognizing activities from sensor streams. The word described in Chapter 7 addressed a very unique problem of predicting activity tasks in complex activities. It shows the possibility of predicting the activity task ahead in time, independent of the subject executing the activity, with an accuracy of about 70%. This

hold tremendous promise to the activity recognition community in many ways such as reducing the activity recognition error rates by restricting the possible set of labels; providing a mechanism for identifying what cues have to be prompted to individuals with mild memory impairment; and for designing energy efficient activity recognition systems that switch on only a relevant set of sensors at a given point in time.

**8.1 Limitations and future directions**

While the proposed solutions for each of the objectives are promising, there are certain limitations if overcome will make the solutions even richer. These limitations are discussed below:

- The proposed simple activity recognition system works extremely well when the activity samples are provided in isolation. It however does not model the transitions between the different activities. Certain applications such as fall detection systems necessitate the recognition of these transitions. Adding this capability to the proposed system, will allow for more reliable tracking of ambulation.
- One of the primary requirements of wearable systems is an accurate positioning and alignment of the sensing elements on the body. This is typically performed by either using tight-fitting garments or by directly attaching the sensing element on to the intended location using tapes. However, in many practical applications it is not feasible to impose these constraints. Using comfortable everyday garments for wearable computing remains a challenging research, since a fix sensor position in relation to the body cannot be guaranteed. Even in the scenario where the sensor is attached directly to the body, the variations in the physical form of the person such as height, weight and body mass index can result in erroneous sensor outputs. Compensating for these aberrations in the sensor output will enrich the proposed system and make it more general.
- The gesture spotting system proposed in this dissertation is computationally expensive and requires the feature vector to be updated with the availability of new data samples. The future work could focus on alleviating this problem. The current approach for gesture spotting and recognition is centralized with the data from all the sensors being processed all together. This can be made way for a more distributed system with the data from each

sensor being processed independently, with only the fusion of the results occurring at a centralized location. This facilitates in graceful degradation of the system in case of sensor failure.

- The multiple one class SVM based out-of-vocabulary sample detection model has been experimented on the activity gesture dataset and other non-activity datasets. In the future, we intend to evaluate the performance of this technique on other activity datasets which will include data collected from other sensors. Furthermore it will be worthwhile to integrate the properties of the multiple one-class SVM into the standard SVM model to facilitate both out-of-vocabulary sample detection and in-vocabulary sample classification into a single step.
- .As discussed in Chapter 6, the cost sensitive based boosting approach for transfer learning is susceptible to the quality of training samples in the same distribution set. Developing better cost estimation methods that also take into account the majority of the unlabeled same distribution dataset will be one way to overcome the sensitivity of the algorithm. Furthermore, other methods of adding the cost factor into the training algorithm that reduces the chances of over fitting is another extension that has to be researched.
- The activity task prediction framework proposed in the dissertation assumes that all tasks are performed in equal time steps. However in real-life tasks are typically of unequal duration. This additional dimension in the variation of the data has not been modeled. Adding appropriate stochastic process that model this distribution of temporal variations in duration of tasks will increase the robustness of the proposed system.

**8.3Dissemination and outcomes**

- This dissertation has directly resulted in 1 journal paper (Journal of Ambient Intelligent Systems and Environments) and 10 conference publications (at venues such as ICME, ACM MM, ICASSP and ECML). This work was also presented as a tutorial at International Conference on Multimedia and Expo 2010. In addition, certain aspects of

the work were also presented at the annual conference of Arizona Alzheimer's research consortium and symposium on coaptive learning: adaptive technology for the aging. The dissertation has also resulted in 5 other indirect conference publications (at venues such as BSN, CSUN and VISAPP).

- **Code:** The dissertation has resulted in software for real-time recognition of simple activities. A brief illustration of this software was presented in Chapter 3. A C# based annotation tool for visualizing the accelerometer data synchronized with the video stream was also developed during the course of the dissertation. We intend to release the matlab implementation of the multiple one class SVM and cost sensitive boosting based transfer learning technique very soon for the benefit for the research community.
- **Datasets:** The activity gesture datasets collected during the dissertation have not yet been publicly released. This is something we intend to do in the near future.
- There have been two offshoots of this doctoral work: an ongoing Master's thesis on developing better techniques for estimating costs for the transfer learning algorithm; and an ongoing PhD thesis on modeling the execution time of tasks using stochastic process for activity task prediction.
- The principles of the simple activity recognitions systems have been used for developing a real-time system for detecting stereotypic movements in individuals who are blind such as body rocking. Furthermore the simple activity recognition techniques have also been used to develop a interactive track and field game, where the players wear the sensors and actually run and jump to play the game.
- Parts of the dissertation have also been used to write grant proposals to various funding agencies; Coaptive framework for assisted living - National Science Foundation, Objective quantification of physical fitness by measuring functional movements through on-body accelerometers - Special Operations Command and Cognitive orthotic for prompting tasks in IADL - Alzheimer's Association Everyday technology for Alzheimer Care.

- **Societal outcomes**: While the work presented in this dissertation was addressed mainly towards the needs of elderly individuals or individuals with Alzheimer's disease, we believe that it can be applied with modifications to many other contexts. For example, it can be used to assist individuals who are blind and visually impaired to experience an enriching shopping experience. The ambulatory movement recognition can be combined with location estimation systems for understanding the movements of the individuals in a shopping space and prompt the daily specials or information about a new product. The wearable RFID reader initially designed towards object identification for activity recognition can be used to provide product information to an individual who is blind.

# REFERENCES


[1] Hoey, Jesse, et al., "Assisting persons with dementis during handwashing using a partially observable Markov decision process." 2007.

[2] J V. R. Jakkula, A. S. Crandall, and D. J. Cook, "Knowledge discovery in entity based smart environment resident data using temporal relation based data mining,"in ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, pp. 625–630, 2007.

[3] Gopalratnam, Karthik and Cook, Diane J., "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm." Intelligent Systems, IEEE, 2007, Issue 1, Vol. 22, pp. 52-58.

[4] J. B. J. Bussmann, W. L. J. Martens, J. H. M. Tulen, F. C. Schasfoort, H. J. G. Van Den Berg-Emons, and H. J. Stam, "Measuring daily behavior using ambulatory accelerometry: The Activity Monitor," Behavior Research Methods, Instruments, and Computers, vol. 33, no. 3, pp. 349–356, 2001.

[5] Bonato, Paolo., "Advances in wearable technology and applications in physical medicine and rehabilitation." Journal of NeuroEngineering and Rehabilitation, 2005, Issue 1, Vol. 2, p. 2.

[6] A. Salarian, H. Russmann, F. Vingerhoets, C. Dehollain, Y. Blanc, P. Burkhard, and K. Aminian, "Gait assessment in parkinson's disease: toward an ambulatory system for long-term monitoring," Biomedical Engineering, IEEE Transactions on, vol. 51, no. 8, pp. 1434–1443, 2004.

[7] N. Bidargaddi, A. Sarela, L. Klingbeil, and M. Karunanithi, "Detecting walking activity in cardiac rehabilitation by using accelerometer," in Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on, pp. 555–560, 2007.

[8] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. Bula, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," Biomedical Engineering, IEEE Transactions on, vol. 50, no. 6, pp. 711–723, 2003..

[9] M. Sekine, T. Tamura, T. Fujimoto, and Y. Fukui, "Classification of walking pattern using acceleration waveform in elderly people," in Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE, vol. 2, pp. 1356–1359, 2000.

[10] P. Veltink, H. Bussmann, W. de Vries, W. Martens, and R. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," Rehabilitation Engineering, IEEE Transactions on, vol. 4, no. 4, pp. 375–385, 1996.

[11] M. S. F. Foerster and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," Computers in Human Behavior, vol. 15, no. 5, pp. 571 – 583, 1999.

[12] K. Aminian, P. Robert, E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," Medical and Biological Engineering and Computing, vol. 37, no. 3, pp. 304–308, 1999..

[13] M.-L.Wetzler, J. R. Borderies, O. Bigaignon, P. Guillo, and P. Gosse, "Validation of a two-axis accelerometer for monitoring patient activity during blood pressure or ecg holter monitoring," Clinical and Pathological Studies, vol. 8, no. 6, 2003.

[14] M. Mathie, A. Coster, N. Lovell, and B. Celler, “Detection of daily physical activities using a triaxial accelerometer,” Medical and Biological Engineering and Computing, vol. 41, no. 3, pp. 296–301, 2003.

[15] R. Jafari, W. Li, R. Bajcsy, S. Glaser, and S. Sastry, "Physical Activity Monitoring for Assisted Living at Home." 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), 2007, pp. 213-219.

[16] T. Al-ani, Q. T. L. Ba, and E. Monacelli, “On-line automatic detection of human activity in home using wavelet and hidden markov models scilab toolkits,” in Control Applications, 2007. CCA 2007. IEEE International Conference on, pp. 485–490, 2007.

[17] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,” Information Technology in Biomedicine, IEEE Transactions on, vol. 10, no. 1, pp. 156–167, 2006.

[18] R. Ibrahim, E. Ambikairajah, B. Celler, and N. Lovell, “Time-frequency based features for classification of walking patterns,” in Digital Signal Processing, 2007 15th International Conference on, pp. 187–190, 2007.

[19] C. Tseng and D. Cook, “Mining from time series human movement data,” in Systems, Man and Cybernetics, 2006. SMC ’06. IEEE International Conference on, vol. 4, pp. 3241–3243, 2006.

[20] J. Mantyjarvi, J. Himberg, and T. Seppanen, “Recognizing human motion with multiple acceleration sensors,” in Systems, Man, and Cybernetics, 2001 IEEE International Conference on, vol. 2, pp. 747–752, 2001.

[21] C. Randell and H. Muller, “Context awareness by analyzing accelerometer data,” in ISWC ’00: Proceedings of the 4th IEEE International Symposium on Wearable Computers, p. 175, 2000.

[22] K. V. Laerhoven and O. Cakmakci, “What shall we teach our pants?,” in Proceedings of the 4th IEEE International Symposium on Wearable Computers, p. 77,2000.

[23] J. F. Knight, H. W. Bristow, S. Anastopoulou, C. Baber, A. Schwirtz, and T. N. Arvanitis, “Uses of accelerometer data collected from a wearable system,” Personal Ubiquitous Computing, vol. 11, no. 2, pp. 117–132, 2007

[24] Kern, Nicky, Schiele, Bernt and Schmidt, Albrecht., "Multi-sensor Activity Context Detection for Wearable Computing." Ambient Intelligence, 2003, pp. 220-232.

[25] M. Uiterwaal, E. B. C. Glerum, H. J. Busser, and R. C. Van Lummel, “Ambulatory monitoring of physical activity in working situations, a validation study,” Journal of Medical Engineering and Technology, vol. 22, no. 4, pp. 168–172, 1998

[26] Bao, Ling and Intille, Stephen S., "Activity Recognition from User-Annotated Acceleration Data." Pervasive Computing, 2004, pp. 1-17.

[27] G. Chambers, S. Venkatesh, G. West, and H. Bui, “Hierarchical recognition of intentional human gestures for sports video annotation,” in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2, pp. 1082–1085, 2002.

[28] S. Antifakos, F. Michahelles, and B. Schiele, “Proactive instructions for furniture assembly,” in UbiComp ’02: Proceedings of the 4th international conference on Ubiquitous Computing, pp. 351–360, 2002.

[29] G. Fang, W. Gao, and D. Zhao, “Large vocabulary sign language recognition based on hierarchical decision trees,” in ICMI ’03: Proceedings of the 5th international conference on Multimodal interfaces, pp. 125–131, 2003

[30] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uwave: Accelerometer-based personalized gesture recognition and its applications,” IEEE International Conference on Pervasive Computing and Communications, vol. 0, pp. 1–9, 2009.

[31] Zinnen, Andreas, Laerhoven, Kristof van and Schiele, Bernt., "Toward Recognition of Short and Non-repetitive Activities from Wearable Sensors." Ambient Intelligence, 2007, pp. 142-158.

[32] H. Junker, O. Amft, P. Lukowicz, and G. Tr¨oster, “Gesture spotting with bodyworn inertial sensors to detect user activities,” Pattern Recognition., vol. 41, no. 6,pp. 2010–2024, 2008.

[33] Lester, Jonathan, Choudhury, Tanzeem and Borriello, Gaetano., "A Practical Approach to Recognizing Physical Activities." Pervasive Computing, 2006, pp. 1-16.

[34] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, “Activity recognition of assembly tasks using body-worn microphones and accelerometers,” IEEE Trans.Pattern Analysis and Machine Intelligence, vol. 28, no. 10, pp. 1553–1567, 2006

[35] Amft, Oliver and Tr\"oster, Gerhard., "On-body sensing solutions for automatic dietary monitoring." IEEE Pervasive Computing, 2009, Issue 2, Vol. 8, pp. 62-70.

[36] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, “Learning and inferring transportation routines,” Artificial Intelligence, vol. 171, no. 5-6, pp. 311 – 331, 2007.

[37] Kasteren, Tim van, Englebienne, Gwenn and Kr\"{o}se, Ben., "Recognizing Activities in Multiple Contexts using Transfer Learning." 2008.

[38] Tapia, Emmanuel Munguia, Intille, Stephen S. and Larson, Kent., "Activity Recognition in the Home Using Simple and Ubiquitous Sensors." Pervasive Computing, 2004, pp. 158-175.

[39] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, “Fine-grained activity recognition by aggregating abstract object usage,” in ISWC ’05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers, pp. 44–51, 2005..

[40] Moeslund, Thomas B. and Bajers, Fredrik., "Computer Vision-Based Human Motion Capture - A Survey." Computer Vision-Based Human Motion Capture - A Survey. 1999.

[41] Gavrila, D. M., "The visual analysis of human movement: a survey." Computer Vision and Image Understanding, 1999, Issue 1, Vol. 73, pp. 82-98.

[42] R. Messing, C. Pal, and H. Kautz, “Activity recognition using the velocity histories of tracked keypoints,” in ICCV ’09: Proceedings of the Twelfth IEEE International Conference on Computer Vision, 2009.

[43] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity recognition and monitoring using multiple sensors on different body positions,” in BSN ’06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, pp. 113–116, 2006..

[44] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in IAAI’05: Proceedings of the 17th conference on Innovative applications of artificial intelligence, pp. 1541–1546, 2005.

[45] F. Allen, E. Ambikairajah, N. Lovell, and B. Celler, “An adapted gaussian mixture model approach to accelerometry-based movement classification using timedomain features,” in Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE, pp. 3600–3603, 2006

[46] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, “A hybrid discriminative/generative approach for modeling human activities,” in IJCAI’05: Proceedings of the 19th international joint conference on Artificial intelligence, pp. 766–772, 2005..

[47] Yang, Ming Hsuan, Ahuja, Narendra and Tabb, Mark., "Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition." IEEE Trans. Pattern Analysis and Machine Intelligence, 2002, Issue 8, Vol. 24, pp. 1061-1074.

[48] V.-M. Mantyla, J. Mantyjarvi, T. Seppanen, and E. Tuulari, “Hand gesture recognition of a mobile device user,” in Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on, vol. 1, pp. 281–284, 2000.

[49] Suutala, Jaakko, Pirttikangas, Susanna and R{\"o}ning, Juha., "Discriminative Temporal Smoothing for Activity Recognition from Wearable Sensors." Ubiquitous Computing Systems, 2009, pp. 182-195.

[50] Suutala, Jaakko, Pirttikangas, Susanna and R{\"o}ning, Juha., "Discriminative Temporal Smoothing for Activity Recognition from Wearable Sensors." Ubiquitous Computing Systems, 2009, pp. 182-195.

[51] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D’Alessio, “Classification of motor activities through derivative dynamic time warping applied on accelerometer data,” in Engineering in Medicine and Biology Society, EMBS 2007. 29th Annual International Conference of the IEEE, pp. 4930–4933, 2007..

[52] Krishnan, N.C. and Panchanathan, S., " Analysis of low resolution accelerometer data for continuous human activity recognition,” in Acoustics, Speech and Signal Processing, ICASSP 2008. IEEE International Conference on, pp. 3337–3340, 2008

[53] N. C. Krishnan, J. Colin, C. Dirk, and S. Panchanathan, “Real time human activity recognition using tri-axial accelerometers,” in Senor Signal and Image Processing Workshop, 2008.

[54] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kr¨ose, “Accurate activity recognition in a home setting,” in UbiComp ’08: Proceedings of the 10th international conference on Ubiquitous computing, pp. 1–9, 2008.

[55] Sakoe, Hiroaki., "Dynamic programming algorithm optimization for spoken word recognition." IEEE Transactions on Acoustics, Speech, and Signal Processing, 1978, Vol. 26, pp. 43-49.

[56] A. Corradini, “Dynamic time warping for off-line recognition of a small gesture vocabulary,” in RATFG-RTS ’01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems(RATFG-RTS’01), p. 82, 2001.

[57] T. Darrell and A. Pentland, “Space-time gestures,” in Computer Vision and Pattern Recognition, 1993. Proceedings CVPR ’93., 1993 IEEE Computer Society Conference on, pp. 335–340, 1993.

[58] S. A. Niyogi and E. H. Adelson, “Analyzing and recognizing walking figures in xyt,” in IEEE Conference on Computer Vision and Pattern Recognition, pp. 469–474, 1994..

[59] D.M. Gavrila and L. S. Davis, “Towards 3-d model-based tracking and recognition of human movement: a multi-view approach,” in In International Workshop on Automatic Face- and Gesture-Recognition. IEEE Computer Society, pp. 272–277,1995.

[60] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, “The function space of an activity,” in CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 959–968, 2006.

[61] Alon, Jonathan, et al., "A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, Vol. 31, pp. 1685-1699.

[62] Starner, Thad, Pentland, Alex and Weaver, Joshua., "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video." IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, Issue 12, Vol. 20, pp. 1371-1375.

[63] C. Bregler, “Learning and recognizing human dynamics in video sequences,” in CVPR ’97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR ’97), p. 568, 1997.

[64] Oka, Kenji, Sato, Yoichi and Koike, Hideki., "Real-Time Fingertip Tracking and Gesture Recognition." IEEE Comput. Graphics Applications, 2002, Issue 6, Vol. 22, pp. 64-71.

[65] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy, “The gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring,” in Wearable Computers, 2000. The Fourth International Symposium on, pp. 87–94, 2000

[66] K. Kahol, N. C. Krishnan, V. N. Balasubramanian, S. Panchanathan, M. Smith, and J. Ferrara, “Measuring movement expertise in surgical tasks,” in MULTIMEDIA ’06: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 719–722, 2006

[67] Wilson, Andrew D. and Bobick, Aaron F., "Parametric Hidden Markov Models for Gesture Recognition." IEEE Trans. Pattern Anal. Mach. Intell., 1999, Issue 9, Vol. 21, pp. 884-900.

[68] M. Brand, “Shadow puppetry,” in ICCV ’99: Proceedings of the International Conference on Computer Vision-Volume 2, p. 1237, 1999

[69] T.-S. Wang, H.-Y. Shum, Y.-Q. Xu, and N.-N. Zheng, “Unsupervised analysis of human gestures,” in PCM ’01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, pp. 174–181, 2001

[70] F. Lv and R. Nevatia, “Recognition and segmentation of 3-d human action using hmm and multi-class adaboost,” in ECCV’06: Proceedings of the 9th European Conference on Computer Vision, pp. 359–372, 2006

[71] Ghahramani, Zoubin., "Learning Dynamic Bayesian Networks." Lecture Notes in Computer Science, 1998, Vol. 1387, pp. 168--.

[72] Kahol, Kanav, Tripathi, Priyamvada and Panchanathan, Sethuraman., "Automated Gesture Segmentation From Dance Sequences." IEEE International Conference on Automatic Face and Gesture Recognition, 2004, Vol. 0, p. 883.

[73] S. Park and J. K. Aggarwal, “Recognition of two-person interactions using a hierarchical bayesian network,” in IWVS ’03: First ACM SIGMM international workshop on Video surveillance, pp. 65–76, 2003.

[74] P. Peursum, G. West, and S. Venkatesh, “Combining image regions and human activity for indirect object recognition in indoor wide-angle views,” in ICCV ’05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1, pp. 82–89, 2005

[75] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, “Learning and detecting activities from movement trajectories using the hierarchical hidden markov models,” in CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 2, pp. 955–960, 2005.

[76] N. Jojic, N. Petrovic, B. J. Frey, and T. S. Huang, “Transformed hidden markov models: Estimating mixture models of images and inferring spatial transformations in video sequences,” in In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 26–33, 2000.

[77] Toyama, Kentaro and Blake, Andrew., "Probabilistic Tracking with Exemplars in a Metric Space." Int. J. Comput. Vision, 2002, Issue 1, Vol. 48, pp. 9-19.

[78] Altun, Yasemin., "Discriminative Methods for Label Sequence Learning." PhD thesis, Brown University, 2005.

[79] Sha, Fei and Saul, Lawrence K., Advances in Neural Information Processing Systems 19. 2007, pp. 1249-1256.

[80] J.-C. Chen and J.-T. Chien, “Bayesian large margin hidden markov models for speech recognition,” in ICASSP ’09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3765–3768, 2009.

[81] Cui, Yuntao and Weng, Juyang., "Appearance-Based Hand Sign Recognition from Intensity Image Sequences." Computer Vision and Image Understanding, 2000, Issue 2, Vol. 78, pp. 157-176.

[82] Ong, Eng-Jon and Bowden, Richard., "A Boosted Classifier Tree for Hand Shape Detection." Automatic Face and Gesture Recognition, IEEE International Conference on, 2004, Vol. 0, p. 889.

[83] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose, “Common sense based joint training of human activity recognizers,” in IJCAI’07: Proceedings of the 20th international joint conference on Artifical intelligence, pp. 2237–2242, 2007

[84] Yang, Hee-Deok, Sclaroff, Stan and Lee, Seong-Whan., "Sign Language Spotting with a Threshold Model Based on Conditional Random Fields." IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, Vol. 31, pp. 1264-1277.

[85] L.-P. Morency and T. Darrell, “Head gesture recognition in intelligent interfaces: the role of context in improving recognition,” in IUI ’06: Proceedings of the 11th international conference on Intelligent user interfaces, (New York, NY, USA), pp. 32–38, ACM, 2006

[86] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in ICML,pp. 282–289, 2001.

[87] S. Kumar and M. Hebert, “Discriminative random fields: A discriminative framework for contextual interaction in classification,” in In ICCV, pp. 1150–1157,2003.

[88] Sminchisescu, Cristian, Kanaujia, Atul and Metaxas, Dimitris., "Conditional models for contextual human motion recognition." Comput. Vis. Image Underst., 2006, Issue 2, Vol. 104, pp. 210-220.

[89] S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian, “Hidden conditional random fields for gesture recognition,” in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1521–1527, 2006

[90] Morency, Louis-Philippe, Quattoni, Ariadna and Darrell, Trevor., "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition." Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 2007, Vol. 0, pp. 1-8.

[91] S. Zhang, M. M. Hossain, M. R. Hassan, J. Bailey, and K. Ramamohanarao, “Feature weighted svms using receiver operating characteristics,” in Siam Data Mining, pp. 497–508, 2009..

[92] R.-H. Liang and M. Ouhyoung, “A real-time continuous gesture recognition system for sign language,” in Automatic Face and Gesture Recognition, 1998. Proceedings.Third IEEE International Conference on, pp. 558–567, 1998.

[93] P. Morguet and M. Lang, “Spotting dynamic hand gestures in video image sequences using hidden markov models,” in International Conference on Image Processing, pp. 193–197, 1998..

[94] R. Cutler and M. Turk, “View-based interpretation of real-time optical flow for gesture recognition,” in FG ’98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition, p. 416, 1998

[95] N. Stefanov, A. Galata, and R. Hubbold, “Real-time hand tracking with variablelength markov models of behaviour,” in CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops, p. 73, 2005

[96] S. Nayak, S. Sarkar, and B. Loeding, “Unsupervised modeling of signs embedded in continuous sentences,” in Computer Vision and Pattern Recognition - Workshops,2005. CVPR Workshops. IEEE Computer Society Conference on, pp. 81–81, 2005.

[97] R. Yang, S. Sarkar, and B. Loeding, “Enhanced level building algorithm for the movement epenthesis problem in sign language recognition,” in Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on, pp. 1–8, 2007.

[98] Gao, Wen, et al., "Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition." IEEE International Conference on Automatic Face and Gesture Recognition, 2004, Vol. 0, p. 553.

[99] J. Deng and H. Tsui, “An hmm-based approach for gesture segmentation and recognition,” in Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 3, pp. 679–682, 2000.

[100] Lee, Hyeon-Kyu and Kim, Jin H., "An HMM-Based Threshold Model Approach for Gesture Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, Vol. 21, pp. 961-973.

[101] R. Yang and S. Sarkar, “Detecting coarticulation in sign language using conditional random fields,” in Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 2, pp. 108–112, 2006.

[102] C. Lee and Y. Xu, “Online, interactive learning of gestures for human/robot interfaces,” in Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, vol. 4, pp. 2982–2987, 1996.

[103] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural Computation, vol. 13, no. 7, pp. 1443–1471, 2001

[104] Canu, S., et al., "SVM and Kernel Methods Matlab Toolbox." SVM and Kernel Methods Matlab Toolbox. 2005.

[105] Freund, Y. and Schapire, R. E., "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting." 1997.

[106] Pan, Sinno Jialin and Yang, Qiang., "A Survey on Transfer Learning." IEEE Transactions on Knowledge and Data Engineering, 2009, Vol. 99.

[107] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in ICML ’07: Proceedings of the 24th international conference on Machine learning, pp. 193–200, 2007

[108] Jiang, Jing and Zhai, Chengxiang., "Instance Weighting for Domain Adaptation in NLP." 2007.

[109] X. Liao, Y. Xue, and L. Carin, “Logistic regression with an auxiliary data source,”in ICML ’05: Proceedings of the 22nd international conference on Machine learning,(New York, NY, USA), pp. 505–512, 2005.

[110] X. Shi, W. Fan, and J. Ren, “Actively transfer domain knowledge,” in ECML PKDD ’08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, pp. 342–357, 2008

[111] Heckman, James J., "Sample Selection Bias as a Specification Error." Econometrica, 1979, Issue 1, Vol. 47, pp. 153-61.

[112] J. Huang, A. Gretton, B. Sch¨olkopf, A. J. Smola, and K. M. Borgwardt, “Correcting sample selection bias by unlabeled data,” in In Neural Information Processing Systems, 2007..

[113] M. Sugiyama, S. Nakajima, H. Kashima, P. von B¨unau, and M. Kawanabe, “Direct importance estimation with model selection and its application to covariate shift adaptation,” in NIPS, 2007.

[114] B. Zadrozny, “Learning and evaluating classifiers under sample selection bias,” in ICML ’04: Proceedings of the twenty-first international conference on Machine learning, p. 114, 2004

[115] S. Bickel, M. Br¨uckner, and T. Scheffer, “Discriminative learning for differing training and test distributions,” in ICML ’07: Proceedings of the 24th international conference on Machine learning, pp. 81–88, 2007..

[116] Argyriou, Andreas., "Multi-task Feature Learning." Advances in Neural Information Processing Systems, 2007, Vol. 19, pp. 243-272.

[117] U. R¨uckert and S. Kramer, “Kernel-based inductive transfer,” in ECML PKDD ’08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, pp. 220–233, 2008

[118] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in EMNLP ’06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128, 2006

[119] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,”in AAAI’08: Proceedings of the 23rd national conference on Artificial intelligence, pp. 677–682, 2008..

[120] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, “Integrating structured biological data by Kernel Maximum Mean Discrepancy,” Bioinformatics, vol. 22, no. 14, pp. e49–57, 2006

[121] A. Torralba, K. Murphy, and W. Freeman, “Sharing features: efficient boosting procedures for multiclass object detection,” in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, 2004.

[122] Thrun, S., Learning to learn. [ed.] Sebastian Thrun and Lorien Pratt. s.l. : Kluwer Academic Publishers, 1998.

[123] T. Evgeniou and M. Pontil, “Regularized multi–task learning,” in KDD ’04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, (New York, NY, USA), pp. 109–117, ACM, 2004

[124] N. D. Lawrence and J. C. Platt, “Learning to learn with the informative vector machine,” in ICML ’04: Proceedings of the twenty-first international conference on Machine learning, p. 65, 2004.

[125] L. Mihalkova, T. Huynh, and R. J. Mooney, “Mapping and revising markov logic networks for transfer learning,” in AAAI’07: Proceedings of the 22nd national conference on Artificial intelligence, pp. 608–614, AAAI Press, 2007.

[126] Richardson, Matthew and Domingos, Pedro., "Markov logic networks." Mach. Learn., 2006, Issue 1-2, Vol. 62, pp. 107-136.

[127] E. H. Spriggs, F. D. la Torre Frade, and M. Hebert, “Temporal segmentation and activity classification from first-person sensing,” in IEEE Workshop on Egocentric Vision, CVPR 2009, 2009.

[128] Yin, Jie, Yang, Qiang and Pan, Jeffrey Junfeng., "Sensor-Based Abnormal Human-Activity Detection." IEEE Trans. on Knowl. and Data Eng., 2008, Issue 8, Vol. 20, pp. 1082-1090.

[129] Wu, J.X., et al., "A Scalable Approach to Activity Recognition based on Object Use." 2007. pp. 1-8.

[130] Valiant, L. G., "A theory of the learnable." 1984. pp. 436-445.

[131] Subramanya, A., et al., "Hierarchical Models for Activity Recognition." 2006. pp. 233-237.

[132] Stikic, M., et al., "ADL recognition based on the combination of RFID and accelerometer sensing." 2008. pp. 258-263.

[133] Raina, Rajat, Ng, Andrew Y. and Koller, Daphne., "Constructing informative priors using transfer learning." 2006. pp. 713-720.

[134] Pentland, A., "Looking at people: sensing for ubiquitous and wearable computing." Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000, Issue 1, Vol. 22, pp. 107-119.

[135] Logan, Beth, et al., "A Long-Term Evaluation of Sensing Modalities for Activity Recognition." 2007. pp. 483-500.

[136] la, F. De, et al., "Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database." 2009.

[137] Krishnan, Narayanan C., et al., "Recognition of hand movements using wearable accelerometers." Journal of Ambient Intelligence and Smart Environments, 2009, Issue 2, Vol. 1, pp. 143-155.

[138] Krishnan, Narayanan C., Lade, Prasanth and Panchanathan, Sethuraman., "Activity gesture spotting using a threshold model based on adaptive boosting." TRCUbiC20101, 2010, Vol. 1.

[139] Kasteren, T.L.M. van and Krose, B.J.A., "A sensing and annotation system for recording datasets in multiple homes." 2009.

[140] Intille, S. S., et al., "A living laboratory for the design and evaluation of ubiquitous computing technologies." 2005. pp. 1941-1944.

[141] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in sOc- EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence, pp. 159–163, 2005.

[142] Hu{\`y}nh, T{\^a}m and Schiele, Bernt., "Unsupervised Discovery of Structure in Activity Data Using Multiple Eigenspaces." Location- and Context-Awareness, 2006, pp. 151-167.

[143] Dai, Wenyuan, et al., "Translated Learning: Transfer Learning across Different Feature Spaces." 2008. pp. 353-360.

[144] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas,"Collecting and disseminating smart home sensor data in the casas project," in Proc. of CHI09 Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, 2009.

[145] Choudhury, Tanzeem, et al., "The Mobile Sensing Platform: An Embedded Activity Recognition System." IEEE Pervasive Computing, 2008, Vol. 7, pp. 32-41.

[146] Aipperspach, Ryan, Cohen, Elliot and Canny, John., "Modeling Human Behavior from Simple Sensors in the Home." International Conference on Pervasive Computing, 2006, pp. 337-348.

[147] G. D. Abowd, C. G. Atkeson, A. F. Bobick, I. A. Essa, B. MacIntyre, E. D. Mynatt, and T. E. Starner, "Living laboratories: the future computing environments group at the georgia institute of technology," in CHI '00: CHI '00 extended abstracts on Human factors in computing systems, pp. 215–216, 2000.

[148] F. Doctor, H. Hagras, and V. Callaghan. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 35(1):55-65, 2005

[149] H. Hagras, F. Doctor, A. Lopez, and V. Callaghan. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. IEEE Transactions on Fuzzy Systems, 15(1):41-55, 2007

[150] S. Intille and K. Larson. Designing and evaluating supportive technology for homes. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003

[151] M. Mozer. The Neural Network House: An environment that adapts to its residents. Proceedings of the National Conference on Artificial Intelligence, pages 110-114, 1998.

[152] G.M. Youngblood, D.J. Cook, L.B. Holder, and E. Heierman. Automation intelligence for the smart environment. Proceedings of the International Joint Conference on Artificial Intelligence, 2005.

[153] Katz S., Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living, J Am Geriatr Soc. 1983 Dec;31(12):721-7.

[154] Marculescu, D., Marculescu, R., Zamora, N.H., Stanley-Marbell, P., Khosla, P.K., Park, S., Jayaraman, S., Jung, S., Lauterbach, C., Weber, W., Kirstein, T., Cottet, D., Grzyb, J., Tröster, G., Jones, M., Martin, T., Nakad, Z., "Electronic Textiles: A Platform for Pervasive Computing," Proceedings of the IEEE, vol. 91, No 12, December 2003, pp. 1995-2018

[155] http://www.sparkfun.com/commerce/categories.php

[156] http://www.freescale.com/webapp/sps/site/taxonomy.jsp?code=ACCLOWG&tid=vanxyz

[157] http://fiji.eecs.harvard.edu/Mercury

[158] http://wockets.wikispaces.com/Wockets+at+Stanford

[159] S. Krishna, V. Balasubramanian, N. C. Krishnan, J. Colin, T. Hedgpeth, and S. Panchanathan, "A Wearable Wireless RFID Systems for Accessible Shopping Environments," Third International Conference on Body Area Networks (BodyNets08), Tempe, AZ, April 2008.

[160] A. Rea, K. Fishkin and M. Philipose. Hands-On RFID: Wireless Wearables for Detecting Use of Objects. Proceedings of ISWC 2005, Osaka, October 2005

[161] J. R. Smith, K. Fishkin, B. Jiang, A. Mamishev, M. Philipose, A. Rea, S. Roy, K. Sundara-Rajan. RFID-Based Techniques for Human Activity Recognition. Communications of the ACM, v48, no. 9, Sep 2005.

[162] E. Munguia Tapia, S.S. Intille, L. Lopez, and K. Larson, "The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection" in Proceedings of PERVASIVE 2006. B. Heidelberg, Ed. Dublin, Ireland: Springer-Verlag, 2006.

[163] Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Haehnel. Inferring Activities from Interactions with Objects. IEEE Pervasive Computing, v3 no. 4, pp. 50-57, Oct 2004.

[164] G.Singla, D. Cook, and M. Schmitter-Edgecombe. Incorporating temporal reasoning into activity recognition for smart home residents. Proceedings of the AAAI Workshop on Spatial and Temporal Reasoning, pages 53-61, 2008.

[165] Kim, H. H., Ha, K. N., Lee, S., and Lee, K. C. 2009. Resident location-recognition algorithm using a Bayesian classifier in the PIR sensor-based indoor location-aware system. Trans. Sys. Man Cyber Part C 39, 2 (Mar. 2009), 240-245.

[166] Orr and Abowd. The smart floor: A mechanism for natural user identification and tracking. CHI'00 extended abstracts on Human factors in computing systems (2000) pp. 276

[167] Richardson et al. Z-Tiles: building blocks for modular, pressure-sensing floor spaces. CHI '04: CHI '04 extended abstracts on Human factors in computing systems (2004)

[168] Liao, L., Patterson, D. J., Fox, D., and Kautz, H. 2007. Learning and inferring transportation routines. Artif. Intell. 171, 5-6 (Apr. 2007), 311-331.

[169] S. S. Intille, K. Larson, E. Munguia Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," in Proceedings of PERVASIVE 2006 , vol. LNCS 3968, K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Berlin Heidelberg: Springer-Verlag, 2006, pp. 349-365.

[170] S. S. Intille, K. Larson, J. Beaudin, E. Munguia Tapia, P. Kaushik, J. Nawyn, and T.J. McLeish, "The PlaceLab: a live-in laboratory for pervasive computing research (Video)," in Proceedings of Pervasive 2005 Video Program. May, 2005

[171] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas. Collecting and disseminating smart home sensor data in the CASAS project. Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, 2009

[172] F.D.laTorre, J.Hodgins, J.Montano, S.Valcarcel,and J.Macey.,"Guide to the carnegie mellon university multimodal activity (cmu-mmac) database,"in Tech report CMU-RI-TR-08-22 ,RoboticsInstitute, CarnegieMellonUniversity, 2009

[173] http://www.mathworks.com/

[174] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, New York: John Wiley & Sons, 2001, pp. xx + 654, ISBN: 0-471-05669-3

[175] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999

[176] P. Komarek and A. Moore, Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs. 2003.

[177] Y. A. Ivanov, A. F. Bobick, Recognition of Visual Activities and Interactions by Stochastic Parsing, Transactions on Pattern Analysis and Machine Intelligence 22(8), 852-872 (2000)

[178] Mike Perkowitz, Matthai Philipose, Donald J. Patterson, Kenneth P. Fishkin.Mining Models of Human Activities from the Web. Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004), May 2004, pp. 573-58

[179] Jeffrey Boyd, Hari Sundaram, Aviral Shrivastava: Power-accuracy tradeoffs in human activity transition detection. DATE 2010: 1524-1529

[180] Viterbi AJ (April 1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". IEEE Transactions on Information Theory 13 (2): 260–269

[181] Fei Sha and Lawrence K. Saul.Large margin Gaussian mixture modeling for phonetic classification and recognition Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), pages 265-268

[182] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern

Recognition. CVPR 2001, vol. 1. Los Alamitos, CA, USA: IEEE Comput. Soc, April 2001, pp. I-511-I-518

[183] del Alamo, C. M., Caminero Gil, F. J., dela Torre Munilla, C., and Hernandez Gomez, L. 1996. Discriminative training of GMM for speaker identification. In Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE international Conference - Volume 01 (May 07 - 10, 1996). ICASSP. IEEE Computer Society, Washington, DC, 89-9

[184] C.-S. Liu, C.-H. Lee, W. Chou, and A. E. Rosenberg , A Study on Minimum Error Discriminative Training for Speaker Recognition,", Jour. Acoust. Soc. Am., Dec. 1994

[185] Paul Horton & Kenta Nakai, "A Probablistic Classification System for Predicting the Cellular Localization Sites of Proteins", Intelligent Systems in Molecular Biology, 109-115. St. Louis, USA 1996

[186] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

[187] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," International Journal of Computer Vision, vol. 42, no. 3, pp. 145-175, May 2001

[188] A. Torralba, "Contextual priming for object detection," International Journal of Computer Vision, vol. 53, no. 2, pp. 169-191, July 2003

[189] Lawrence R. Rabiner (February 1989). "A tutorial on Hidden Markov Models and selected applications in speech recognition". Proceedings of the IEEE 77 (2): 257–286.

[190] Jiang, J. and Zhai, C.X.: Instance weighting for domain adaptation in NLP. Annual Meeting-Association For Computational Linguistics 1, vol. 45, pp. 264. (2007)

[191] Eaton, E. and desJardins, M.: Set-Based Boosting for Instance-Level Transfer. IEEE International Conference on Data Mining Workshops, pp. 422–428. IEEE (2009)

[192] Freund, Y. and Schapire, R.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Computational learning theory, pp. 23–37. Springer (1995)

[193] Y. Freund and R. Schapire, "A short introduction to boosting," Japonese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, 1999

[194] Sun, Y. and Kamel, M.S. and Wong, A.K.C. and Wang, Y.: Cost-sensitive Boosting for Classification of Imbalanced Data. Pattern Recognition 12, vol. 40, pp. 3358–3378. Elsevier (2007)

[195] R. E. Schapire and Y. Singer, "Improved boosting using confidence-rated predictions," Machine Learning, vol. 37, no. 3, pp. 297-336, 1999

[196] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," The Annals of Statistics, vol. 26, no. 5, pp. 1651-1686, 1998.

[197] Kullback, S.; Leibler, R.A. (1951). "On Information and Sufficiency". Annals of Mathematical Statistics 22 (1): 79–86

[198] E. Baum and D. Haussler, "What size net gives valid generalization?" Neural Computation, vol. 1, pp. 151-160, 1989.

[199] Caruana, Rich, "Multitask Learning." Machine Learning, Vol. 28, pp. 41-75

[200] http://people.csail.mit.edu/jrennie/20Newsgroups/

[201]P.Beaudoin, S.Coros, M.vandePanne, and P.Poulin, Motion-motifgraphs .ACMSIGGRAPH, Eurographics Symposium on Computer Animation, 2008.

[202] Q.Yang, Kangheng,Wu, and Y.Jiang, Learning action models from plan examples using weighted MAX-SAT .Artificial Intelligence,171,2-3,107-143,Feb2007

[203] M.E.Pollack, C.E.McCarthy, S.Ramakrishnan, I.Tsamardinos, L.Brown, S.Carrion, D.Colbry, C.OroszandB.Peintner, Autominder: APlanning, Monitoring, and Reminding Assistive Agent .7th International Conferenceon Intelligent Autonomous Systems,2002

[204] C.L.Isbell, O.Omojokun ,and J.S.Pierce, From devices to tasks :automatic task prediction for personalized appliance control.Personal Ubiquitous Computing8,3-4,146-153,Jul 2004.

[205] Raftery, A.E., 1985. A model for high-order Markov chains. J. Roy. Statist. Soc. B 47, 528-539.

[206] Berchtold, A., Estimation in the mixture transition distribution model. J. Time Ser. Anal. 22, 379-397.

[207] Fawcett, T. An introduction to ROC analysis. Pattern Recognition Letters, 27, 861–874 2006.

[208] Prasanth Lade, N C Krishnan, S Panchanathan, ' Task prediction in cooking activities using hierarchical Markov chain and object based task grouping', IEEE International Symposium on Multimedia workshop on multimedia for cooking and eating activities, Taiwan, 2010

[209] Ashok Venkatesan, N C Krishnan, S Panchanathan, Cost sensitive boosting for concept drift, European Conference on Machine learning workshop on handling concept drift in Adaptive Information systems, Barcelona, Spain, 2010.

[210] K. Yin and D. K. Pai, ``FootSee: an Interactive Animation System," in Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation, San Diego, July 26-27, 2003

[211] Bulling, A., Ward, J. A., Gellersen, H., and Tröster, G. 2009. Eye movement analysis for activity recognition. InProceedings of the 11th international Conference on Ubiquitous Computing (Orlando, Florida, USA, September 30 - October 03, 2009). Ubicomp '09. ACM, New York, NY, 41-50

[212] Stager, M., Lukowicz, P., and Troster, G. 2004. Implementation and Evaluation of a Low-Power Sound-Based User Activity Recognition System. In Proceedings of the Eighth international Symposium on Wearable Computers (October 31 - November 03, 2004). ISWC. IEEE Computer Society, Washington, DC, 138-141.

[213] Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., Klasnja, P. "., Koscher, K., LaMarca, A., Landay, J. A., LeGrand, L., Lester, J., Rahimi, A., Rea, A., and Wyatt, D. 2008. The Mobile Sensing Platform: An Embedded Activity Recognition System. IEEE Pervasive Computing 7, 2 (Apr. 2008), 32-41

[214] K. Van Laerhoven and H W. Gellersen Spine versus Porcupine: A Study in Distributed Wearable Activity Recognition, , 8th International Symposium on Wearable Computers (ISWC 2004), p.142-149

[215] N. Kern, B. Schiele,and A. Schmidt,"Multi-sensor activity context detection for wearable computing," Ambient Intelligence, pp.220–232, 2003.

[216] Daniel Olguín Olguín, and Alex (Sandy) Pentland. Human Activity Recognition: Accuracy across Common Locations for Wearable Sensors. IEEE 10th International Symposium on Wearable Computing. Montreaux, Switzerland. October 11-14, 2006