

Behaviour Profiling using Wearable Sensors for Pervasive Healthcare

Syed Muhammad Raza Ali
The Hamlyn Centre
Imperial College of Science, Technology and Medicine

2012

This thesis is submitted to the Imperial College London in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Except for where indicated, it presents entirely my own work and describes the results of my own research.

Abstract

In recent years, sensor technology has advanced in terms of hardware sophistication and miniaturisation. This has led to the incorporation of unobtrusive, low-power sensors into networks centred on human participants, called Body Sensor Networks. Amongst the most important applications of these networks is their use in healthcare and healthy living. The technology has the possibility of decreasing burden on the healthcare systems by providing care at home, enabling early detection of symptoms, monitoring recovery remotely, and avoiding serious chronic illnesses by promoting healthy living through objective feedback. In this thesis, machine learning and data mining techniques are developed to estimate medically relevant parameters from a participant's activity and behaviour parameters, derived from simple, body-worn sensors.

The first abstraction from raw sensor data is the recognition and analysis of activity. Machine learning analysis is applied to a study of activity profiling to detect impaired limb and torso mobility. One of the advances in this thesis to activity recognition research is in the application of machine learning to the analysis of 'transitional activities': transient activity that occurs as people change their activity. A framework is proposed for the detection and analysis of transitional activities. To demonstrate the utility of transition analysis, we apply the algorithms to a study of participants undergoing and recovering from surgery. We demonstrate that it is possible to see meaningful changes in the transitional activity as the participants recover.

Assuming long-term monitoring, we expect a large historical database of activity to quickly accumulate. We develop algorithms to mine temporal associations to activity patterns. This gives an outline of the user's routine. Methods for visual and quantitative analysis of routine using this summary data structure are proposed and validated.

The activity and routine mining methodologies developed for specialised sensors are adapted to a smartphone application, enabling large-scale use. Validation of the algorithms is performed using datasets collected in laboratory settings, and free living scenarios.

Finally, future research directions and potential improvements to the techniques developed in this thesis are outlined.

Acknowledgements

I'd like to thank my advisor, Professor Guang-Zhong Yang for providing me the opportunity, guidance, and often impetus to undertake, and persist with this dissertation. The breadth of his research interests and the capacity to excel in each direction is inspirational, and will serve as my example to emulate.

I am grateful to Dr. Louis Atallah for his patient tutoring and collaboration. For most of the research included in this dissertation, I was privileged to collaborate with him: from the design of the analysis algorithms to designing and conducting experiments. I am especially grateful for his help in writing, and his feedback at each stage of my PhD.

The pervasive sensing group I was fortunate to be part of provided friendly assistance and advice whenever I needed. I hope to carry these associations on at Imperial and away from it.

I want to thank my family. My sisters, Samreen Sarfaraz and Farheen Shiraz made me want to be a doctor too, although I still can't treat people. My parents Ziarat Ali and Shamim Fatima let me pick my own path, and supported me morally and materially. And my wife Madiha, has brought me the luck promised in Pakistani folk stories to good wives.

Finally, I am grateful to friends, present and departed. Bismillah.

To my family

Contents

CHAPTER 1.....	18
1.1 INTRODUCTION	18
1.2 CHALLENGES IN PERVASIVE HEALTHCARE SYSTEMS	20
1.2.1 <i>Clinical Requirements</i>	20
1.2.2 <i>Technology Opportunities</i>	22
1.2.3 <i>Data Analysis Requirements</i>	23
1.3 PROBLEM STATEMENT.....	24
1.4 ORGANISATION OF THE THESIS.....	25
1.5 ORIGINAL TECHNICAL CONTRIBUTION OF THE THESIS.....	26
CHAPTER 2.....	28
BEHAVIOUR PROFILING WITH PERVASIVE SENSING	28
2.1 INTRODUCTION	28
2.2 SENSING MODALITIES	31
2.3 SENSOR DATA REPRESENTATION	34
2.3.1 <i>Synopsis Structures</i>	35
2.3.2 <i>Feature Extraction</i>	38
2.4 BEHAVIOUR PROFILING.....	41
2.4.1 <i>Activity Analysis</i>	42
2.4.2 <i>Transitional Activities</i>	46
2.4.3 <i>Modelling Behaviour</i>	48
2.4.4 <i>Detecting Anomalous Behaviour</i>	51
2.5 CONCLUSION	54
CHAPTER 3.....	56
DETECTION AND CLASSIFICATION OF ACTIVITIES OF DAILY LIVING.....	56
3.1 INTRODUCTION	56
3.2 ACTIVITY DETECTION AND CLASSIFICATION	60
3.2.1 <i>Sensing Hardware</i>	60
3.2.2 <i>Feature Extraction</i>	61

3.2.3 Dimensionality Reduction.....	66
3.2.4 Generating Class Labels.....	67
3.3 LABORATORY EXPERIMENT SETUP FOR ACTIVITY CLASSIFICATION	70
3.4 RESULTS	71
3.4.1 Classification of Impairment.....	73
3.4.2 Impact of Dimensionality	74
3.5 CONCLUSIONS	75
CHAPTER 4.....	77
ANALYSING TRANSITIONAL ACTIVITIES.....	77
4.1 INTRODUCTION	77
4.2 ANALYSIS FRAMEWORK FOR TRANSITIONAL ACTIVITY RECOGNITION	79
4.3 MANIFOLD MODELLING OF TRANSITIONAL ACTIVITY	80
4.3.1 Representing Activity with Manifolds.....	81
4.4 DETECTING TRANSITIONAL ACTIVITY	83
4.5 USING MANIFOLDS TO CAPTURE INTRINSIC DIMENSIONALITY	86
4.5.1 Isomap.....	88
4.5.2 Estimating Intrinsic Dimensionality.....	89
4.5.3 Representing Learned Manifolds as Models	90
4.6 CATEGORISATION AND ANALYSIS	90
4.7 EXPERIMENTAL SETUP	91
4.8 RESULTS	93
4.8.1 Feature Extraction	94
4.8.2 Transition Detection	95
4.8.3 Dimensionality Reduction.....	97
4.8.4 Transition Categorisation	99
4.8.5 Transition Analysis.....	99
4.9 CONCLUSIONS	104
CHAPTER 5.....	106
DISCOVERING ROUTINES FROM ACTIVITIES	106
5.1 INTRODUCTION	106
5.2 ACTIVITY CLASSIFICATION.....	107
5.3 MINING ROUTINE ACTIVITIES.....	109
5.3.1 Background	110
5.3.2 Routine Tree	113

5.3.3 Transaction Representation of Sensor Data.....	114
5.3.4 Tree Generation.....	115
5.3.5 Tree Pruning	116
5.3.6 Aggregating Routine Trees.....	117
5.4 RESULTS	118
5.5 CONCLUSIONS	122
CHAPTER 6.....	124
MINING ACTIVITY AND ROUTINE USING SMARTPHONES.....	124
6.1 INTRODUCTION	124
6.2 ACTIVEMILES	126
6.3 OVERVIEW OF METHODOLOGY	129
6.4 FEATURE EXTRACTION	130
6.5 DIMENSIONALITY REDUCTION USING MANIFOLD EMBEDDING	131
6.6 ACTIVITY ANALYSIS	132
6.6.1 K-means Clustering.....	133
6.6.2 Gaussian Mixture Model based Clustering.....	133
6.7 INCORPORATING LOCATION INTO ROUTINE TREE	134
6.7.1 Location as Feature	134
6.7.2 Location as Dimension.....	135
6.8 RESULTS	137
6.8.1 Analysis of Study I - Labelled Activity Dataset.....	138
6.8.2 Analysis of Study II – Routine Mining of Historical Activity Data	144
6.9 CONCLUSIONS	150
CHAPTER 7.....	152
LONG-TERM PROFILING WITH PERVASIVE SENSING.....	152
7.1 INTRODUCTION	152
7.2 METHODOLOGY	155
7.3 GENERATING A DISTANCE MATRIX FROM ROUTINE TREES	155
7.3.1 Computing Node Distance Matrices.....	156
7.3.2 Generating Composite Distance Matrix	158
7.4 ANALYSIS OF DISTANCE MATRICES.....	160
7.5 RESULTS	160
7.5.1 ActiveMiles Dataset.....	161
7.5.2 SAPHE Dataset.....	168

7.6 CONCLUSIONS	171
CHAPTER 8.....	173
CONCLUSIONS AND FUTURE WORK.....	173
8.1 CONCLUSIONS	173
8.2 SUMMARY OF METHODOLOGY.....	174
8.3 FUTURE WORK	177
8.4 CONCLUSIONS	180
8.5 ACKNOWLEDGEMENT	181
APPENDIX A.....	183
MANIFOLD LEARNING	183
A.1 INTRODUCTION	183
A.2 MOTIVATION	183
A.3 DIMENSIONALITY REDUCTION	185
A.4 WHAT ARE MANIFOLDS?.....	188
A.5 DIMENSIONALITY REDUCTION WITH MANIFOLD EMBEDDING	189
<i>A.5.1 Quality of Embedding.....</i>	<i>190</i>
<i>A.5.2 Complexity.....</i>	<i>191</i>
<i>A.5.3 Sensitivity to Training Conditions.....</i>	<i>192</i>
A.6 CONCLUSION.....	192
APPENDIX B.....	194
WAVELET ANALYSIS	194
B.1 INTRODUCTION	194
B.2 MOTIVATION	194
B.3 WAVELETS	196
B.4 WAVELET TRANSFORM.....	197
B.5 CONCLUSION.....	199
REFERENCES.....	200

List of Figures

Figure 1.1 The main differences between episodic and continuous monitoring and how the traditional episodic way of patient information management can be avoided by using pervasive sensing.....	19
Figure 2.1 Data flow for behaviour profiling in a pervasive sensing environment. Raw sensor data is progressively refined and abstracted as it passes through the machine learning framework.	31
Figure 2.2 Sensors for behaviour profiling – wearable e-AR sensor (a), camera based ambient sensor (b) and Passive-Infra Red (PIR) sensor (c).	33
Figure 2.3 Accelerometer Sensor Data (a) represented as data histogram (b), wavelet coefficients (c), and SAX Representation (d). Histograms represent data in terms of counts associated with signal values. Wavelet and SAX representations can significantly reduce numerosity by representing information with fewer data points, as shown in the figure.	37
Figure 2.4 Activities of daily living can be detected using wearable sensors such as the ear-worn activity recognition sensor. The change in sensor data can be seen for each activity performed by the participant.	43
Figure 2.5 HMMs used in activity recognition: (a) illustrates a simple activity, such as preparing breakfast, where the circles represent the constituent tasks involved in the overall process and shaded boxes represent observations made by sensors. For more complex activities, a coupled HMM (b) allows a collection of HMMs to represent multiple interleaving activities.	45
Figure 2.6 Participant performing transitional activities demonstrating the gradual postural change between different stable states.	47
Figure 2.7 Intuitive activity grids used to visualise a user’s overall behaviour pattern in a home environment. These two simulated activity grids illustrate contrasting behavioural patterns: (a) the user spends most time in the living room and (b) more time is spent in the bedroom and toilet indicating anomalous behaviour.	52
Figure 3.1 Typical machine learning pipeline in machine learning applications. ‘Data’ refers to sensor data, or features extracted from sensor data. Typically preprocessing is required before the data can be further analysed. Feature analysis can consist of selecting optimal feature sets or reducing the dimensionality after feature extraction. Finally, machine learning tools can be applied for particular applications.....	59

Figure 3.2 Participant wearing e-AR sensor (circled in red and shown separately), with raw sensor data showing the participant sitting, transitioning, walking to TV, transitioning to sitting again. Regions for sitting, transitioning and walking are coloured on the data plot as blue, brown and red respectively.	60
Figure 3.3 Waveform statistics extracted from accelerometer sensor data shown in Figure 3.2. Energy and variance are strongly indicative of activity intensity.	63
Figure 3.4 Features extracted from wavelet analysis. X and Y-axes plot window numbers and feature values respectively. At each level the mean and standard deviation from the wavelet coefficients is extracted as the feature.	64
Figure 3.5 Pairs of sensor channels and their associated covariances, channels 1 and 2 (a), 1 and 3 (b) and 2 and 3 (c).	65
Figure 3.6 Examples of classification algorithms. The decision tree (a) partitions the dataset at each node based on attribute values. The Naive Bayesian (b) model shows the conditional independence assumption graphically, where the arrows indicate dependence. The Multi-Layer Perceptron (c) shown here has a single hidden layer has an activation layer corresponding to the attributes, a single hidden layer with two neurons, and an output layer giving the classification.	69
Figure 3.7 A participant performing the washing up activity while wearing the braces system designed to impair lower limb and truncal movement (a), and the corresponding sensor data (b).	70
Figure 3.8 Impairment detection data in first three dimensions of reduced feature space for all participants together. Red indicates simulated impairment, while blue indicates normal data... 72	72
Figure 3.9 Impairment detection data in first three dimensions of reduced feature space for a single participant. Red indicates simulated impairment, while blue indicates normal data. Impaired and normal data points cluster more distinctly compared to the full feature space shown in Figure 3.9.....	73
Figure 3.10 Classification accuracy for impairment detection using PCA reduced features (a) and Simba selected features (b). Using Simba features leads to an average 10% increase in accuracy.	74
Figure 3.11 Classification accuracy of C4.5 classifier with increasing number of features selected using the Simba algorithm. In general there is a small gain in classification accuracy with increasing number of features.	75
Figure 4.1 BSN sensor data from a participant performing the Stand-to-Sit transition. Data morphs from standing to sitting in a distinctive pattern.....	78

Figure 4.2 Overview of Transition detection and analysis methodology, with unsupervised learning based preprocessing before classification and labelling of transitions is performed. ...	80
Figure 4.3 Participant approaches bed and lies down: sensor signal (a) and in a manifold visualisation (b). The two activities cluster distinctly in manifold space.	82
Figure 4.4 Two-dimensional embedding of Sitting to Walking sensor data (top) with first two dimensions of PCA and Isomap (bottom). Isomap preserves transitional activity relationship in embedded space, which is lost in PCA reduction.	87
Figure 4.5 Accelerometer data from participants stepping up on, and down from a 20cm high bench. The signals show a difference between healthy participants and participants moving with difficulty after 3 weeks of surgery.	93
Figure 4.6 Levels of Accuracy, Precision and Recall against recursion level in Transition Detection Algorithm for Study I. Both statistics are influenced by level of recursion.	95
Figure 4.7 Transitions detected using recursive partitioning method at different levels of recursion. The Sit-to-Stand transition is under-segmented when the recursion depth is 2, and over-segmented at the next level.....	96
Figure 4.8 Comparison of the performance of Isomap against PCA using residual variance. Isomap achieves low residual variance with fewer dimensions compared to PCA.	98
Figure 4.9 Residual Variance of classifiers learning embedding for different activities. Linear algorithms are used based on this analysis as they are low in computational cost and achieve low residual variance.	99
Figure 4.10 Manifold embedding of three participants performing the step transition naturally (a) and with impairment (b). The clearly separated clusters correspond to each participant. ...	100
Figure 4.11 Stepping transitions for patients at different weeks of recovery embedded in separate spaces. Figures (a)-(e) show Participants (1-5). Early recovery (1-6 weeks) and Late Recovery (12-24) stages are embedded closer together.....	102
Figure 4.12 Transitions from five patients recovering from surgery. Figures show embedding of all patients together, with Step-up transitions in (a), and Stand-to-Sit transitions (chair) in (b). Clusters correspond to two stages of recovery observed during data collection.....	103
Figure 5.1 Segmentation of raw sensor data into Activities using the Activity Classifier. Each window corresponds to 200 samples. A coarse segmentation can be achieved by filtering the activity classification, removing noise. The classifications can be subsequently mined for routine extraction.	109
Figure 5.2 Example of an activity database where activity levels are aggregated every minute.	111

Figure 5.3 FP-Tree after scanning the first three transactions respectively (a), (b) and (c). The node is labelled as Item:Count. It is possible to mine frequent itemsets in the database using this data structure without rescanning the database.	112
Figure 5.4 Routine Tree with support shown between 12:00 and 23:59. Each node in the tree is associated with a table of activity patterns frequent in that time period.	114
Figure 5.5 Pruning a Routine Tree –an example of a merge operation. Nodes meeting user-specified criteria can be merged to display a more compact tree.	116
Figure 5.6 Leaf view of routines trees for simulated dataset showing the behaviour of a participant. Five trees are shown at the leaf level, with the highest level of detail.	120
Figure 5.7 Tree view for a participant wearing the sensor during late evening hours of the day. A single tree is shown, in increasing degree of detail. The numbers on the left correspond to levels of the tree. Higher levels show broad trends, while lower levels of the tree show more detail.	121
Figure 5.8 Tree view for a participant wearing the sensor during a wakeful night.	121
Figure 5.9 Tree view of participant on two days. (a) shows a day with high daytime activity, particularly between close to 14:00, while (b) shows a day where the user had low activity during the afternoon.	122
Figure 6.1 Activity feedback shown to users of the Active Miles application: daily, weekly and monthly activity levels are compared against other users of the system, and a designated competitor termed Nemesis. Day plots are over 24 hours, with each bar corresponding to an hour. Week and Month plots are over 7 and 30 days respective, with each bar corresponding to a day. The goal of the system is to encourage users to increase the points they win by being more active.	127
Figure 6.2 Data collected from the Android™ based Active Miles application: GPS based location (a), Cell-tower triangulation based location (b), Accelerometer sensor (c) and Orientation sensor (d) . A Samsung Galaxy Mini™ phone was used for this circuit.	128
Figure 6.3 Features extracted from accelerometer and orientation sensors, plotted against window index. Each window corresponds to 2 seconds of sensor data. User is initially walking, and then proceeds to sit down.	131
Figure 6.4 First two dimensions of Isomap reduction showing participant transition from Walking to Sitting activities.	132
Figure 6.5 Accelerometer data showing the walking activity when the phone is in the jacket (a) and trouser front-pocket (b). The signal features can be significantly different for the same activity depending on the position of the data.	138

Figure 6.6 First three dimensions of Isomap embedding of activities performed in Study I. Data corresponding to each activity clusters together, but may belong in multiple clusters depending on phone location, as in the walking activity.	139
Figure 6.7 Using Residual Variance to estimate the intrinsic dimensionality of activity data. The arrow shows the ‘elbow’ of the residual variance curve. This gives an estimate of the dataset’s true dimensionality.	139
Figure 6.8 Accuracy of Classifiers for Activity Detection versus Manifold Dimensionality. There is diminishing impact of dimensionality on classification accuracy after three dimensions.	140
Figure 6.9 Isomap embedding of accelerometer data for the running activity. Clusters correspond to running speed.	141
Figure 6.10 Running speed (in km/h) as labelled and as estimated using a GRNN regression classifier.	142
Figure 6.11 Manually placed activity labels versus cluster labels generated by K-Means and GMM Algorithms for Study I. GMM outperforms the K-Means when compared against activity labels.	143
Figure 6.12 Membership sizes of GMM clustering for Study II (a) and average activity intensity for each cluster (b).	145
Figure 6.13 Membership sizes of GMM clustering for Study II (a) and average activity intensity for each cluster (b).	145
Figure 6.14 Routine Tree of a Participant showing office-activity, including a weekly meeting	146
Figure 6.15 Routine Tree of a Participant showing holiday activity, with low activity in the daytime and high-intensity activity patterns in the evening.	146
Figure 6.17 Semantic labels attached to significant labels found by the algorithm can be placed by users.	148
Figure 6.18 Routine tree incorporating location into the visualisation. Location can be used to contextualise the activity patterns.	148
Figure 6.19 Routine tree for the ‘Home’ location, showing activity in the morning before the participant leaves for work, and in the evening when he returns.	149
Figure 7.1 A comparison of two days of sensor data using with the real signal on the left and alignment using Dynamic Time Warping (DTW) on the right. The missing data in day 1 results in an incorrect alignment.	153
Figure 7.2 Methodology for generating a composite distance matrix from a set of routine trees.	156

Figure 7.3 Three Node Distance Matrices that can be generated from three routine trees, with nodes colour coded according to their associated distance matrix, where $\emptyset$ denotes a null entry.	157
Figure 7.4 Five Node Distance Matrices (a) , their corresponding similarity matrices (b) and the Composite Distance Matrix (c) containing information from all the NDM. Blue corresponds to low distance, yellow to medium and red to high.	159
Figure 7.5 Activity patterns corresponding to the centroid node of the NDM. This corresponds to a ‘typical’ routine of the user.	161
Figure 7.6 Isomap Embedding of Composite Distance Matrix, with each day coloured according to the month.....	162
Figure 7.7 Residual Variance of the Isomap reduction of the CDM. The elbow of the curve occurs at the seventh dimension.....	163
Figure 7.9 Gap Statistic analysis on the CDM of the Active Miles dataset. Based on the analysis the number of clusters is set at 3.	164
Figure 7.10 GMM clustering of CDM shown in manifold space (a), and average activity level of each cluster.	165
Figure 7.11 Representative Routine Trees corresponding to the three clusters respectively..	166
Figure 7.12 Membership of days of the week in the three clusters. Cluster 3 has low weekend representation, in contrast to Clusters 1 and 2.	167
Figure 7.13 Gap analysis of SAPHE dataset. Plot shows number of times cluster size is selected by the Gap criteria, with six cluster sizes selected most often.	168
Figure 7.14 The participant’s routine changed during the month of July (red) compared to the routine in other months (blue).....	169
Figure 7.15 A manifold embedding of the CDM with Saturday routines shown in red and remaining weekday routines in blue. Although there are too few points to make a strong claim, a weak grouping of red points could indicate change in behaviour associated with the weekend.	170
Figure A.1 The number of samples required to achieve a fixed Vapnik–Chervonenkis (VC) generalisation bound, with increasing dimensionality. Higher dimensionality results in significantly increased training data requirements.....	184
Figure A.2 Helix dataset generated through equations A.2, with a set of points overlaid with increasing t . We will seek to preserve relationships between each pair of points when representing the data in fewer dimensions.	185

Figure A.3 Helix dataset generated through equations A.2, embedded in two dimensions using Multi-dimensional Scaling (MDS). Linear dimensionality reduction fails to preserve the order of points.....	187
Figure B.1 FFT analysis of signal from accelerometer Up-Down channel (a) sampled at 50Hz worn by a walking participant. The power of the FFT components spikes close to 2 seconds/cycle, indicating the dominant pattern in the data	195
Figure B.2 The Haar Wavelet	196
Figure B.3 Filter banks for computing DWT using the Mallat algorithm showing a filter bank of three scales.....	198
Figure B.4 Successive application of the Haar wavelet transform to the accelerometer signal shown in Figure B.1(a). Each application of the transform decimates information in the resulting signal	199

List of Tables

Table 2.1Activities of daily living commonly studied in existing research using wearable, vision based and location sensing modalities.	29
Table 2.2Sensors commonly used in pervasive sensing applications categorised in terms of sensor type, information measured and information extracted.	32
Table 2.3Features used in sensing applications, adapted from [8]. To capture the complete information in the signal it is often necessary to utilise a combination of features.	40
Table 3.1Features extracted from e-AR sensor signal.	62
Table 3.2Activities performed by participants in the impairment detection experiment. The circuit was repeated with and without braces.	71
Table 3.3Features selected by the Simba algorithm for classification of impairment. The channels from which the feature is extracted is specified in brackets. For example Kurtosis (2) refers to the kurtosis of channel 2, while Covariance(2,3) refers to covariance between channels 2 and 3.....	71
Table 4.1Categories of transitions performed by participants in Study I and Study II.	91
Table 4.2Features used for Impairment Detection from Activities of Daily Living.	94
Table 4.3Accuracy of Transition Detection Algorithm for Study I.	97
Table 4.4Accuracy of Transition Detection Algorithm for Study II.....	97
Table 4.5Accuracy of Transition Categorisation for Study I.	99
Table 4.6Impairment detection using the Step Transition in Study I. Higher accuracy can be achieved by training participant-specific classifiers.	101
Table 5.1Accuracy and compression factor for real and simulated data. The routine tree can take upto 12 times less space than the base dataset on simulated home environment data, however we have smaller gains in real data collection due to greater variation in activity patterns.....	119
Table 6.1Features extracted locally from Accelerometer and Orientation sensors. Six features are extracted from the accelerometer and three features are extracted from the orientation sensor.	130
Table 6.2Mean and Standard Deviation of sampling frequency from sensors on the phone. The phone operating system attempts to deliver sensor data on a ‘best effort’ basis, and does not guarantee constant sampling frequencies.....	137
Table 6.3Activities corresponding to annotations placed by smartphone user, along with the duration of the label.	138

Table 6.4Accuracy of Activity Detection for Study I using first five dimensions of Isomap embedding.....	140
Table 6.5Comparison of Clustering Algorithms with Ground Truth using the Adjusted Rand Index (ARI). GMM performs better than K-Means clustering and is used in the subsequent analysis.....	144
Table 7.1Months for which routine data is available in the SAPHE dataset and the number of days the user wore the e-AR sensor. The user wore the sensor most regularly in July.	168

Chapter 1

Introduction

1.1 Introduction

MODERN sensor technology has benefited from continuing advances in the semiconductor industry, with a large number of commercially available sensors for a wide range of sensing modalities. These sensors are small, increasingly powerful and affordable, driven by continuing demand on miniaturisation and on-board processing power. A sign of the maturity of sensor technology is the recent success of consumer electronics products, such as smart phones and gaming consoles, which utilise a range of different motion sensors to enhance user experience. These developments strongly suggest the suitability of the pervasive sensing paradigm, with which sensors are transparently embedded into our everyday environments for continuous data collection. An instance of this technology is that of Body Sensor Networks (BSN) for providing “*ubiquitous* and *pervasive* monitoring of physical, physiological, and biochemical parameters in any environment and without activity restriction and behaviour modification.” [1].

One of the most important target domains for BSN-based pervasive sensing is healthcare. The population of the developed world is ageing. Trends indicate that the population of the developed world will mostly consist of older people within 50 years [1, 2]. Older people require significantly greater healthcare resources, for both treatment and management of ailments. Due to modern medicine, a large number of diseases that used to be fatal are now treatable. However, many of these, such as diabetes and hypertension, remain chronic conditions that

require continued management over the lifetime of the patient. This has resulted in a strain on healthcare systems in almost all Western countries [3]. The financial burden of burgeoning healthcare needs has led to research into effective, low-cost alternatives to traditional medical practice. The adoption of pervasive technology is one such approach, an example application being the monitoring of key indices of well-being for chronic disease sufferers at home. The potential benefits of pervasive sensing in healthcare are not just financial. Providing care at home increases convenience and independence for patients and social carers. Avoiding frequent visits to healthcare facilities also reduces exposure to dangerous hospital infections. Post-operative recovery can also be measured at home [4], reducing hospital admission times after surgery. Finally, the ability to continuously monitor patient health, physical and physiological data opens up the possibility of a paradigm shift in healthcare, moving away from snap-shot assessment of health to a richer understanding of the behaviour of patients in their home environments, and the relationship between changes in behaviour and exacerbations in disease. Traditional data collection on the other hand has relied on episodic data collection, which necessitates the data collection by care providers, which only offers a snapshot of patient's health status or wellbeing. The contrasting properties of the two approaches are shown in Figure 1.1.

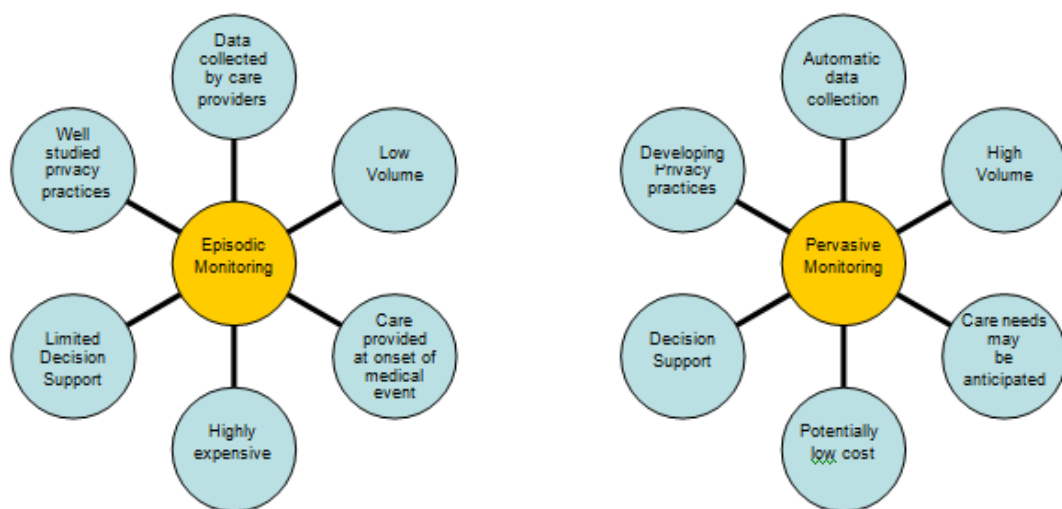


Figure 1.1 The main differences between episodic and continuous monitoring and how the traditional episodic way of patient information management can be avoided by using pervasive sensing.

As pervasive systems are adopted in such applications, a key enabling technology will be the development of flexible, light-weight methods for automatically analysing the continuously sensed data. It is unsurprising that the volume of data engendered by pervasive sensing is beyond the processing capacity of human interpreters, therefore pervasive systems also need to include powerful data mining tools to render this data comprehensible to both expert and non-expert users [5]. It needs to be emphasised that for these systems to be cost effective, the data analysis sub-systems must also be lightweight, in order to simultaneously cater for a large number of users.

One important goal of data analysis in healthcare is behaviour profiling [6]. The onset or complication of a disease may be preceded by changes in patterns of behaviour or activity. Changes in sleeping patterns, social activities or eating, for example, can be due to gastro-esophageal reflux, heart disease and urinary tract infections, amongst others. Changes in gait can indicate recovery from injury or exacerbation of it, or according to recent research, even neurodegenerative diseases such as dementia [7]. The main focus of this thesis is on behaviour profiling in the context of pervasive healthcare applications.

In this chapter, we will discuss in detail major challenges that pervasive healthcare systems need to meet, particularly concerning those of requirements and technology. A description of behaviour profiling for healthcare in pervasive sensing environments follows, including an outline of the key areas that this research intends to target and the main structure of the thesis.

1.2 Challenges in Pervasive Healthcare Systems

As pervasive systems move from research labs into people's homes, addressing real healthcare concerns, it is very clear that there is a wide scope and range of challenges that need to be addressed [8-10]. Some of the system level challenges any pervasive healthcare system must meet are described below.

1.2.1 Clinical Requirements

The healthcare domain where pervasive computing can play a major role has so far been reliant on physical health assessment by professional care providers. Systems like National Healthcare

Service (NHS) employ community matrons to periodically check up on vulnerable or chronically ill patients [11]. A pervasive healthcare system needs to regard the needs of these care providers as very important use cases. Any medical care provided to the patient as a result of the pervasive sensing will most likely come from a care provider being able to track the wellbeing of patients both remotely and during visits. It is important therefore that the visualisations provided by the system be easily interpretable, abstracting out human understandable information from raw sensor data. Decision support systems need to be provided wherever possible, taking into consideration the risks of false positives and false negatives.

The need to abstract and simplify the data is counterbalanced by the need for data in its highest detail for research purposes. This involves collecting contextual data along with the sensor data. For example, it is insufficient to collect sensor data from ten people over a period of time to assess their risk for a particular disease. The data may provide some insight, but for a complete analysis, a large amount of metadata is needed in the system. Information about the participants such as age, ethnicity, weight and much more is all relevant when it comes to analysing sensor data from a medical perspective. This is especially true if the aim is to mine the data and analyse for trends and correlations automatically, over the long term.

This leads to a very different and conflicting concern, specifically that of privacy. There are stringent standards for storing and transmitting user information that need to be adhered to, some of which preclude the extensive meta-data collection that would be desirable for behaviour profiling. We note here that behaviour profiling hasn't always been regarded in a positive light. Due to books like George Orwell's '1984' [12], popular imagination associates the notion with authoritarian regimes tracking citizens in order to control them. There is therefore a strong possibility that the idea of tracking behaviour over long periods may be met with resistance, even in the context of healthcare [13]. People will however accept intrusions of privacy when benefits are well-understood. Beach et al. [14] found that people who reported current disability were more accepting of recording and sharing sensor data than non-disabled adults. They suggested that people with greater needs will be more willing to trade privacy for the benefits technology can bring. As these studies suggest that privacy will remain a significant factor in regulations concerning pervasive sensing, it is important that systems for data mining and behaviour profiling in healthcare explicitly address these concerns [15]. Strategies for this can include high security data transmission, user anonymity, privacy preserving data analysis,

and also by giving users control over what is sensed and for what purpose, explaining the benefits of any perceived privacy violation.

Data collection needs to be suitable for the patient's health condition. A disease such as Chronic Obstructive Pulmonary Disease (COPD) will have very different sensing requirements compared to monitoring someone who is recovering from surgery. Some people spend most of their time indoors, others lead an active life, and the system needs to allow for both and assume neither.

1.2.2 Technology Opportunities

Pervasive healthcare needs to economically scale to hundreds of thousands, possibly millions of users, if it is to account for a substantial reduction in the healthcare costs mentioned above. This mandates that every component of the system, from the sensor to the visualisation of the data be designed with a view towards scalability. A useful idea here is incremental processing. A substantial gain in performance can be achieved by processing the data as it arrives as close to the source as possible and updating these derived statistics when new data arrives [16]. As sensor technology gets cheaper, some of the load can be reduced by performing on-board processing and rejecting data that would otherwise be rejected much further down the transmission pipeline, having incurred the security, transmission and processing cost [17].

The distributed nature of pervasive systems, varying requirements and the need for incremental processing mandates a flexible, customisable yet lightweight software infrastructure. Sensors tend to have diverse data formats, operating frequencies and meta-data requirements. To incorporate the resulting diverse range of information unifying data formats are required to facilitate seamless integration with processing components. Systems should also be able to support a range of information visualisation devices, such as off-the-shelf smartphones. There are a large number of vendors of healthcare products, resulting in a need for platforms that can support the integration of these heterogeneous software and hardware components with minimal effort. XML standards, distributed design patterns, operating system and programming language independence should be a part of this system [15, 18].

Fault tolerance is another critical requirement. Pervasive systems should be robust against local failure [19]. People's health and lives could depend on the successful, reliable, long-term functioning of the system. This includes resistance to hardware failure and software crashes [20]. Data redundancy is extremely important, as are audit trails to ensure that any failures are quickly diagnosed and rectified. Components must be decoupled so that the failure of one component does not cause a ripple effect in the system. Communication failure should be detected, and wherever possible memory buffering should store any sensor data locally until communication is restored [15].

Thus far, different streaming infrastructures have been developed that cater for some of these requirements [15, 21-23]. Shared features of these systems include OS independence, interfaces in multiple languages, a design oriented towards facilitating fault tolerant and decoupled distributed interactions, portability to different hardware platforms, support for other popular distributed frameworks, visualisations components for both computer screens and handheld devices and support for a number of sensors [24].

One promising technology to assist with the metadata requirement is Electronic Medical Record (EMR) systems [25]. Increasingly medical records are computerised in developed countries, or are in process towards computerisation [26]. Integrating pervasive healthcare systems with EMR systems could provide a wealth of metadata to supplement sensor data, which can be used by behaviour profiling and data mining algorithms [27]. Another technology worth exploring is the field of medical ontologies, such as Unified Medical Language System [28]. As pervasive systems discover deeper medical information, capturing that and sharing with the biomedical community in a standardised format will be important.

1.2.3 Data Analysis Requirements

Technological advances such as miniaturised sensors, ubiquitous computing, streaming networks and integrated medical databases need to be exploited with optimised data analysis algorithms. Furthermore, the special needs of the healthcare domain and the nature of the often heterogeneous sensor data imposes significant data processing challenges.

While it is intuitively plausible that the incorporation of multiple information sources will provide a stronger, more robust analysis, care must be taken in the addition of each data source.

If certain sensors provide better quality information, the analysis must be biased towards this information. The noise characteristics of each sensor should be taken into account during fusion. As the number of dimensions in the data increases, data must be pre-processed for example by being mapped into a reduced dimensionality space, or by discarding less useful ‘features’.

Data in pervasive systems is by definition continuous and temporally varying. Analysis algorithms should take into account the temporal evolution of data. Furthermore, as opposed to the traditional paradigm where data is first collected, then processed in large batches, continuous data should be processed as it arrives. It is far too expensive to repeatedly process entire databases; therefore results from previous processing must be updated with new sensor data.

Much of behaviour is determined by a range of factors [29]. People of a certain age, of a certain gender, or geographic residence, with a particular ailment, may all have similarities in their behaviour. As more metadata becomes available, analysis needs to be multidimensional. Data-warehousing allows for the organisation and analysis of data in different dimensions to reflect distribution with respect to such groups.

1.3 Problem Statement

In light of the above research challenges, this thesis investigates a number of issues for behaviour profiling in pervasive sensing in a health and wellbeing context. The goal of this research is to develop a rich understanding of behaviour in a pervasive sensing environment by presenting analysis and visualisation of sensor data collected in experiments designed to highlight healthcare applications. The research has been divided into three parts. The first considers the application of machine learning methodologies for the analysis of activities of daily living for detection of impairment through sensor data. The second part develops methodology for the analysis of transitions between activities with a view to characterise post-operative recovery. The third part concerns the abstraction of activity into a higher granularity, with a view towards visualising vast amounts of human behaviour data and quantitatively analysing an individual’s routine.

1.4 Organisation of the Thesis

The rest of this thesis is organised as follows. Chapter 2 describes the data flows of a pervasive sensing environment, presenting related work and providing an overview of the methods proposed.

Some of the techniques outlined in Chapter 2 are concretised in Chapter 3, which demonstrates the analysis of activities of daily living for the study of impairment, through laboratory simulation of impairment.

The analysis of activities of daily living is extended with the detection and analysis in manifold space of transitional activity, described in Chapter 4. Validation is performed based on data collected from participants recovering from knee replacement surgery to categorise stages of improvement.

In Chapter 5, an algorithm to abstract routine from activity is described. The goal here is to abstract from classification of activity to the processing of historical data. A first step for this is the presentation and representation of routine in a concise data structure, which we call the routine tree.

The analysis of activity and behaviour to pervasive healthcare is then mapped to smartphone users in Chapter 6. The existing consumer base of smartphones can be used for activity and behaviour profiling where the use of specialised sensors is not feasible, or difficult to adopt. We demonstrate the feasibility of this approach through supervised and unsupervised categorisation of activity, and through the long term mining of routine.

Measuring differences in routines is discussed in Chapter 7, with a view towards quantitatively and visually grouping routines discovered through simple activity classification. A technique is proposed for measuring the difference between two time-periods in a routine, and deriving a distance matrix from the comparison of corresponding time periods in a set of routine trees. We will present analysis of data from two sources: a long-term study of chronic patients monitored at their homes with wearable and ambient sensors in the SAPHE [5] and the ActiveMiles project which collects data continuously from smartphone software developed during the course of this PhD.

Finally Chapter 8 concludes the thesis and presents directions for future research.

1.5 Original Technical Contribution of the Thesis

This thesis extends the state of the art in two main areas: transitional activity analysis and profiling of routine in the context of pervasive, long-term monitoring of health and wellbeing. The technical contributions of the thesis are as follows:

- Application of well-established machine learning methods to healthcare through laboratory collected data simulating impaired mobility at the knee and abdomen. The feasibility of behaviour profiling for healthcare applications is demonstrated with this case study.
- Proposing a detection and analysis methodology for short-duration transitional activities that are not commonly studied in activity recognition research. The algorithms are applied to a real world study of people undergoing knee arthroplasty surgery, to track their recovery based on the change in performance of transitional activities.
- Development of a routine-specific data structure, mined from long-term activity data, which can be used for qualitative analysis and visualisation of routine activity patterns. A further utility lies in the compression achieved from the analysis, enabling efficient retrieval and processing of routine related data.
- Extension and adaptation of activity and behaviour profiling methodologies developed for specialised wearable sensors to widely used smartphone technology. In addition to the analysis algorithms, a software platform for Android™ capable smartphones has been developed for the collection of activity and behaviour relevant sensor data.
- Application of behaviour profiling methodologies to real world sensor data including from: knee surgery participants at six stages of recovery, chronically ill users at home wearing activity sensors and normal participants profiled using consumer phones. Quantitative methods were developed for profiling routine and structure in routine using the routine tree data structure.

The work presented in this thesis has resulted in the following publications in peer reviewed international journals and conference proceedings:

- R. Ali, M. ElHelw, L. Atallah, B. Lo, and G. Z. Yang, "Pattern mining for routine behaviour discovery in pervasive healthcare environments," in *International Conference on Information Technology and Applications in Biomedicine, 2008. ITAB 2008.*, 2008, pp. 241-244.
- R. Ali, L. Atallah, B. Lo, and G. Z. Yang, "Transitional Activity Recognition with Manifold Embedding," presented at the Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks, 2009.
- M. ElHelw, J. Pansiot, D. McIlwraith, R. Ali, B. Lo, and L. Atallah, "An integrated multi-sensing framework for pervasive healthcare monitoring," in *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, 2009, pp. 1-7.
- L. Atallah, B. Lo, R. Ali, R. King, and G. Z. Yang, "Real-time activity classification using ambient and wearable sensors," *Trans. Info. Tech. Biomed.*, vol. 13, pp. 1031-1039, 2009.
- J. Patterson, R. Ali, and G. Z. Yang, "Wireless Sensor Network," in *Handbook of Healthcare Delivery Systems*, ed: CRC Press, 2010, pp. 46-1-46-18.
- L. Atallah, G. G. Jones, R. Ali, J. J. H. Leong, B. Lo, and G. Z. Yang, "Observing Recovery from Knee-Replacement Surgery by Using Wearable Sensors," presented at the Proceedings of the 2011 International Conference on Body Sensor Networks, 2011.
- R. Ali, L. Atallah, B. Lo, and G. Z. Yang, "Detection and analysis of transitional activity in manifold space," *IEEE transactions on information technology in biomedicine*, vol. 16, pp. 119-28, Jan 2012.

Chapter 2

Behaviour Profiling with Pervasive Sensing

2.1 Introduction

HEALTHCARE is a key challenge for most countries due to the demographic shift of the ageing population and larger numbers of people living alone with chronic diseases [30]. For the management of chronic diseases such as diabetes and COPD, for example, the functional status of patients is important, particularly for elderly patients, and those recovering from trauma or surgery. To this end, a promising approach is to provide continuous home monitoring through pervasive sensing and smart environments [8]. This approach ensures a lower demand on limited healthcare resources and more efficient monitoring of patient status. Furthermore, it opens a new way for proactive treatment of disease through continuous monitoring, rather than episodic measurement, of physical and physiological features that may be indicative of gradual exacerbation [31]. The technology can also be employed to ensure progress in the recovery of patients, for example from surgery [32], thus reducing time spent at the hospital whilst ensuring appropriate post-operative monitoring and care.

Clinically, activity is an important indicator of health and well-being. Activity and changes in activity have been linked to the onset of disease and changes in life expectancy [33, 34]. Decreased or unusual patterns of activity can signal worsening of condition [35, 36] and poor recovery [37, 38]. People in high-risk age groups are encouraged by healthcare providers to increase their activity through individual and community based efforts [39-41]. A selected set of activities, commonly termed Activities of Daily Living (ADL) [42] are used to infer the

functional ability of the elderly to live independently. Automatic activity recognition, therefore, is an important and widely researched problem [43-45]. Thus far, a wide range of sensors have been used for activity detection, ranging from ambient sensing with cameras [46, 47], miniaturised wearable sensors such as MEMS accelerometers and gyroscopes [48, 49], to a combination of ambient and wearable sensing [50, 51]. The choice of sensing technology determines the range of activities detected. Complex systems however, particularly those which may affect wearability, can inhibit adoption of the technology. There are also privacy concerns [52] associated with ambient sensors that rely on cameras. There is, therefore, significant value in extracting activity information from unobtrusive wearable sensors. Table 2.1 lists some of the commonly studied ADL, along with representative studies on detecting and/or analysing them. For certain activities, such as kitchen activities and falling down, it is often useful to combine wearable sensing with inexpensive and, relative to cameras, privacy-preserving location sensors.

Activity	Types of Sensing used in Study
Walking	Wearable [53], Vision [54], Vision + Wearable [55]
Sitting	Wearable [56], Vision[57], Vision + Wearable [58]
Reading	Wearable [59], Vision [57], Vision+Wearable [60]
Kitchen Activity	Vision [61], Location+Wearable [51]
Running	Wearable [62]
Eating	Wearable [63], Vision [64]
Using Stairs	Wearable [62], Vision [65]
Falling down	Wearable [66], Vision [67], Wearable + Location [68]
Office work	Wearable [56], Vision [69]

Table 2.1 Activities of daily living commonly studied in existing research using wearable, vision based and location sensing modalities.

The need to continuously gather and process large volumes of sensor data, coupled with repeated querying of the database, would introduce significant computational and storage demands. This problem is compounded when a large number of users need to be catered for at the same time. As a result, algorithmic and system- and functional-level complexities comprise significant challenges to pervasive healthcare system developments. In this case, rigid software architectures designed to be application specific cannot cope with the diverse and evolving requirements of this rapidly evolving field of research and development. Recently, a number of light-weight software methodologies suitable for scalable data processing, transmission and storage have been introduced [52, 70]. Synopsis structures [71, 72], for example, can act as substantially smaller visualisation and querying surrogates for the actual data for a specific set

of queries. Techniques such as wavelets, histograms, sketches and sub-sampling can also reduce the resource utilisation of data dramatically, thus freeing up main memory, lowering access rates at the database, and improving responsiveness for web clients. For these reasons, there has been an increasing interest in efficient methods for synopses for data streams, and there is a similar need to explore such representations for sensor data streams in a healthcare context.

One important aspect of human behaviour is routine. Key indicators of wellbeing can be inferred from daily routines, including sleeping habits, social interaction, regular eating, changing of clothes, exercise. A noticeable change in these can indicate a health issue [4]. Behaviour profiling in the context of healthcare refers to the understanding a person's routines, activities and habits often in order to track change over the long term. Physiological measures, such as blood oxygenation, are generally not used to characterise behaviour, although they may be combined with behaviour information to get a more complete picture. Sensors that detect a user's activities and location are more relevant. This includes both body-worn and environmental sensors.

Sensor data in pervasive environment is often transmitted as a data stream. A data stream, loosely defined, is a contiguous sequence of data, usually time-stamped, often continuous, or at least very large. Data streams represent a very different model of data from the traditional database point of view, in that it is not feasible to collect the entire data in a storage space and then process it [73]. Instead, mechanisms are needed to process the data as it arrives to achieve processing goals. This can include generating easily understood visualisations, data compression and indexing, detecting changes and anomalies, and supervised or unsupervised learning.

Data collected from sensors is often noisy and incomplete. This can be due to malfunctions in operation, errors in communication, or motion artefacts in wearable sensors. At the very least, algorithms must be designed to be resistant to such error sources. Alternatively, the issue can be addressed by correcting noisy and incomplete data by using multiple sensors [60, 74]. Missing or noisy data from one sensor is 'corrected' based on values from other sensors taking into account the relationship between the sensors.

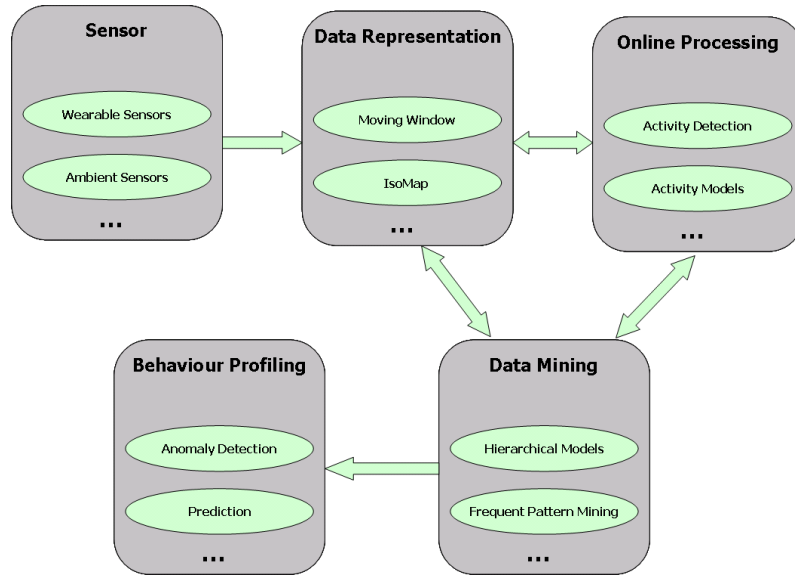


Figure 2.1 Data flow for behaviour profiling in a pervasive sensing environment. Raw sensor data is progressively refined and abstracted as it passes through the machine learning framework.

Figure 2.1 shows the data flow in a behaviour profiling system. An appropriate sensor for monitoring activity is selected. Raw data from this sensor is transformed into representations suitable for storage and processing, which can then be processed online. This could be, for example, to perform activity analysis. Activity information can then be mined to understand behaviour. The arrows between these three components are bidirectional: online processing and data mining can lead to compressed representations suitable for high level behaviour. These compressed representations, referred to as synopsis structures (Section 2.3.1), have much smaller size, and can be used in place of the raw data for a specific set of queries. For example, information mined regarding a person’s routine, mined from sensor data, can be fed back to activity recognition components to provide context allowing more specific classification. Finally, a behaviour profiling system tracks changes in behaviour over time, to detect changes and make predictions about future behaviour.

2.2 Sensing Modalities

Table 2.2 summarises the features of some of the sensors used in pervasive sensing environments. Different sensors can capture information within each broad category, for

example activity, at different levels of detail. A wearable accelerometer, such as in Figure 2.2(a) will provide very specific information about the current activity of the user, whether he is walking, running or sitting. Visual sensors, such as the blob sensor in Figure 2.2(b) can provide detailed user activity information, although their wider applicability has so far been restricted due to privacy concerns. Infrared sensors, such as the PIR sensor (Figure 2.2(c)) can track the movement of a user from room to room, which can be used to characterise typical room occupancy and movement over a person's day.

It is often possible to estimate the activity of the user from sensors that do not provide activity information as their primary measurement. For example, although PIR or GPS sensing is specifically for tracking user location, activity can be inferred based on location. Similarly, changes in heart rate can indicate exertion along with general physiological state, which can be used to infer certain activities such as exercise and sleeping.

Sensor	Type	Information Measured	Information Extracted
Accelerometer	Wearable	Acceleration	Activity, Motion, Posture
Sound	Ambient	Sound, Voice	Location, Activity
Blood Oxygenation	Wearable	Blood Oxygen	Physiology
Contact Switch	Ambient	Opening of doors	Activity, Location
Flow	Ambient	Flow of water for example in Kitchen or Toilets	Activity, Location
GPS	Wearable	Satellite based Positioning	Location, Activity
Gyrometer	Wearable	Angular motion	Activity, Motion, Posture
Infrared	Ambient	Obstruction of infrared light	Activity, Location
Magnetometer	Wearable	Strength of magnetic fields	Orientation, Location
Pedometer	Wearable	Hip movement	Activity
Pulse	Wearable	Heart Rate	Physiology, Activity
RFID	Ambient, Wearable	Identification of objects or use	Activity, Location
Temperature	Wearable	Skin Temperature	Physiology
Video	Ambient	Video, Image	Activity, Location, Movement, Inclination, Posture

Table 2.2 Sensors commonly used in pervasive sensing applications categorised in terms of sensor type, information measured and information extracted.

As an example, the e-AR (Ear-worn Activity Recognition) sensor, shown in Figure 2.2(a) utilises a three axis accelerometer to capture activity and mobility, while also sensing heart rate and blood oxygen saturation. The sensor is based on the BSN platform, which consists of a Texas Instrument MSP430 processor, a ChipCon CC2420 radio transceiver, an Atmel 512KB EEPROM, along with a MEMS (Micro Electro-Mechanical Sensor) accelerometer. Balance in human beings is largely regulated on the basis of the vestibular system in the inner ear, a natural motion sensor. The positioning at the ear is an evolutionary insight. In addition to detecting head movement, shock waves transmitted along the spine are also picked up every time the wearer's foot strikes the ground. This insight is utilised by the e-AR sensor, which is worn at the ear. The e-AR sensor has been used to track post-operative recovery [31, 32], changes in gait [75] and various sports and well-being applications such as rowing, swimming and climbing [76-78].

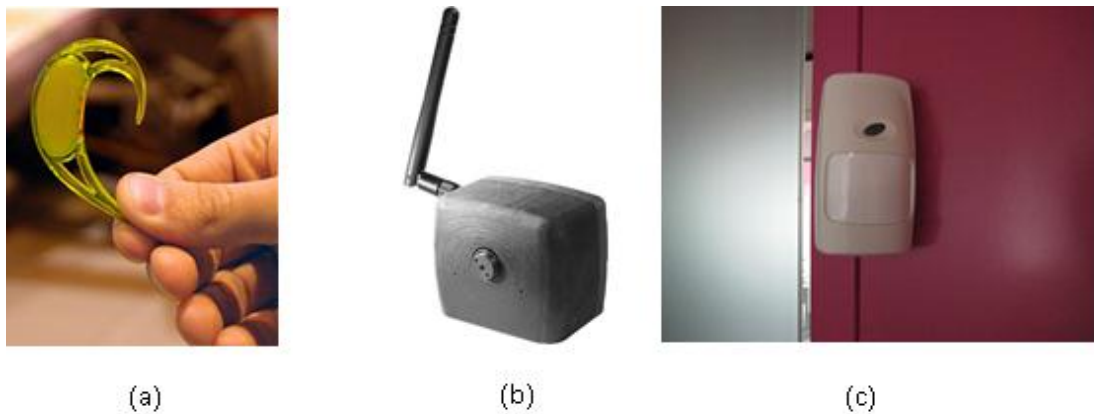


Figure 2.2 Sensors for behaviour profiling – wearable e-AR sensor (a), camera based ambient sensor (b) and Passive-Infra Red (PIR) sensor (c).

Higher quality analysis can be achieved by using fusing data from multiple sources. Yang [1] notes the following improvements of using multiple sensors versus single sensor systems:

- Improved Signal-to-Noise-Ratio (SNR)
- Enhanced robustness and reliability in the event of sensor failure
- Extended parameter coverage
- Integration of independent features and prior knowledge
- Increased dimensionality
- Improved resolution, precision, confidence and hypothesis discrimination
- Reduced uncertainty

The utility of fusing multiple modalities is demonstrated in [50] through an extension of a classifier originally designed for an activity sensor by incorporating Passive-Infrared Sensors (PIR) and Vision based Sensors into the classifier. PIR sensors provide event-based readings allowing an estimate of the most recent user location. They can therefore allow better discrimination between activities that could be confused by incorporating the sensor's context, with fewer concerns related to privacy than camera based sensors [79]. On the other hand, camera-based sensors provide both location and a different modality for activity, based on the change related to motion in the camera images. Fusing these modalities allows for greater specificity in the type of activity performed. Privacy concerns resulting from such image based sensors, and pervasive sensors in general, can be mitigated to an extent by transforming raw sensor data into representations that increase the data abstraction, reduce resource utilisation while aiding in data analysis [80].

The noise and reliability characteristics of sensors can significantly influence the performance of analysis algorithms. For instance, classifiers relying on fused data may be unusable if a sensor fails, or goes temporarily offline. To accommodate for such sensor failures, Atallah et al [74] propose a Gaussian-Process (GP) based algorithm for the prediction of missing values from a failed sensor channel by using sensor channels that are operational. GP regression is a Bayesian approach that assumes a GP prior over the function to be estimated. A GP is a collection of random variables for which any subset of variables has a joint, multi-variate Gaussian distribution. The channels of an accelerometer sensor are modelled as GP. With known covariance and means from training data, errors can be corrected by estimating the missing data from the GP regression on the remaining sensor channels.

2.3 Sensor Data Representation

A consideration for pervasive sensing is representing the data in the system, optimising it for processing in machine learning algorithms and for storage. Raw, unprocessed data is likely to assume unwieldy proportions, incurring both higher storage cost and higher retrieval times. Furthermore, reducing the size of the data may also benefit classification and mining algorithms in terms of reducing dimensionality. This can be seen as a pre-processing step that usually results in representations that are significantly smaller than the original data.

When considering time-varying sensor data, it is also important to consider how much temporal context is needed, which in turn determines how much information is packaged in a single input vector for a data classification or mining algorithm. Short time window analysis is a simple mechanism for breaking the signal into small segments. A window function with a chosen width is shifted across the signal, each segment forming a frame, or input for further analysis. The window function can be uniform, or biased to emphasise a particular portion of the data. A further parameter determines the number of temporal units the window is shifted by. This, along with the width, determines the resolution of the input data. The smallest shift is one unit, in which case for each data point a frame is created. On the other end of the scale is the case where the shift equals the window width. In this case the number of frames is equal to the size of the input data set divided by the width.

Data representation can be considered from two perspectives. From the perspective of database research, synopsis structures are representations of data that are significantly compressed, and are suitable for querying aspects of the original data. These are called synopsis, and are reviewed in 2.3.1. The second perspective is from the field of machine learning, where information is extracted from the raw data to make it more suitable for data analysis. These fall under the topic of feature extraction, reviewed in 2.3.2. There is considerable overlap between synopsis structures and feature extraction in the methods used. The difference mainly lies in the motivation and emphasis of methodologies.

2.3.1 Synopsis Structures

Synopsis data structures [72] are substantially smaller representations of the data that can act as surrogates for the data for a set of queries. There has been considerable interest in efficient methods for synopsis structures of data streams [71] since these continuous sequences of data are voluminous in raw form.

Sampling is one of the simplest ways of reducing the size of the data. Considerable research exists in generating representative samples of the entire data while only processing a segment of the data at a time, and without knowing the size of the entire dataset. A representative approach here is to maintain a fixed reservoir of data from which the sample can be generated [81].

Histograms are a very commonly used summarisation mechanism, where frequency counts are calculated for different ranges. Figure 2.3(b) shows a histogram representation of a data series. A histogram is useful in situations where we wish to visualise the spread of the data as can be seen in the example. There has been work on online construction of histograms [82] for data streams.

Wavelets [83] allow a multi resolution representation of the data that incorporate time and frequency information. Wavelets are curves containing one or more oscillations confined in a finite interval. They are defined by so-called ‘mother functions’ that determine the shape of the curve, and are designed with mathematical properties that permit decomposition of signals. During a wavelet transformation a windowed basis function (based on the mother function) is progressively applied to achieve coefficients at multiple levels. The data can be reconstructed perfectly through an inverse transformation using the complete coefficient set. By discarding coefficients with less information (i.e. higher order coefficients) the data can be compressed significantly, as shown in Figure 2.3 (c) Furthermore, as most of the information is contained in a small number of coefficients, the transformed signal can be encoded much more efficiently using lossless compression techniques such as run-length encoding. The transformation can also be used for smoothing the signal, by discarding higher-order coefficients that represent noise instead of actual data. A more detailed description of wavelets is provided in Appendix B.

A simple representation that has been compared favourably with the Wavelet Transform is the Symbolic Aggregate Approximation (SAX) [84]. In this algorithm data is divided into segments, the aggregate of each segment is calculated to get a Piecewise Aggregate Approximation (PAA). A time series x of length n can be approximated by w segments, where a segment s_i can be calculated by the equation:

$$s_i = \frac{w}{n} \sum_{j=a_i}^{b_i} x_j$$

where

$$a_i = \frac{n}{w}(i-1) + 1$$

$$b_i = \frac{n}{w}i$$
(2.1)

This is essentially equivalent to a sliding window, with each window represented by the mean value of the data falling within it. The PAA segments are then discretised into a symbol string with a given alphabet size. Discretisation is performed by thresholding against a fixed statistical table. Under a Gaussian assumption, the PAA can be expected to discretise into equal-sized bins. Figure 2.3(d) shows a SAX representation of the data.

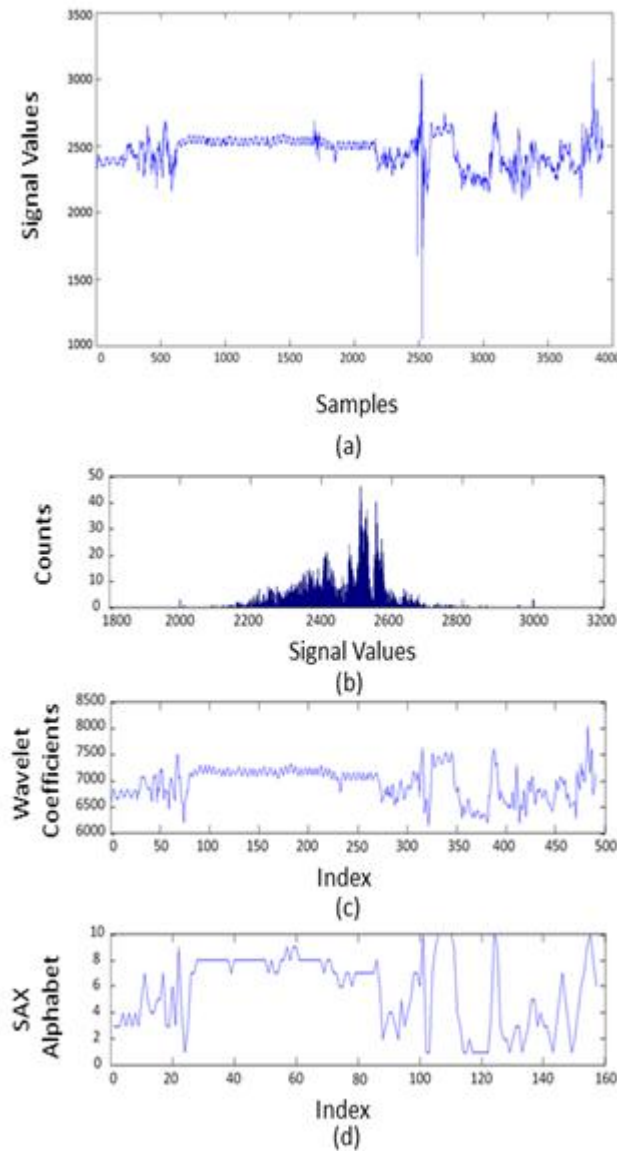


Figure 2.3 Accelerometer Sensor Data (a) represented as data histogram (b), wavelet coefficients (c), and SAX Representation (d). Histograms represent data in terms of counts associated with signal values. Wavelet and SAX representations can significantly reduce numerosity by representing information with fewer data points, as shown in the figure.

The SAX transform results in a reduction of the number of values (cardinality) used to represent the data. The input data has numeric values, whereas the output data is a representation of the data with a much smaller alphabet. This can help machine learning algorithms scale up. In addition to reducing cardinality, the numerosity (i.e. size of the data stream) can also be significantly reduced. As data is represented by a smaller alphabet, and in data series with infrequent variations the alphabet is likely to be repeated often, the series can be represented by recording changes to the current symbol, similar to the run-length-encoding compression algorithm. The size of the output data set depends on the parameter chosen for segment size. However, the data is represented with fewer symbols and a smaller size alphabet. A disadvantage of SAX is that the data is no longer in Euclidean space, and therefore special functions need to be used to estimate the distance between data points.

2.3.2 Feature Extraction

During data analysis, instead of raw data, abstractions called features can be used that can accurately and concisely represent the original information while maximising discriminative power. These abstractions are usually significantly smaller than the raw data, and therefore be considered as synopsis structures. The aim however, is not storage, but to maximise the accuracy of classification and identification tasks.

Yang [1] classifies signal features into time-domain, frequency domain and hybrid categories. This is shown in Table 2.3. Time domain features include signal statistics such as mean, standard deviation, and basic waveform characteristics.

Frequency domain concentrates on periodic structures, such as Fourier components. The wavelet transformed is a hybrid approach combining both time and frequency properties. Features may also be extracted from multiple signals, for instance cross-correlation between two signals.

While it is intuitively plausible that the incorporation of multiple information sources and a combination of features from each source will provide a stronger, more robust analysis, care must be taken in the addition of each data source. If certain sensor features provide better quality information, the analysis must be biased towards this information. This is known as the

feature selection problem. This involves a combinatorial optimisation over the feature space to select subsets of features. Greedy search strategies [85] can seek to begin with the entire feature set, eliminating redundant features (backward elimination) or begin with the most relevant feature and increase the feature set size until the accuracy criteria is met (forward selection). For large feature sets, heuristic approaches have been proposed to make the search efficient, including Genetic Algorithms [86] and Tabu search [87]. The feature evaluation mechanism can be broadly categorised into *wrapper* and *filter* types.

Filter algorithms [88] select features based on an evaluation function that relies on properties of the features for separating classes. The evaluation function is independent of the learning algorithm and may rely on information theoretic [89] and statistical [90] aspects of the feature space. Commonly used metrics for measuring dependence between features include mutual information, covariance and cross-entropy distance. The optimal subset has minimal inter-dependence between features. An alternative strategy is taken by margin-based algorithms [91] that search for features that maximise the ‘margin distance’ statistic. The margin gives an indication of the class separation in the selected feature space. Feature selection is predicated on whether a given feature increases the margin.

Wrapper algorithms [92] use a supervised learning algorithm to evaluate feature subsets for classification accuracy. Some wrapper methods can perform simultaneous feature selection and classifier training [93, 94]. The complexity of the algorithms however is usually high, as evaluations require testing against a trained classifier. Practical considerations may limit the complexity of the classifier to avoid rendering the computation intractable [95]. Filter algorithms are faster and offer the advantage of independence from any particular classification algorithm’s weakness. Conversely, wrapper methods can offer superior classification accuracy [96] because of the use of the associated classifier instead of a fixed evaluation function.

Atallah et al. [97] use the margin-based Simba [98] algorithm to select wavelet-based features that are most relevant for analysis of gait impairment. A related effort has been on the optimal placement of sensors for general activity recognition, and for application specific analysis. For instance, King et al. [99] compare Simba with the Relief [100] and Minimum Redundancy Maximum Relevance [101] algorithms for the placement of sensors for activities of daily living. In their study, all three feature selection algorithms achieved comparable performance. Further discussion and evaluation of feature selection for activity recognition is provided in Chapter 3.

While feature selection algorithms have been successfully applied to a wide range of activity classification tasks, one disadvantage lies in the necessity of class labels. The feature space can be drastically reduced using a Dimensionality Reduction algorithm that proceeds by mapping the full feature space into a reduced dimensionality that preserves the essential variations of the dataset. Well known examples of such algorithms includes Principal Components Analysis (PCA) [102] and Multidimensional Scaling (MDS) [103]. The effect of dimensionality reduction is to be able to present data in a concise form that nevertheless retains the most significant information from the complete feature space.

Dimensionality reduction techniques that assume relationships between distances can be expressed using straight-lines incorporate no information from the structure of the data. Feature space is assumed to be unconstrained, and therefore linear relationships such as Euclidean distance and variance can be preserved in the optimal low-dimensional space. However, if the feature space is assumed to contain structural constraints, straight-line distances are no longer valid. The class of techniques known as Non-linear Dimensionality Reduction (NLDR) [104] performs non-linear transformations on the feature space data. In Chapter 4 we use an instance of NLDR: manifold-based data clustering and low-dimensional embedding.

Time Domain	Frequency Domain	Hybrid
Waveform characteristics such as slope, amplitude, envelope, rise time, width, maxima-minima locations, pulse duration and zero crossing rate	Periodic structure in the frequency domain	Wavelet Representation (e.g. Gabor wavelet features)
Waveform statistics such as mean, standard deviation, peak to valley ratio, kurtosis and energy	Fourier coefficients	Wigner-vill distributions
Information theoretic statistics such as entropy, complexity, compression	Chebyshev coefficients	Cyclostationary representations
Chaotic models and fractal features	Spectral peaks	
Ringing, overshoot phenomena, and pulse ambient noise floor relationship	Power Spectral Density	

Table 2.3 Features used in sensing applications, adapted from [8]. To capture the complete information in the signal it is often necessary to utilise a combination of features.

2.4 Behaviour Profiling

One of the most widely developed applications of pervasive sensing are smart home environments. Smart home technology [105] relies on (usually ambient) sensors to provide intelligent environments that can assist occupants, typically elderly occupants. This includes projects such as MavHome [106], PlaceLab [107] and the Gator environment [108]. The home may also facilitate the operation of devices in the house based on sensed information. The sensors utilised in these projects include temperature, water flow and utility usage sensors as well as pressure sensors on furniture, proximity sensors for tracking user position in rooms as well as devices for monitoring vital signs. Data from these sensors can be analysed to observe patient behaviour or detect the occurrence of critical events such as falls. For instance, Lühr et al. [109] mine activation of these sensors to detect abnormal behaviour.

Tracking these changes also falls under the purview of the field of Lifestyle Monitoring [110], where the time horizon under consideration is typically smaller, generating alarms if a particular sort of behaviour deviates from normal patterns. Lifestyle monitoring typically relies on ambient sensing, i.e. environmental sensors such as PIR sensors and sensors that can detect sleep quality when placed under the mattress. Activity is characterised in terms of the activation of the PIR sensors and sensors on objects.

Atallah et al. [50] note the limitations of using ambient sensing alone for tracking user behaviour. For example, the analysis is complicated by the presence of other users in the sensed environments. Detailed changes in activity and physiological data are difficult to obtain. In addition, wearable devices can also track the user outside their home environments. Ambient and wearable sensors are combined to track the movement of a user across rooms, modelled with HMMs. They also study the similarity of users by visualizing the activation of these models in a reduced dimensionality space.

There has been some use of monitoring through wearable or ambient sensors to encourage specific types of behaviour. For instance, in the field of assisted living, elderly people are given advice on ways to improve their lifestyle on the basis of sensor information [111]. In obesity monitoring, for example, people are helped to keep track of calories spent during activity in order to lose weight [112]. This however has been difficult to do with external sensors, and

subcutaneous devices have been proposed. Previous research related to this thesis can be categorised as relating to analysis of activity and analysis of behaviour.

2.4.1 Activity Analysis

Activity monitoring has been an intensely researched area over the past decade [113-115]. In the context of studying behaviour, it is important not only to detect the user's activity, but also to assess how this activity is being performed. For example, the gait of a user could be indicative of exacerbations in medical conditions. The activities performed can also characterise the extent to which a person is capable of living independently. The Katz Activities of Daily Living (ADL) index [116] is one such measure of the autonomy of a person. Participants are required to fill a questionnaire that assesses their ability to perform a set of activities that includes bathing, dressing, toileting, transferring (transitioning between postures), continence and feeding. The assessment results in a score determining the functional independence of the participant. A goal of pervasive healthcare is to substitute such questionnaires with objective processes, such as sensor-based data [117]. The sensor signal obtained when a participant performs some of these activities is shown in Figure 2.4.

In general, activity analysis can be divided into two broad tasks. The first is *activity detection*, which is the near-instantaneous detection of current user state based on a typically small time horizon. Subsequently, temporal aspects can be introduced, which is called *activity modelling*. To some extent, the distinction between the tasks can be seen on the basis of complexity. In activity detection, "atomic" activities such as standing, sitting, walking etc. are detected possibly for subsequent profiling and analysis. In activity modelling, more complex activities temporally composed of atomic activities may be detected, for example, food preparation and eating. Activities can be interleaved, suspended and resumed, or performed concurrently.

A first step in any system aimed at behaviour profiling is to detect atomic activities of users. This can range from providing specific information about the user's current activity, such as brushing teeth or reading, to abstracted, yet still medically relevant information such as the current intensity of activity. An example of this type of activity recognition is the work by Ravi et al. [62], where the activities recognised include standing, walking, running, climbing upstairs and downstairs, sit-ups, vacuuming, and brushing teeth. Activities were detected using data

collected from a triaxial accelerometer worn on the pelvic region. Four features are extracted from the data for each axis: mean, standard deviation, correlation, and energy. The authors compare the performance of different classifiers using these features.

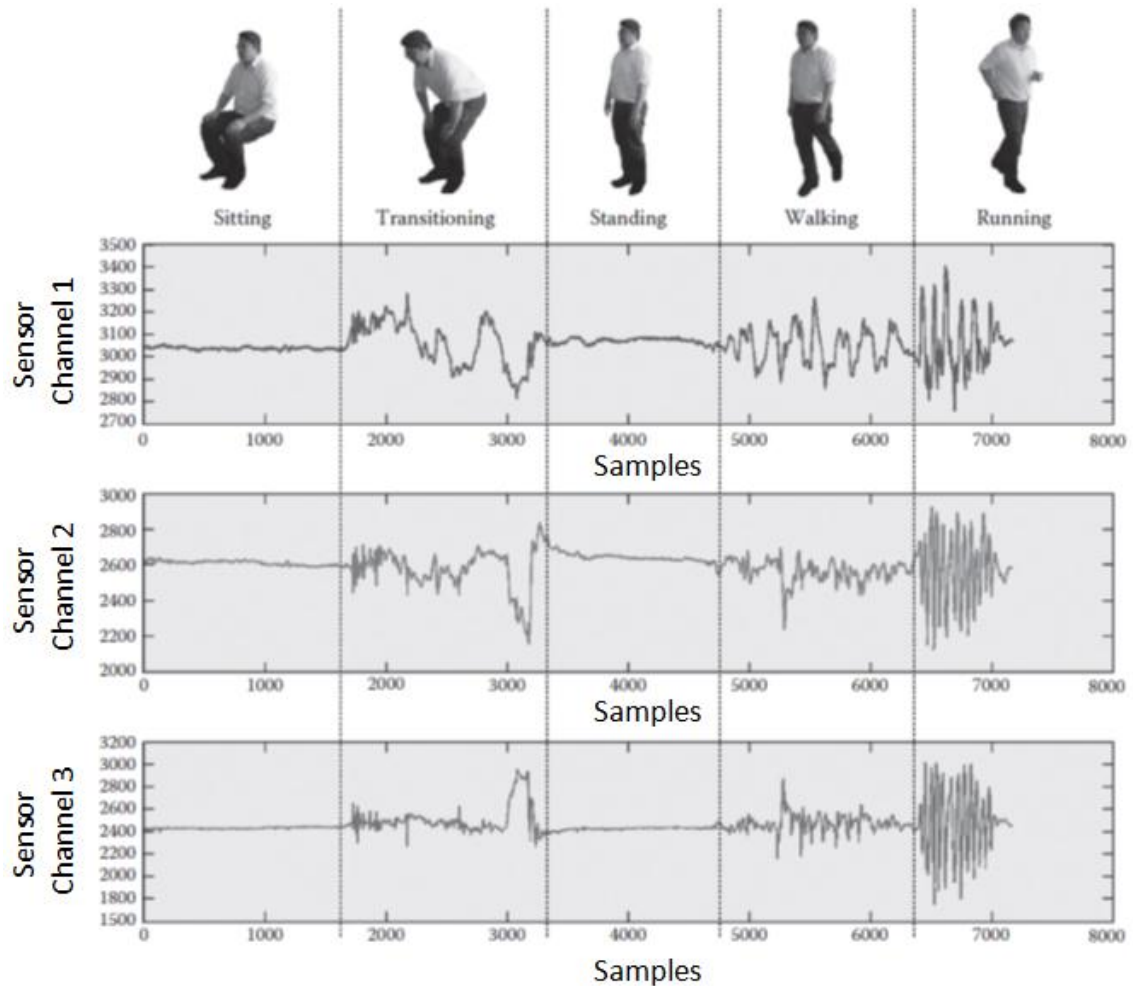


Figure 2.4Activities of daily living can be detected using wearable sensors such as the ear-worn activity recognition sensor. The change in sensor data can be seen for each activity performed by the participant.

While precise information about a user's activity provides valuable information for further mining, a pragmatic reason for avoiding this is the issue of user privacy. Lo et al. [115] have developed a multivariate Gaussian Bayes classifier that produces an activity level from an ear worn accelerometer. The classifier defines activity in four values, the lowest of which indicates almost no movement (e.g., sleeping) and the highest indicates an activity involving vigorous movement (e.g., running). Quantitative measures of activity have been used in cardio-

respiratory fitness studies [118] and postoperative recovery [31, 54, 119]. Low levels of activity have also been associated with type-II diabetes [120].

It is worth noting that most of the existing work on activity recognition is limited to detecting atomic activities. Sequences of atomic activities can be temporally composed into more complex activities through generative models such as hidden Markov model (HMM) [121]. In addition to their modelling power, HMMs amortise the computation cost and can be trained offline and deployed online with relatively small computation cost. The HMM shown in Figure 2.5 (a) contains two layers: an observable and a hidden layer of states. The hidden states are governed by a Markov model—a model where future behaviour depends only on the present state. Transitions between such states are determined by state-transition probabilities. At any point in time however, the state is not directly accessible. Instead, the agent has access to observations, which are related to the hidden states through observation probabilities; i.e. the probability of encountering an observation in a particular state. There is furthermore a prior state probability distribution. Given these parameters it is possible to

- Generate a sequence of observations consistent with the HMM. For this reason models such as HMM are called *generative models*. A learned HMM model can be used to generate behaviours consistent with the model.
- Determine the probability of an observation sequence. A HMM can be used to compare against new sequences to determine difference. This has been one approach taken to compare behaviours, as in [60] where anomalous behaviours are detected.
- Probability of the current state. An HMM can be used to determine the likely state of the participant following a sequence of observations. This is a common use-case during the behaviour profiling applications that will follow, and can be computed efficiently using the ‘forward algorithm’.

In Figure 2.5(a), A1 can represent a simple activity, such as preparing breakfast. The hidden states, shown as circles, indicate separate stages that comprise preparing breakfast, for example, preparing ingredients, cooking food, and serving it. Example observations would be sensor readings that can probabilistically indicate each of these hidden states.

Extensions of HMMs allow modelling of more complex and interleaving activities. For example, a coupled HMM [122], shown in Figure 2.5 (b), allows the modelling of concurrent activities by utilising a collection of HMMs. In this case, the model can “switch” between

independently evolving Markov models, A1 and A2. Another extension of HMMs is that of hierarchical models. Hierarchical approaches for modelling activity and behaviour mirror research in the field of ethology (the biological study of behaviour) where hierarchy has been shown to be underlying certain kinds of behaviour [123]. One example of such an extension is work is the abstract hidden Markov model (AHMM) [124]. Plan recognition is the artificial intelligence problem of inferring an intelligent agent's plans—a hierarchy of actions that allow an agent to carry out its goal. These actions however are deterministic, and the observations are error free. This limitation is resolved by the AHMM, where policy states are connected to an underlying HMM. A system based on AHMM discriminates between sequences of actions, such as different strategies of movement in an office space. Based on this, the user is classified as having a particular “plan” based on a predefined plan library. AHMM offers a powerful model for defining how low-level activities compose into higher level plans and allows the inference of the goals of participants based on their behaviour.

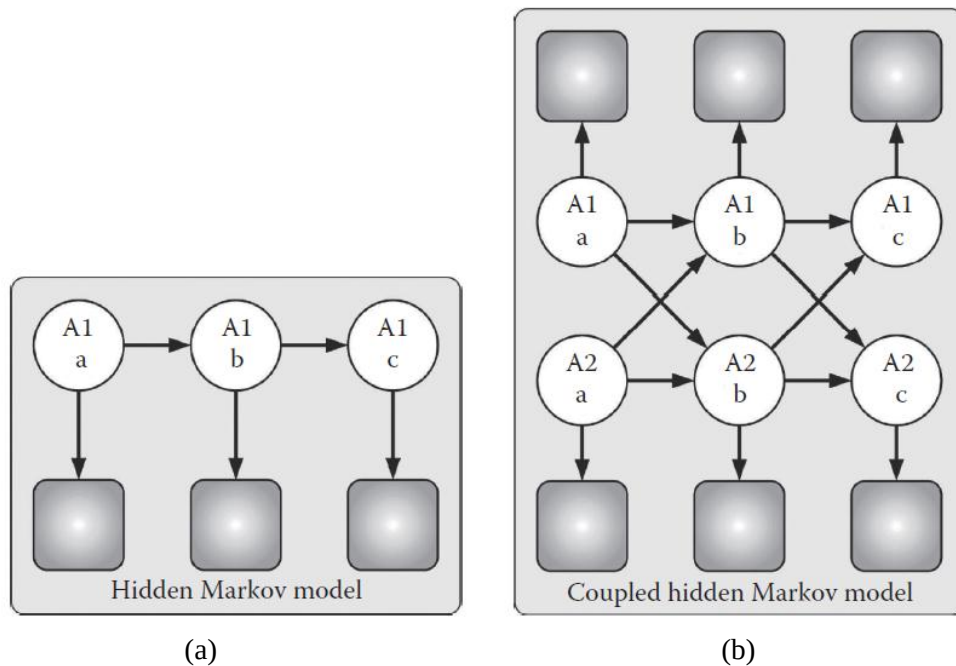


Figure 2.5 HMMs used in activity recognition: (a) illustrates a simple activity, such as preparing breakfast, where the circles represent the constituent tasks involved in the overall process and shaded boxes represent observations made by sensors. For more complex activities, a coupled HMM (b) allows a collection of HMMs to represent multiple interleaving activities.

The computational capacity of sensors has made the idea of placing analysis algorithms on the sensor itself feasible. The benefit of this is significant. Not only does incremental processing decrease processing costs further down the pipeline, but transmission and security costs can be saved if data is processed on board and sent in a reduced form. For instance, if the activity state of the user can be reliably detected, this information can be used to adapt resource utilisation of the pervasive system. With domain specific information on the usefulness (or utility) of particular activity states, the resource utilisation of sensors can be adjusted according to the current state with a view to maximizing utility [125, 126]. In this case, several sensors are used to monitor a patient's state, with known utility for specific states. Utilizing a genetic algorithm-based controller, the operation of sensors is adjusted in order to match the quality of information with the utility of the patient state. Here quality is measured in terms of both the sensing modality and the sampling rate. While one method relies on a centralised controller residing on the gateway [126], the controller is shifted to the nodes themselves by Anand et al. [17], relying on a simpler Markov model to perform resource management. The controller is also shown to operate in more dynamic environments; the controller meets design imperatives (e.g., a minimum operating life) when the system model is not close to the real deployment conditions.

Changes in how specific activities are performed can indicate a change in the mental or physical state of participants. For instance gait changes can suggest injury or impairment, but are also associated with certain neurological conditions; for instance it is notably an early indicator of Alzheimer's disease [7]. Using inexpensive wearable sensors, it is possible to detect changes in walking. Yoshida et al. [127] use a waist-worn accelerometer to detect leg injury by analysing signal frequency in the anterior plane. Atallah et al. [97] develop a broader approach using wavelets to capture frequency and time-domain information from a tri-axial accelerometer worn at the ear. The most relevant planes of motion are selected automatically using feature selection, instead of using a fixed feature set.

2.4.2 Transitional Activities

A recent direction of research has been to analyse the transitions between activities. Joint diseases such as arthritis, trauma or other conditions such as obesity, can impair the ability of patients to fluidly transition between activity states. More than two million people over 65 are estimated to experience difficulty in rising from a chair [128]. This difficulty has been

associated with the likelihood of falls [129, 130]. The Sit-to-Stand (STS) transition also been studied in association with stroke [131, 132], neuromuscular conditions [133], Alzheimer's [134] and chronic lower back pain [135]. A natural application therefore arises to detect impairment in mobility based on this transition. A further application is compliance with physiotherapy guidelines on the optimal method of transitioning for such patients. Examples of such transitional activities are shown in Figure 2.6.



Figure 2.6 Participant performing transitional activities demonstrating the gradual postural change between different stable states.

While there is abundant clinical research in laboratory settings, there has been limited work in developing pervasive systems to detect transitions. Recently, a decision-tree classifier is trained to discriminate between sit-to-stand strategies based on single and multiple camera sensors in Allin and Mihailidis [136]. The strategies studied were divided into arm and foot strategies simulating movement patterns adopted by people suffering from impairment. Arm strategies include: no use of arms while rising, use of arms to push from the seat, and generating momentum by swinging the arms. Foot strategies were comprised of extension of the knee at 80, 90 and 100 degrees. Using image features C4.5 based decision trees were trained to recognise strategies. The authors found it was possible to achieve high accuracy levels when multiple cameras were used.

Najafi et al. [137] use a gyroscope to record STS transition extracting the average and standard deviation of transition durations and the occurrence of abnormal successive transitions. The measures extracted were correlated with risk of fall determined by standard biomechanical metrics such as the Tinetti score [138]. In both of these studies, participants performed exclusively sit-to-stand transitions in controlled settings. A challenge in translating laboratory research into home environments lies in detecting transitions in activities of daily living, and analysing transitions performed at patient's own home environments.

In Chapter 4 we propose a methodology for the detection and analysis of transitions in e-AR sensor data, applied to the tracking the recovery of knee-replacement surgery patients. Our methodology is not tailored towards any particular transitional activity, which is an advantage when compared to approaches [139, 140] that focus on detection of specific transitions using supervised classifiers. Furthermore we rely on a single unobtrusive sensor, designed for long-term home-monitoring with a view towards proposing a procedure that can be used to record recovery from home with significantly reduced dependency on laboratory data collection.

2.4.3 Modelling Behaviour

With the emergence of pervasive sensing technologies, the goal of behaviour modelling is shifting from modelling and detecting individual activities to understanding the typical structure of a person's activities. One instance of this is capturing daily routines. Living beings have circadian rhythms [141], 24-h cycles in their behavioural, biochemical, and physiological processes. Healthy people have characteristic circadian rhythms, deviations from which can indicate a change in the state of health.

Differentiation between behaviour and complex activity can be somewhat difficult, with the same activities being referred to in literature as activity and behaviour. Many techniques for behaviour modelling can be seen as extensions of hierarchical activity modelling techniques discussed in the last section. There is theoretical basis for grounding behaviour models in hierarchical approach as this mirrors research in the field of ethology (the biological study of behaviour) where hierarchy has been shown to be underlying at least certain kinds of behaviour. For further information on ethological and psychological methods for studying behaviour we

refer the reader to Dawkins [123] and Martin and Bateson [142]. Examples of such models of behaviour include Layered HMM [143] and Factorial HMM [144], compared in [145] and [6].

A limitation of these approaches however, is the need to develop and train models in the first place. Furthermore, since there is considerable variety in the behaviour expressed by an individual, and each type of behaviour has significant complexity, specialised models may be needed for particular activities. For example Chen et al. [144] devise a Factorial HMM method for analysing gait. While this allows very detailed analysis, it may be impractical to apply this in a pervasive home-monitoring scenario, given the volume of the data, and the lack of access to detailed annotations.

A related effort to discover social behaviour using simple sensors was conducted by Wren [146] where simple ambient sensor firings were used in a crowded office environment. The challenge here lies in the discovery of more complex information from sensor events in a complex environment. The first step undertaken by the researchers was to uncover a map of the sensors by analysing the co-occurrence of sensor activations. Using Multi-Dimensional Scaling on a matrix of Co-Occurrence times, the approximate locations of the sensors was determined. The subsequently associated journeys undertaken by office workers were modelled as sequences using a composite hidden Markov model. A similarity matrix is first derived from paths taken in the office environment by training simple HMMs for each sequence. The HMMs thus trained are used to find sequence distances, similar to the approach taken by Atallah et al. [50]. These distances are used to cluster the sequences. Each of these clusters is used to train a composite HMM.

Dynamic Bayesian Network (DBN) [147] represents sequences of variables in a Bayesian Network. HMMs are a simple form of DBN. The unrestricted nature of DBNs in general makes them hard to solve in certain configurations, for instance in the case of loopy graphs. DBN based solutions are particularly suitable for cases where multiple modalities need to be fused together to perform classification. For instance, DBN is used to model motion at multiple granularity in [148] by differentiating between global features and local features. Hierarchical DBNs can model complex activities by the composition of simpler activities at lower levels. Low-level activities can be automatically discovered using clustering, as in [149], where a hierarchical framework is applied to discover unusual behaviour from a video database. A hierarchical DBN is used in [150] to model travel using GPS readings, information from bus

stops, along with features extracted from the motion of the traveller (such as mode and velocity of transport).

Behaviour is manifested over a period of time, and therefore methods designed to analyse large datasets, such as those developed in the field of data mining, can be utilised. Frequent pattern mining [151] is one such problem in data mining, related to searching for frequently repeating patterns in a database. A pattern is considered to be frequent if its occurrence in the database (called support) is above a user-specified threshold. The first applications of these algorithms focused on supermarket data, which is one reason why the algorithms are also known as ‘market-basket analysis’. This problem has exponential complexity if performed with a brute force search. Algorithms for efficiently solving the problem focus on pruning the search space. Extensions may also seek to find more complex patterns; for instance Lühr et al. [109] use an extension of frequent pattern mining in a smart home to model temporal relationships in activities. Data are obtained from activations on object sensors. Associations between these activations are mined over the long term, and the extended mining algorithm allows these associations to span database transactions.

An example of these approaches can be learning typical paths taken by users in moving between rooms or the sequence of objects activated during a typical kitchen task. One of the key problems in behaviour profiling is to determine which manifestations of behaviour actually indicate the onset of an adverse event as opposed to normal variations. This relates to the concept of Interestingness, i.e., the importance of discovered knowledge to the application at hand. Ohsaki et al. [152] consider this with respect to the analysis of medical data. Recent work by Jorosziewicz [153] has sought to improve the accuracy of automatic approaches based on pattern mining by involving a domain expert when to select interesting sequences from automatically found patterns to train HMMs interactively. While the authors proposed the technique for mining web logs and protein databases, characterising the utility of patterns in a BSN environment is essential to train models to be used by non-specialist users.

Structure in social behaviour was identified in [154] by using mobile-phone data during the ‘reality mining’ study. The study utilised Bluetooth, phone usage and location data, and extracted principal components of the dataset. A weighted sum of these components approximated an individual’s behaviour. The study used this measure, termed Eigenbehaviours,

to cluster people and analyse social groups, for instance to analyse friendship and work group affiliations.

Routine activity is an important aspect of behaviour. One of the main influences on daily routine is the circadian rhythm, a natural regulation of hormones that determine rest and activity cycles. There has been research primarily in the smart-home area for associating ambient sensor statistics with the circadian rhythm, with anomalies detected for significantly large variations in the expected activity [155, 156].

2.4.4 Detecting Anomalous Behaviour

Anomaly detection algorithms seek to find deviation from normal behaviour in a data stream. It involves detecting meaningful change in the activities, transitional activities, or routines of the user. The variable nature of human activity and behaviour makes this a challenging area of research.

The movement of a user from room to room can be represented as a sequence of symbols on a temporal grid. Two such activity grids are shown in Figure 2.7 to illustrate different behaviours. Figure 2.7 (a) shows the grid for a user who spends most of their time in the living room and kitchen while Figure 2.7 (b) shows the grid for a user who stays either the bedroom or toilet. The latter case may be indicative of a health issue in need of intervention. One complication in comparing behaviours is the alignment of sensor sequences. Participants may perform the same behaviour differently, or with small variations. In this case it is a challenge to compare the difference between sequences of sensor data. One approach is to find the distance between sequences using HMMs. For each sequence, an HMM is trained on a selected reference sequence from the dataset. These trained HMMs can then be used to analyse new sequences based on the distance of the given sequence to the trained models. Sequences with large distances can be identified as anomalous. Another commonly used technique for aligning sequences is to use Dynamic Time Warping (DTW) [157], which finds the optimal match between two sequences through a recursive application of simple operations such as insertion of the same component, or deletion of a component. DTW has been used in the context of anomaly detection by Kan and Dailey [158] to detect anomalous behaviour in a video surveillance system to cluster sequences of features extracted from video data before training an HMM to model the

clusters. DTW has also been used for activity recognition in ambient sensing environments [159] by matching low-level sensor activation sequences with templates, a technique that allows for variations in the activation sequences and stored templates due to noise.

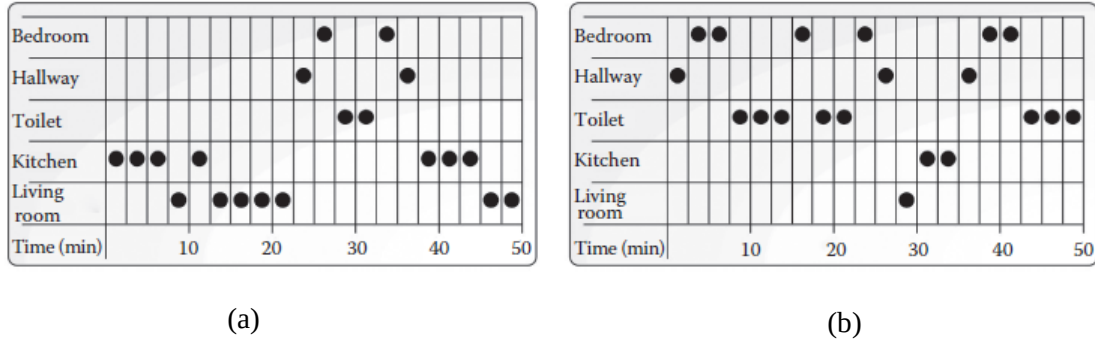


Figure 2.7 Intuitive activity grids used to visualise a user’s overall behaviour pattern in a home environment. These two simulated activity grids illustrate contrasting behavioural patterns: (a) the user spends most time in the living room and (b) more time is spent in the bedroom and toilet indicating anomalous behaviour.

Li and Parker [160] utilise a Fuzzy-ART (adaptive resonance theory) neural network supplemented with a Markov model to detect abnormal events generated by ambient sensors. The neural network categorises raw sensor data, with each category representing a state in the Markov model, which then learns the state transitions during normal behaviour. Detection of anomalous behaviour is based on state occupancy time by recording the average time spent in each state during normal behaviour.

One such approach is a behaviour profiling system for elderly patients residing in a smart home [161]. The smart home is equipped with ambient pyroelectric (i.e., location) sensors tracking the time a patient spends in each room. Behaviour is tracked by maintaining a Gaussian mixture model (GMM) of room occupancy duration at different time periods. Changes are detected at two scales. The first is at the level of “local anomaly” where an outlier detection algorithm is used to detect behaviours with unlikely timing or duration. For example, an inordinate time spent in the bedroom would be classified as an anomaly. The second is a “global anomaly” where changes in the model are tracked over longer time periods. Daily differences in behaviour are computed and an anomaly is flagged if a sudden change in one day exceeds a threshold after discounting for seasonal variations.

Virone et al. [155] associate circadian activity statistics of elderly participants in a smart home with every hour of the day, while Barger et al. [156], while not associating explicit statistics with every hour, also begin their analysis with clusterings of hourly firings of ambient sensor data. Generating hourly mean and standard deviation values for activity levels with a view of approximating circadian rhythms may be too limited a framework to account for real-life variability in routines. This variability can stem from many factors, such as social, personal, and environmental. Models that articulate routines only in terms of circadian activity conceptually ignore such factors. Night-shift workers are an example where circadian activity is not the main driver of routine activity. Furthermore, hourly statistics impose artificially rigid boundaries on human schedules, which are actually comprised of activity patterns that often have irregular starting times and variable duration. This exaggerates differences between similar, but temporally shifted routines. Finally, human behaviour can be expressed only to a limited extent in terms of average hourly activity or room occupancy times. While such metrics may be reasonable for simpler ambient sensors such as PIR, more expressive models should be considered for camera and wearable sensors that provide more information. In Chapters 5-7, we will develop a multi-resolution data structure for routine behaviour that associates time periods with patterns of activity.

The distance of a data-point from the ‘normal’ distribution can be measured in different ways. The simplest distance utilised, which assumes a spherical distribution of data is the Euclidean distance, which assumes spherical data distribution. A more powerful technique is the Mahalanobis distance, which takes into account the covariance of the data. If the covariance matrix is an identity matrix, the Mahalanobis distance reduces to the Euclidean distance. Formally for a group of size p with mean $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_p)^T$ and covariance matrix Σ the Mahalanobis distance D_M is defined for a data point $x = (x_1, x_2, x_3, \dots, x_p)^T$ as:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (2.2)$$

Tarassenko et al. model a dataset using a GMM, and use the smallest Mahalanobis distance between a data-point and existing Gaussian mixtures to determine whether or not to add a Gaussian unit during training. Their method has been demonstrated to work with high accuracy to detect epileptic seizures based on EEG data [162].

Parzen windows approximate a training dataset with a linear combination of kernels, which are frequently chosen to be Gaussians, centred on observation points. Formally, for $x_1, x_2, x_3, \dots, x_N$ variables belonging to distribution f , the Parzen window approximation is

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^N \phi\left(\frac{x - x_i}{h}\right) \quad (2.3)$$

where h is the bandwidth or smoothing parameter and ϕ is a kernel function. This approximation is particularly useful when there is not enough training data to accurately construct parametric models such as GMM. Tarassenko [163] used Parzen windows for the detection of masses in mammograms. The method has also been applied for network intrusion [164].

2.5 Conclusion

As pervasive systems move from research laboratories into people's homes and from toy problems to addressing real health-care concerns, key areas of research emerge in applying sensor data to profile behaviour.

The first problem any data-mining algorithm has to contend with is the problem of data scale, and limited annotation. Conversely, it is not always feasible to deploy completely automatic methods, as the domain knowledge and end-user status belongs with groups that are unlikely to understand abstract labels. Algorithms should seek to combine supervised with unsupervised machine learning methods. The trade-off between the difficulty of precise annotations and the comprehensibility of algorithm output needs to be carefully explored, with real-world deployment scenarios in mind.

There is considerable research towards developing scalable algorithms to detect and track behaviour from wearable and ambient sensors, with applications ranging from the detection of atomic activities, to the modelling of complex behaviour. An emerging area in this field is that of transitional activities: subtle activities that occur at the boundaries of commonly studied activities of daily living. Once such activities have been classified, a subsequent step is the analysis of change in their performance over time, which can indicate the presence or exacerbation of a patient condition requiring intervention.

Hierarchical models can abstract from simple to more complex activities, leading the way to the analysis of longer term behaviour. It is necessary to cater for the scale of the data as pervasive sensing systems collect data over larger durations. Data structures that present the summary of the user's behaviour without necessitating the reprocessing of the activity stream will need to be developed. Such structures will need to serve the dual purpose of compressing the activity stream as well as enabling mechanisms for the long-term profiling and classification of the participant's routines.

The purpose of this thesis is to propose techniques for the analysis of wearable sensor data, supplemented with location information, to analyse behaviour at the small scale of transitional activities, to long-term mining of behaviours to detect changes in routine. One of the main goals will be the scalability of methods, and the reliance on semi-supervised approaches by combining the strengths of clustering with supervised learning based classification. One of the main goals of long-term care is the timely provision of emergency care. An important use-case is therefore the detection of anomalous activity or behaviour. This applies equally to short-term activities and long-term behaviour. We will therefore demonstrate at each stage of our analysis the ability to detect unusual patterns in the sensor data.

In the next chapter we concretise the above surveyed research with a case study of a medical application that highlights the strengths of existing research. The study is based on an existing laboratory dataset. Aspects of the study will allow us to highlight opportunities for research that this thesis will pursue.

Chapter 3

Detection and Classification of Activities of Daily Living

3.1 Introduction

In the last chapter, we have outlined the key technical challenges related to activity monitoring and behaviour profiling with pervasive sensing. In this chapter we will concretise some of the concepts reviewed in the last chapter by applying machine learning and pattern recognition techniques in controlled laboratory experiments. The purpose of this chapter is to present some of the main data processing techniques that will be used in the remainder of this dissertation.

Machine learning approaches for activity analysis typically adopt a number of standard steps, which include both supervised and unsupervised techniques. For supervised classification techniques, analysis usually begins with visual inspection of the raw data, to select segments of the data most relevant for the analysis goal. This introduces a complexity when considering large volumes of data. Due to resource constraints, it may only be feasible to examine and label a small subset of the data. Furthermore, manual labelling introduces the risk of human error. Several automatic segmentation techniques have been proposed to deal with this problem, offering the possibility of a ‘mixed mode’, whereby segmentation and even labelling task is partially done automatically.

Once data is labelled and segmented, the first consideration is selecting the most useful information from the raw-data. As discussed in the previous chapter, a large number of features can be extracted from the sensor signal. The goal of feature extraction is to accurately and

concisely represent the original information while maximising discriminative power. Subsequent classification and modelling of activity may operate on the complete set of features, or may seek to reduce it by using class labels to select the most relevant features. This is referred to as feature selection. For instance, Atallah et al. [97] use feature selection to select the optimal wavelet scales for classifying changes in gait. Dimensionality reduction algorithms complement feature selection by mapping the full feature space into a space that is both lower in dimension, and concisely representative of the true or intrinsic dimensionality of the dataset. The algorithms typically do not require class labels, which is an attractive property in continuous home-monitoring scenarios. Related to the problem of feature selection is that of sensor placement. Sensor placement studies, such as [99] have classified body locations based on classification accuracy for ADL recognition using inertial sensing. In addition to data quality, however, where the sensor is placed on the body is determined by comfort and ergonomics and safety. Complex systems involving multiple sensors may discourage adoption. For this reason, in this thesis we will focus on developing analysis based on single devices. Although multiple sensors are used in Chapter 6, they reside on the same device.

If multiple sensors are being used, one design consideration is the stage at which the signals can be fused to produce a consolidated input for analysis. There are three stages where the fusion is typically performed:

- Data level – The raw sensor signals may be aligned based on the timestamp. This is also referred to as data alignment. The consolidated data is used to generate features. An example of this is in King et al. [165], where data from multiple sensors is synchronised before features are extracted.
- Feature level – Features may be extracted separately from the sensor signals and consolidated before classification or clustering, as in the approach taken by Pradhan and Prabhakarn [166] the fusion of body-worn accelerometer and orientation sensor derived features for human activity or
- Decision level – Classifiers may be trained separately for each sensor, with a higher level classifier for fusing classification labels from each source. This is the approach taken by [167] for the correlation of ambient and wearable sensor data streams.

In addition to these commonly used techniques, it is also possible to utilise information from a sensor, or a sensor channel to improve the quality of the data from other sensors. For instance, Atallah et al. [74] use a Gaussian process to correct for noise or missing data in one channel from other channels. Similarly, Talukder et al. [168], balance the bias induced from low-quality, inexpensive sensing and conserve power of high-quality, expensive sensors using the covariance of the sensors for specific participant states.

The next step in the machine learning pipeline is to apply classification or clustering algorithms on the features extracted from the data. The choice between classification and clustering is a trade-off between specificity of analysis output versus independence from the annotation. Within these two broad categories, there is significant overlap of methodologies, and specialisation of algorithms for particular applications.

Machine learning algorithms tend to draw on extensively developed sources: function approximation algorithms such as artificial neural networks; statistical inference based algorithms most significantly Bayesian learning; decision tree based algorithms such as C4.5. If a model for generating new data is included in the algorithms, they are referred to as generative models. Examples of this include Gaussian Mixture Models, Hidden Markov Models, Probabilistic Context Free Grammars. If the probability distribution of the labels to data is modelled directly, as in neural networks or decision trees, the algorithms are known as discriminative models. While there are known algorithms that outperform others for specific applications, it is difficult to know beforehand which algorithm will be best suited for a given machine learning application.

In summary the typical activity detection pipeline, shown in Figure 3.1, broadly consists of:

- 1) Preparing sensor data for analysis, performing preprocessing (e.g. fusion, filtering) if necessary.
- 2) Extracting features from the raw sensor data, and optionally reducing the dimensionality of the data fed to subsequent steps using feature selection or dimensionality reduction algorithms.
- 3) Generate labels for the data using classification or clustering algorithms.

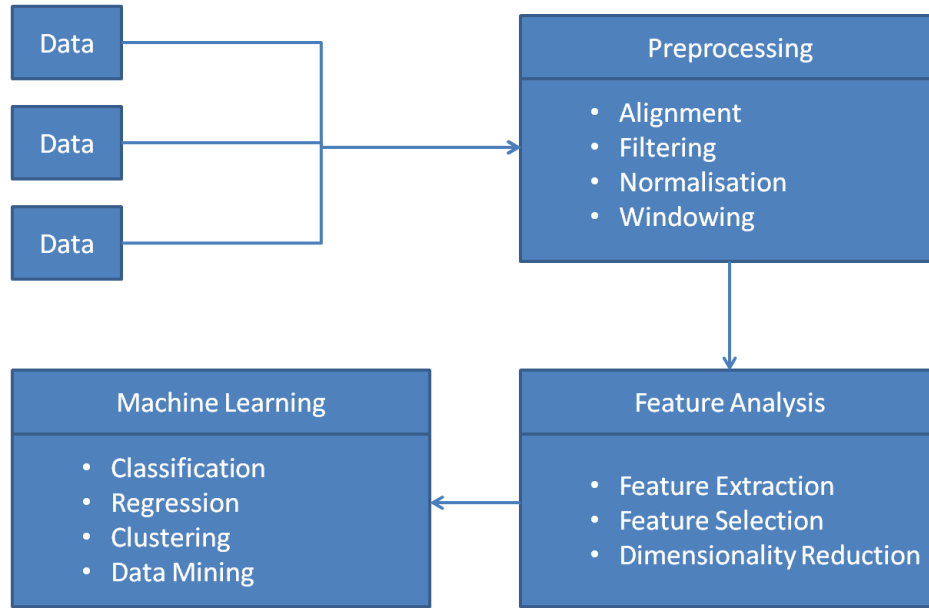


Figure 3.1 Typical machine learning pipeline in machine learning applications. ‘Data’ refers to sensor data, or features extracted from sensor data. Typically preprocessing is required before the data can be further analysed. Feature analysis can consist of selecting optimal feature sets or reducing the dimensionality after feature extraction. Finally, machine learning tools can be applied for particular applications.

Once labels are generated, the goal of the analysis is to proceed from a stream of labels to a higher level output. This can be in the form of information richer labels; for instance the change between sequences such as “Walking, Standing, Sitting, Sitting, Standing, Walking” to “Going home”. A number of probabilistic algorithm based techniques were surveyed for this analysis in Chapter 2.

In this section we will apply some of the techniques surveyed so far to concrete problems. This will illustrate the strengths of current research and allow us to highlight the novelty of research proposed in the remaining chapters. Home-based healthcare and profiling wellbeing are target applications of our work. To demonstrate the use of sensing for such applications we demonstrate the detection of disability from profiling of activities of daily living, in a simulated home environment.

3.2 Activity Detection and Classification

In this section, we will examine in depth some of the components of the machine learning pipeline. The techniques used will be briefly reviewed, with references provided for further reading. We will also describe our sensing devices, the features extracted from the raw sensor, and the subsequent learning algorithms applied to the sensor data.

3.2.1 Sensing Hardware

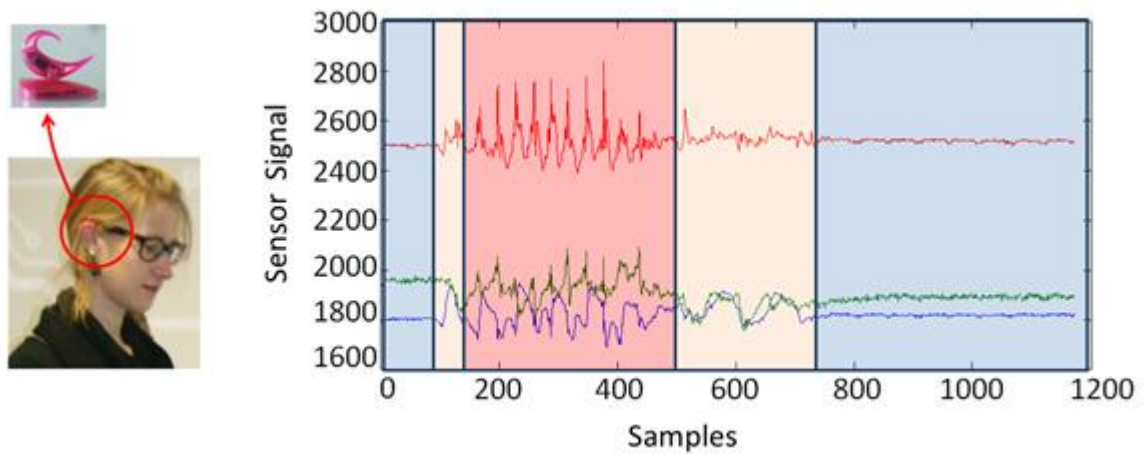


Figure 3.2 Participant wearing e-AR sensor (circled in red and shown separately), with raw sensor data showing the participant sitting, transitioning, walking to TV, transitioning to sitting again. Regions for sitting, transitioning and walking are coloured on the data plot as blue, brown and red respectively.

The main device used for data collection in our experiments is the e-AR sensor [4], which is a three-axis accelerometer built on the Body Sensor Network (BSN) sensing platform. The typical data rate for experiments in this research is 50 Hz. The e-AR sensor is inspired from the human body's natural mechanisms for posture and balance control. Evolution has determined the location of the ears as very well placed for feedback to the brain on movement. Yang et al. [1] describe the propagation of the shock wave along the spine, which is the acceleration picked up

by the accelerometer. Furthermore, changes in orientation can also be found in the e-AR data, based on the mean of the sensor channels.

Figure 3.2 shows an example of sensor data where a participant is watching TV while sitting, gets up to change the channel, and sits back down. The moving to TV and changing channels is a complex activity, involving a combination of walking and manipulation of the TV's buttons.

The sensor signal also shows the transitional periods that are of interest but have received relatively little research focus in comparison to general activity recognition. One difficulty for automatic analysis of such brief activities is that it is difficult to label them accurately, even in laboratory settings. In the dataset considered in this chapter, for instance, the experiment did not label transitional activities. In Chapter 4, we propose a methodology for automatically segmenting transitional activities for analysis.

3.2.2 Feature Extraction

The signal is partitioned using the sliding window technique, with each window corresponding to two seconds of data, sliding every half a second. For each window, we extract commonly used features for studying activities and change in activities. The features extracted from each channel are summarised in Table 3.1. In addition to these, three cross-channel features are also extracted. The features can be broadly categorised as statistical and wavelet. Statistical features are widely used in activity recognition and have previously been used for general activity recognition [62, 99, 169] and also for the classification of particular activities such as gait [170]. Wavelet based features have also been used to study gait and changes in gait [97] in addition to general activity recognition [171, 172]. As we will demonstrate visually, change in activity induces corresponding changes in each feature. These changes can sometimes be correlated, which means we do not gain information by including them in our analysis. It is difficult to predict beforehand which features will be effective for the classification task. In this chapter we consider simple approaches for reducing the dimensionality. In Chapter 4, an alternative approach will be explored, with the goal of uncovering parameters associated with human motion that underlie change in features.

Feature Number	Channel Feature	Feature Type
1	Energy	Statistical
2	Mean	Statistical
3	Variance	Statistical
4	Kurtosis	Statistical
5	Skewness	Statistical
6 - 13	Mean and Standard Deviation of wavelet coefficients, level 1-4	Wavelet

Table 3.1 Features extracted from e-AR sensor signal.

The first statistical feature is signal energy. Given a window x comprised of n samples, the energy E can be computed as:

$$E = \sum_{i=1}^n |x_i|^2 \quad (3.1)$$

The mean μ of the window gives its average value, while the variance σ^2 gives the average deviation from the mean. The sum of variances of the accelerometer signal has been used to characterise recovery from surgery in [115].

$$\mu = \frac{1}{N} \sum_{i=1}^n x_i \quad (3.2)$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2 \quad (3.3)$$

Skewness and Kurtosis are higher-order moments that can be used to characterise the asymmetry and shape of the signal relative to the normal distribution, and have been used for analysis of gait [173]. Skewness and kurtosis for a window are defined as:

$$skewness = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2\right)^{3/2}} \quad (3.4)$$

$$kurtosis = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2\right)^2} \quad (3.5)$$

Figure 3.3 shows the statistical features extracted from the data shown in Figure 3.2. It is obvious that not all of the features are informative for a given activity. For instance, the signal mean is virtually flat for each channel, which can be understood as there being no baseline changes to the channel data, suggesting that orientation did not change. In contrast, both the

signal energy and the variance peak when the participant starts walking. This suggests that these features will be most useful for discriminating between these two activities.

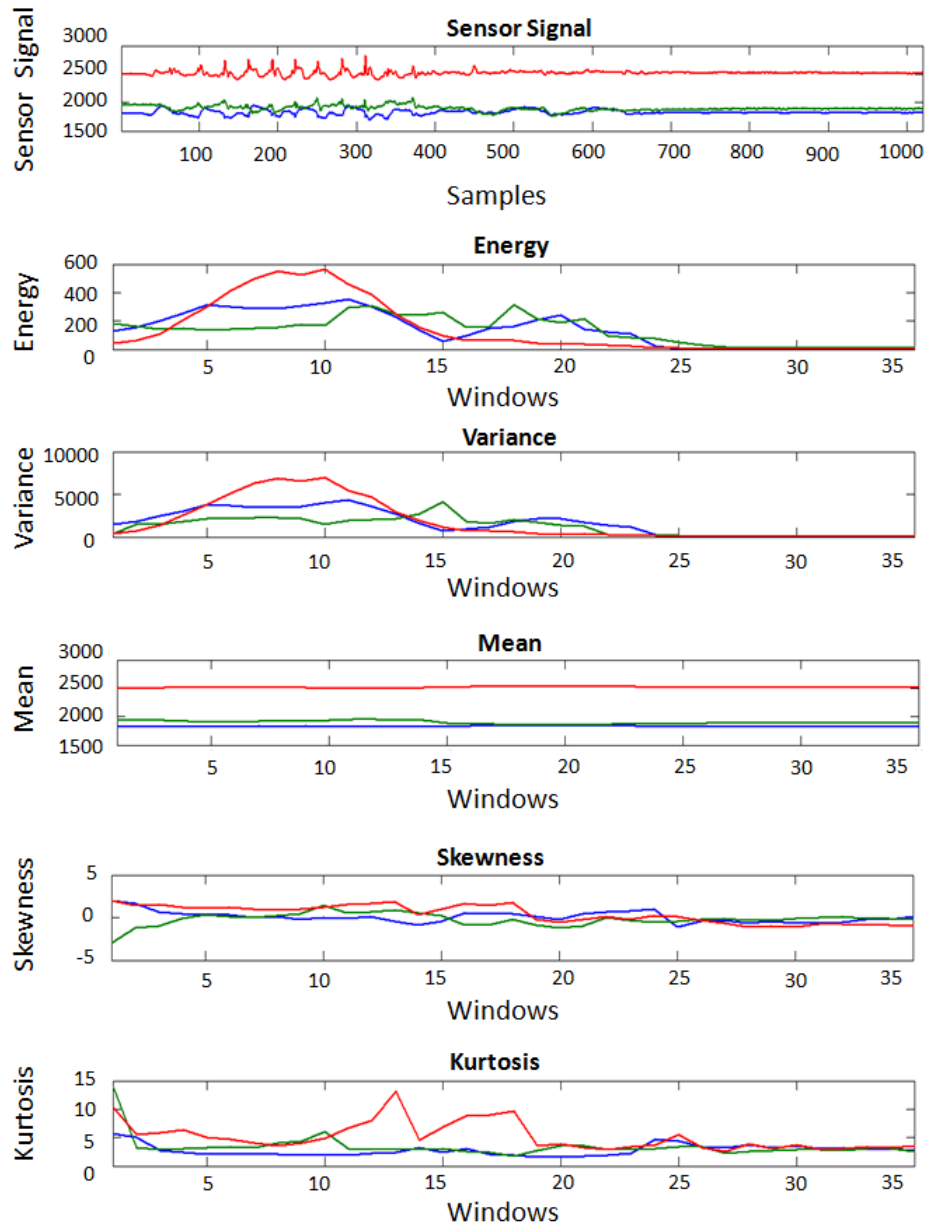


Figure 3.3 Waveform statistics extracted from accelerometer sensor data shown in Figure 3.2. Energy and variance are strongly indicative of activity intensity.

The second type of features extracted from the signal is of wavelet features. Wavelets extend FFT by incorporating frequency and time information through a multi-resolution data structure.

The signal is decomposed using a wavelet defined by a ‘mother wavelet’ and a scale. The Discrete Wavelet Transform (DWT) is defined by the following equation:

$$W(s, i) = \sum_s \sum_i x_i 2^{-s/2} \psi(2^{-s} n - i) \quad (3.6)$$

where s represents the scale, i represents time, and ψ refers to the basis function defining the wavelet, which is called the mother function. Appendix B provides further background on wavelets.

We compute the Haar wavelet coefficients across four time scales, as proposed by Atallah et al. [97] using a moving window. The coefficients are summarised using the mean and standard deviation of the wavelet coefficients. Therefore for each channel we have eight wavelet features. Figure 3.4 shows the wavelet features computed for the sensor signals introduced in Figure 3.3. It is evident that the walking activity is separated from the sitting activities.

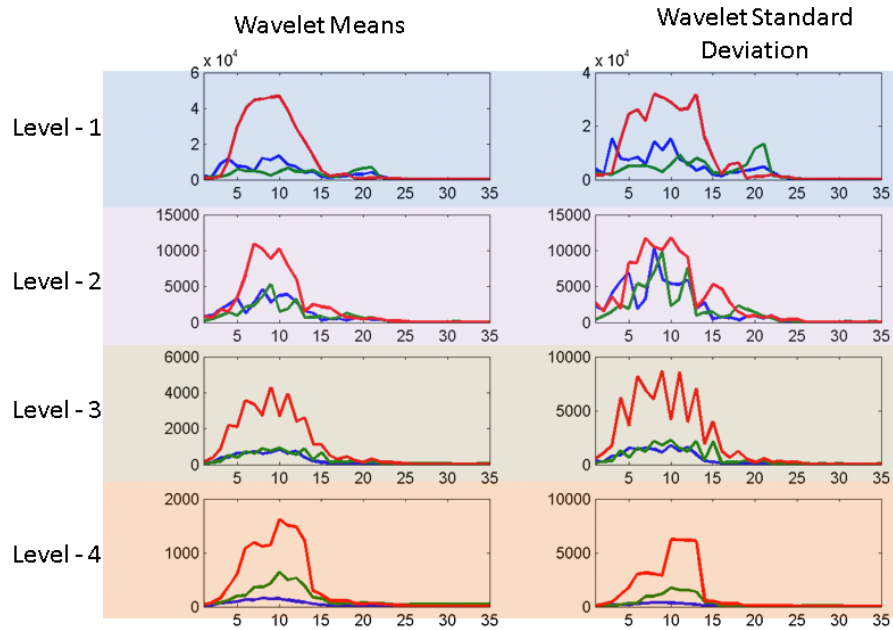


Figure 3.4 Features extracted from wavelet analysis. X and Y-axes plot window numbers and feature values respectively. At each level the mean and standard deviation from the wavelet coefficients is extracted as the feature.

The remaining features represent cross-channel information. The accelerometer data is significantly correlated for particular activities. We represent this using the covariance between each pair of channels. Given N observation the covariance between channels x and y can be computed according to the equation

$$Cov(x, y) = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu(x))(y_i - \mu(y))$$

The covariance features computed for the three sensor channels are shown in Figure 3.5^(3,7). The covariance between channels 1 and 2 in 3.5 (a) suggests an association with the sitting down transition. On the other hand, none of the covariance features can be used to reliably detect the walking activity.

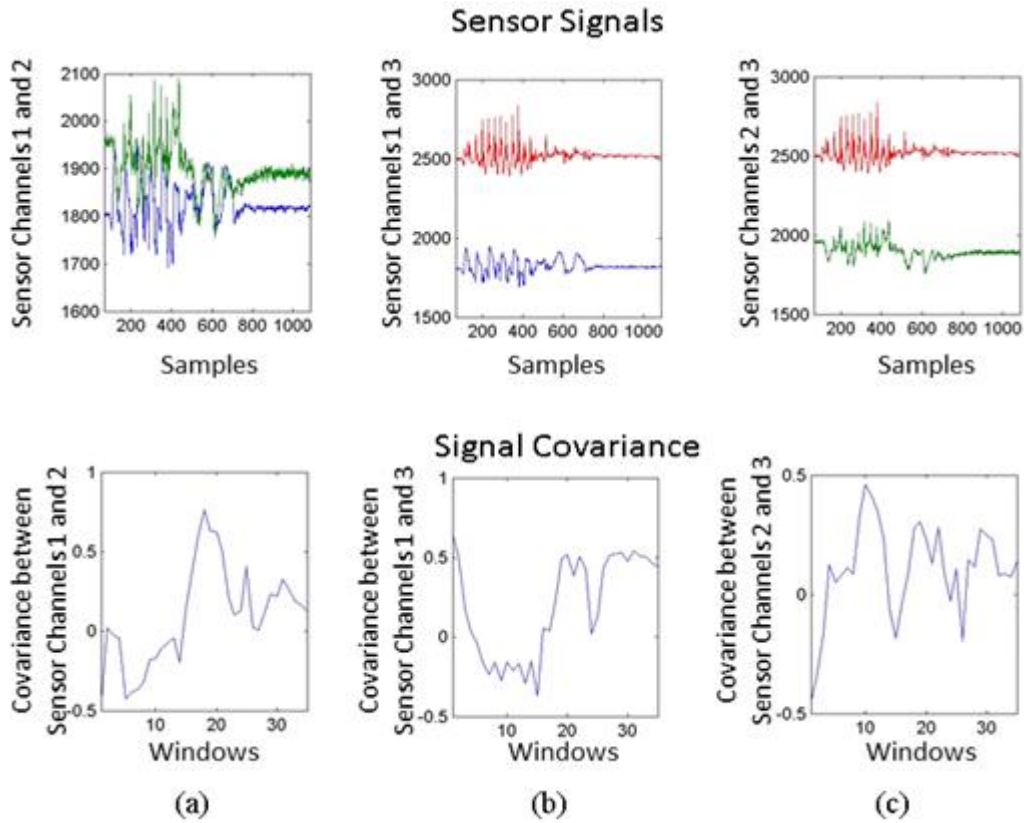


Figure 3.5 Pairs of sensor channels and their associated covariances, channels 1 and 2 (a), 1 and 3 (b) and 2 and 3 (c).

With the features discussed above extracted per-channel, the dimensionality of the data is considerable. In addition to increasing computational complexity, higher dimensional data requires more labelling in order to train reliable classifiers. There is a significant degree of redundancy in the features, as can be seen in Figure 3.4. The features for different levels of wavelet classification are highly correlated for the activity shown.

Therefore it is useful to reduce the feature space, either by selecting features most useful for classification, or transforming the data into a reduced dimensionality space that preserves the properties of the full feature set.

3.2.3 Dimensionality Reduction

A category of algorithms known as feature selection can be used to select the optimal signals for a given classification task. As the search space is comprised of every combination of features available, increasing dimensionality results in drastically large search spaces. However, algorithms can utilise heuristics to approximate an optimal solution. For instance margin-based algorithms seek to maximise a margin-function that measures the confidence of a classifier in

$$\mathcal{G} = \frac{1}{2} (\|x - nearmiss(x)\|_w - \|x - nearhit(x)\|_w) \quad (3.8)$$

where

$$\|z\|_w = \left(\sum_i w_i^2 z_i^2 \right)$$

making a decision. Let w be a weight vector over a feature set P , the margin of a point x can be written as

Near-hit and Near-miss points denote the nearest points to x with different and same labels respectively. A gradient based algorithm is used to find the weight vector w . The algorithm is of polynomial complexity. The Simba algorithm [98] has been used with success in previous work on e-AR sensor data analysis [99], and will be used in this work.

An alternative to reducing the feature space is to morph the data into a lower-dimensionality space representing important variations in the dataset. One of the most commonly used algorithms for this is the Principal Components Analysis (PCA), which through an eigen-decomposition of the covariance matrix represents data in the dimensions of maximum

variance. A related algorithm that has been shown to be mathematically equivalent to PCA is Multidimensional Scaling (MDS). MDS projects data into a low-dimensional space that preserves the distance between each pair of points. When this distance is expressed as Euclidean distance, MDS is equivalent to PCA. Formally, given a set of n objects with δ_{ij} the distance between points i and j , MDS finds points in transformed space with distance d_{ij} , minimising the following function, called the stress function

$$Stress = \left(\frac{\sum_{i,j} (d_{ij} - \delta_{ij})^2}{\sum_{i,j} \delta_{ij}^2} \right)^{\frac{1}{2}} \quad (3.9)$$

We will present results using the MDS algorithm, and compare its effectiveness against the feature selection approach.

3.2.4 Generating Class Labels

In machine learning, if class labels are available, we will train supervised learning based classifiers. Given participant variability, it is sometimes necessary to train per-participant classifiers, and per-activity classifiers. We present results for classifiers covering three broad areas of classification: inductive learning based decision trees, probabilistic inferencing based Naive Bayes algorithms and Artificial Neural Network based Multi-Layer Perceptrons. For a full review of these algorithms we refer the reader to [174, 175], a brief description follows:

- 1) **C4.5** –Decision tree algorithms construct a tree structure wherein the data is partitioned into sets, with inclusion criteria based on the value of attributes. The goal of the partition is to split into sets enriched in one class or the other. An example decision tree is shown in Figure 3.6 (a). A decision tree can be said to generalise the data if it is able to describe the data with a small number of decision nodes. The C4.5 algorithm [176] bases the partitioning criterion on the information theoretic concept of entropy: at each partition the partition is chosen to maximise the resulting information gain. The information gain for a set s on attribute A is computed as follows:

$$Gain(S, A) = E(S) - \sum_{A_i \in A} \left| S_{A_i} \right| * E(S_{A_i}) \quad (3.10)$$

where S_{A_i} denotes the subset of S with $A = A_i$, $\left| S_{A_i} \right|$ denotes the frequency of S_{A_i} in S and E denoting the entropy of the set. At every iteration the attribute maximising information gain is selected, resulting in a new branch of the tree, with the same procedure applied to child sets falling either side of the decision boundary. Decision tree classifiers are especially good for handling data with redundant or irrelevant features. At each decision point, the attribute providing the most information gain is used to partition the data. It can be seen as including a built-in feature selection mechanism.

- 2) **Naive Bayes** – Naive Bayes [177] is a simple probabilistic classifier that can be trained very efficiently by exploiting strong independence assumptions between features. In practice these have been observed to outperform more complex classifiers for many problems [178]. Probabilistic classifiers seek to learn a conditional probability model $P(C | A_1, A_2, \dots, A_m)$ of the class label C , over attributes $A_1, A_2, \dots, A_m$. By Bayes Rule the probability can be written as

$$P(C | A_1, A_2, \dots, A_m) = \frac{P(C)P(A_1, A_2, \dots, A_m | C)}{P(A_1, A_2, \dots, A_m)} \quad (3.11)$$

The complexity of Bayesian classification lies in computing $P(A_1, A_2, \dots, A_m | C)$. The conditional independence assumption of Naive Bayes is that $P(A_i | C, A_j) = P(A_i | C)$ for any i and j . Given this independence assumption $P(A_1, A_2, \dots, A_m | C)$ is simplified as shown below:

$$\begin{aligned} & P(A_1, A_2, \dots, A_m | C) \\ &= P(A_1 | C)P(A_2, A_3, \dots, A_m | C, A_1) \\ &= P(A_1 | C)P(A_2, A_3, \dots, A_m | C) \quad [\text{Given Conditional Independence Assumption}] \\ &= P(A_1 | C)P(A_2 | C)P(A_3, \dots, A_m | C, A_2) \\ &= P(A_1 | C)P(A_2 | C)P(A_3, \dots, A_m | C) \\ & \quad [\text{by repeating this procedure}] \\ &= P(A_1 | C)P(A_2 | C) \dots P(A_m | C) \\ &= \prod_{i=1}^m P(A_i | C) \end{aligned} \quad (3.12)$$

Therefore $P(C | A_1, A_2, \dots, A_m)$ can be computed as

$$P(C | A_1, A_2, \dots, A_m) = \frac{1}{Z} P(C) \prod_{i=1}^m P(A_i | C) \quad (3.13)$$

where z is a scaling constant. Figure 3.6 (b) shows an example of a Naive Bayes model.

- 3) **Multilayer Perceptron** – Artificial Neural Networks (ANN) are computational models inspired from biological neural processes, where data flows through a connected network of distributed processing nodes. Learning algorithms train the network learn weights for the connections in order to meet the classification goal. Multilayer perceptron [175] is a feedforward artificial neural network with an input layer of neurons corresponding to the input data, one or more hidden layers, and an output layer corresponding to the classification or regression task. Hidden layers allow the neural network to learn non-linear decision boundaries. An example of this type of network is shown in Figure 3.6(c). The backpropagation algorithm is used to train the perceptron. A large number of parameters can influence the performance of the network, ranging from the number of hidden layers to the choice of transfer functions used in the neurons.

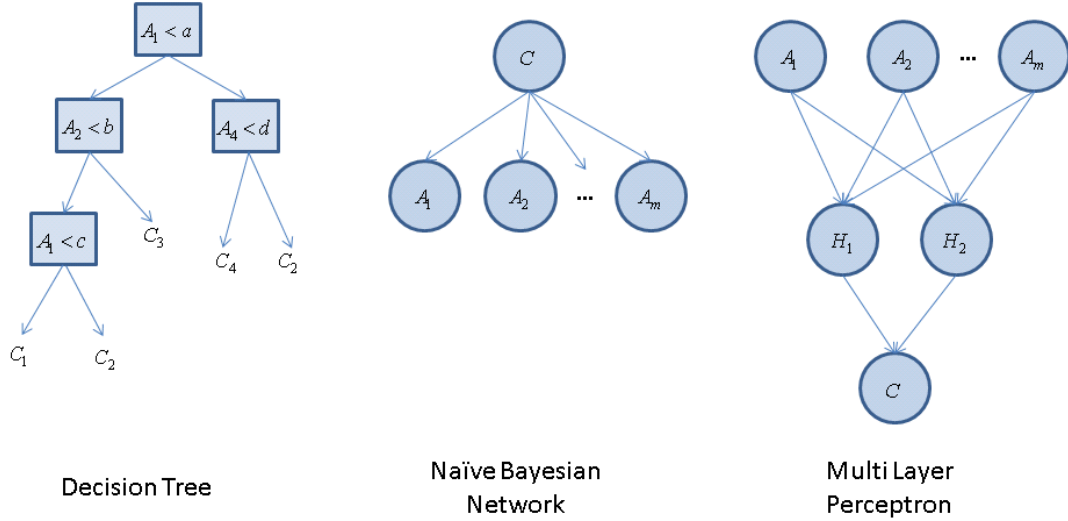


Figure 3.6 Examples of classification algorithms. The decision tree (a) partitions the dataset at each node based on attribute values. The Naive Bayesian (b) model shows the conditional independence assumption graphically, where the arrows indicate dependence. The Multi-Layer Perceptron (c) shown here has a single hidden layer has an activation layer corresponding to the attributes, a single hidden layer with two neurons, and an output layer giving the classification.

3.3 Laboratory Experiment Setup for Activity Classification

Two comparative simulated datasets of normal versus impaired mobility were acquired for this study. 16 participants (11 male, 5 female) performed a circuit comprising of household activities. Their average age was 27 years with a standard deviation of 1.86 years. In the first dataset participants performed the circuit naturally. For the second dataset lower limb mobility was mildly impaired using Tubigrip, and truncal mobility was impaired with an abdominal brace system (Orthomerica Air Back™ Spinal Support System). Participants wore an e-AR sensor on their right ear, secured using a headband. Figure 3.7 shows a participant wearing the impairment simulation system. Table 3.2 lists the activities analysed in this chapter, along with the instructions given to the participant.

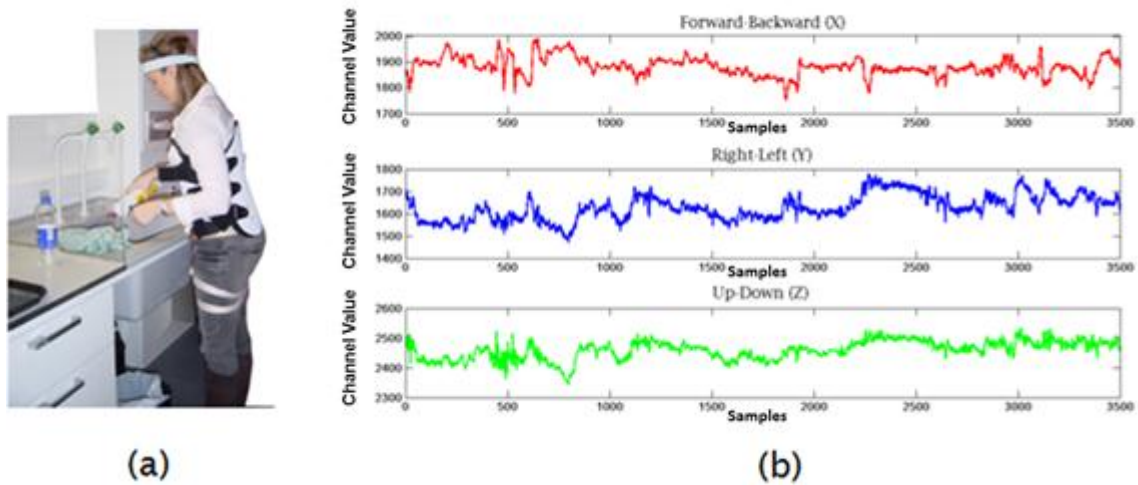


Figure 3.7 A participant performing the washing up activity while wearing the braces system designed to impair lower limb and truncal movement (a), and the corresponding sensor data (b).

Activity Number	Activity Name	Instructions Given to Participant
1	Walking	Walk to dining table holding a tray with food
2	Eating	Eat a banana sitting at dining table
3	Drinking	Drink tea, taking 5 sips
4	Washing up	Throw rubbish in bin and wash dishes
5	Operating Television	Stand up, go to television, change channel, sit down
6	Use Stairs	Climb a staircase with ten steps

Table 3.2 Activities performed by participants in the impairment detection experiment. The circuit was repeated with and without braces.

3.4 Results

Table 3.3 lists five features selected by Simba for the detection of impairment based on the specified activity. The algorithm was run for 500 iterations. The higher moments of kurtosis and skewness are selected most often: the skewness of atleast one channel is selected for every activity. In contrast, the wavelet features tend to be excluded, with only one feature selected for the six activities.

Activity Number	Features Selected
1	Kurtosis (2), Covariance(1,2), Skewness (1), Covariance(2,3), Kurtosis (1)
2	Kurtosis (1), Kurtosis (2), Skewness(1), Skewness (2), Mean(3)
3	Covariance(1,2), Covariance(1,3), Skewness (1), Kurtosis (1), Mean(3)
4	Kurtosis (2), Skewness (2), Mean (1), Skewness (1), Mean (3)
5	Variance (2), Mean (1), Skewness (3), Skewness(1), Mean(3)
6	Variance (1), Wavelet L1 Mean (1), Kurtosis (3), Skewness (1), Mean (3)

Table 3.3 Features selected by the Simba algorithm for classification of impairment. The channels from which the feature is extracted is specified in brackets. For example Kurtosis (2) refers to the kurtosis of channel 2, while Covariance(2,3) refers to covariance between channels 2 and 3.

Figure 3.8 shows Simba and PCA reduced features for each activity. Visually, neither algorithm performs consistently better than the other, although the impairment labelled data may be better visible for a particular instance in the table; for example PCA offers better separation than Simba for the walking activity, while being more mixed for the washing up activity. On the whole it is clear that in three dimensions neither approach is able to separate the classes. One reason for this is that participants adapted differently to the impairment. When participants are plotted separately, the normal and impaired data points cluster more distinctly for a number of activities, as shown in Figure 3.9. While training a single classifier for all participants enables deployment without the need for retraining for new participants, customizing the algorithms for each participant is likely to improve accuracy. We will demonstrate this quantitatively in the next section.

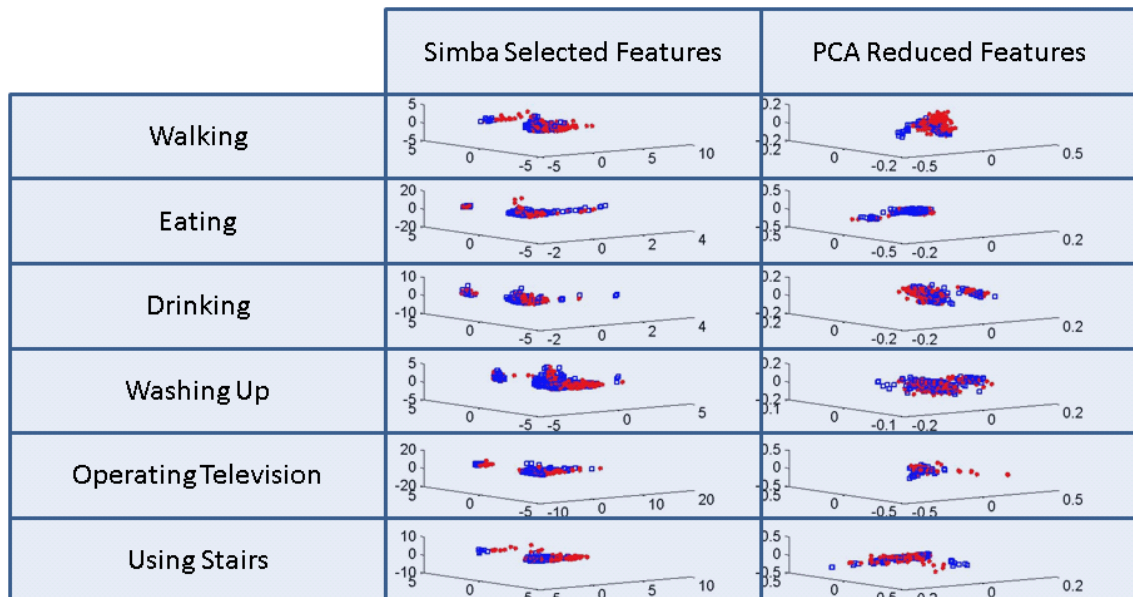


Figure 3.8 Impairment detection data in first three dimensions of reduced feature space for all participants together. Red indicates simulated impairment, while blue indicates normal data.

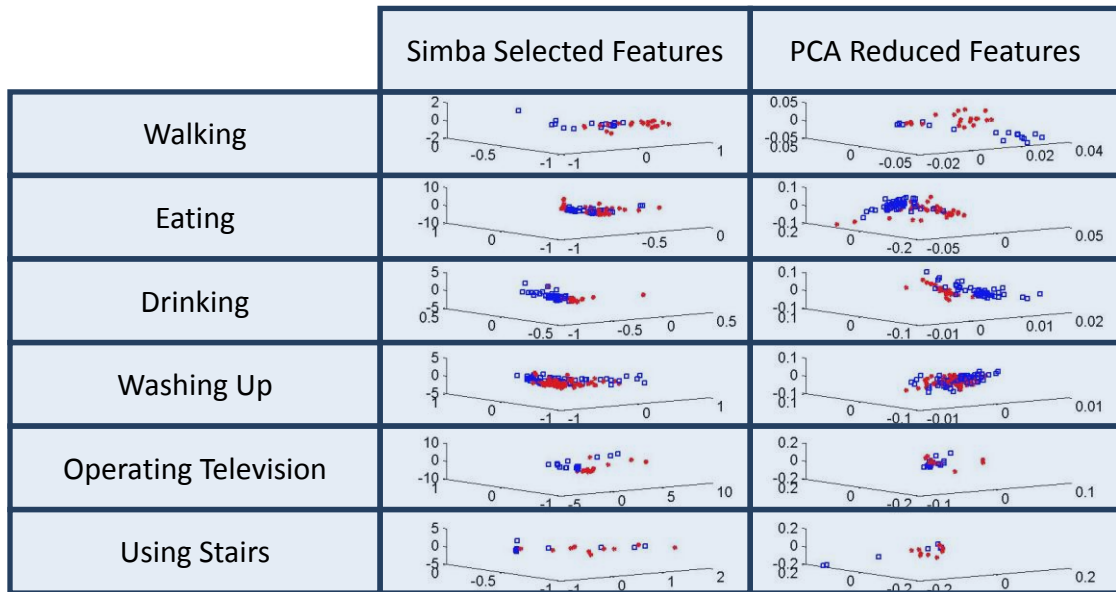
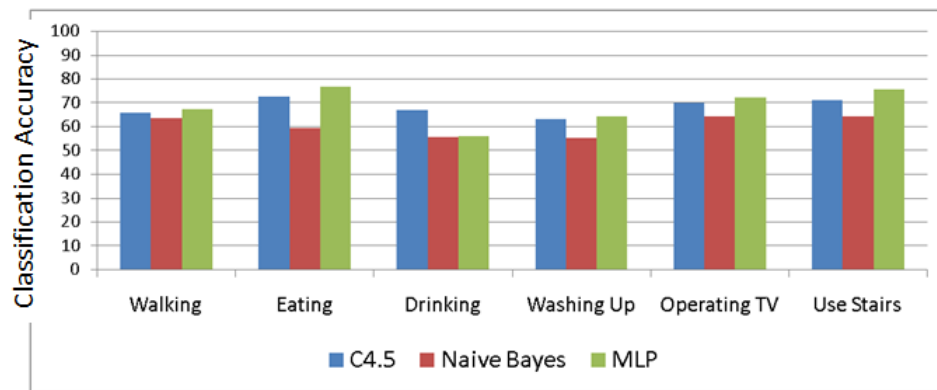


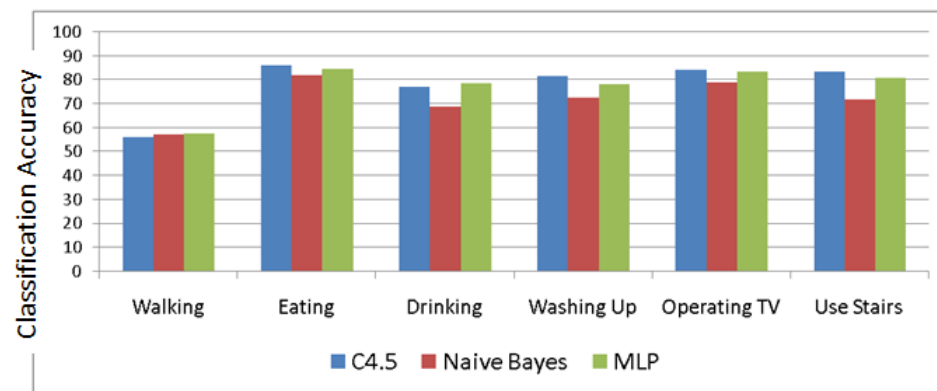
Figure 3.9 Impairment detection data in first three dimensions of reduced feature space for a single participant. Red indicates simulated impairment, while blue indicates normal data. Impaired and normal data points cluster more distinctly compared to the full feature space shown in Figure 3.9.

3.4.1 Classification of Impairment

C4.5, Naive Bayes, and MLP classifiers were separately used to detect impairment in performing each activity, with ten folds cross validation. Results for PCA and Simba reduced features are shown in Figures 3.10 (a) and (b) respectively. In general using Simba features results in an average 9.9 % increase in classification accuracy, which is a significant improvement. With the exception of the walking activity, where there is a noticeable decrease in performance relative to the PCA derived features. For both PCA and Simba features, C4.5 and MLP classifiers have similar performance, and generally outperform Naive Bayes classifiers.



(a)



(b)

Figure 3.10 Classification accuracy for impairment detection using PCA reduced features (a) and Simba selected features (b). Using Simba features leads to an average 10% increase in accuracy.

3.4.2 Impact of Dimensionality

To investigate whether increasing the number of features selected by Simba impacts the classification performance a C4.5 classifier was trained while varying the number of features. The results are shown in Figure 3.11. In general there is a small increase in classification accuracy, although as in the case of the ‘Use Stairs’ activity, adding features can result in a decrease in performance. There is a significant increase in the classification accuracy for the walking activity. This is to be expected, as the initial accuracy was poor. This is an instance

where information from more features is needed to correctly classify the data, which is why PCA performed better than Simba for this activity.

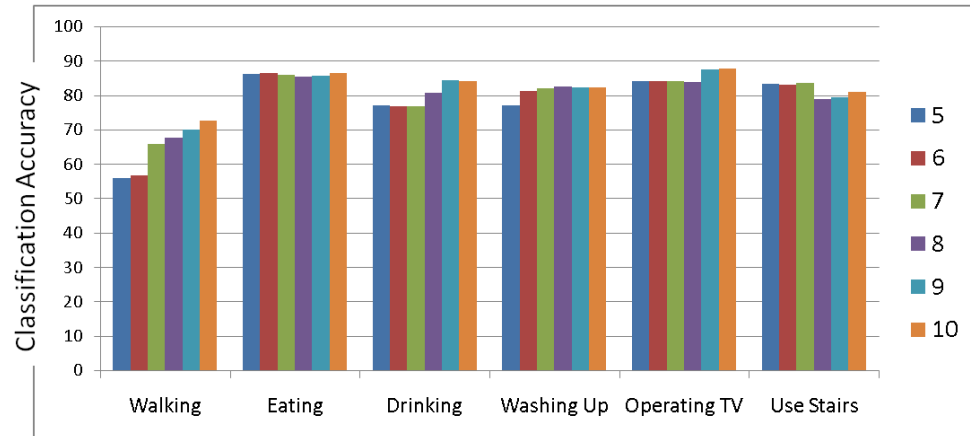


Figure 3.11 Classification accuracy of C4.5 classifier with increasing number of features selected using the Simba algorithm. In general there is a small gain in classification accuracy with increasing number of features.

3.5 Conclusions

In this chapter, we have applied components of a machine learning pipeline to sensor data from an impairment detection study. The analysis focused on activities of daily living performed in a simulated home environment. The results show that it is possible to detect impairment using off the shelf classification algorithms. There is also significant value in reducing the number of features. Not only may this improve classification accuracy, it is also beneficial from a systems perspective to compute, transmit and store a smaller number of features.

While analysis of ADL for impairment is promising, there are a number of directions in which research can be further developed. Changes in activity are usually marked as instantaneous in usual data collection protocols. For example, in the dataset considered in this chapter, the participant was asked to get up from sitting and start walking. Whereas this was labelled as a single activity, this is in fact comprised of two activities: the walking activity and a transition between sitting and standing. Not only is it more correct to consider this transition separate from

walking, the nature of the transition can be medically relevant. Labelling such short lived transitions accurately as it occurs is difficult. While it is possible to retrospectively label the data, the process is prone to errors in judgement, and requires an expert to gauge where a transition is occurring in the sensor signal. In Chapter 4 we propose a methodology for the automatic detection and analysis of such transitional activities. The medical relevance of transitional activities will be demonstrated by tracking the change in the performance of specific transitions by participants recovering from a knee replacement surgery.

A further direction of research lies in progressing from analysis of activity at small scales to the mining of historical activity data. This is the type of data that can be expected to result from deployments of pervasive sensing systems in real-world scenarios. As there is likely to be little annotation available, an exploratory machine learning strategy will need to be employed. In Chapter 5, we propose a data-mining based technique for representing an individual's daily routine from a database of activity. The proposed analysis algorithms are adapted to non-specialised sensors on smartphones in Chapter 6. Finally in Chapter 7 quantitative analysis of routine is demonstrated.

Chapter 4

Analysing Transitional Activities

4.1 Introduction

ALTHOUGH the use of wearable sensors for activity classification has been explored extensively in recent years, as mentioned in the last chapter that an area that has received relatively little attention is transitions between commonly studied activities. For example, many elderly people face difficulties in rising from a chair [179]. As functional mobility is impaired with age, disease, disability or injury, transitional movements such as these can become more laboured and distinct. The ability to perform transitions with ease has been suggested as a target for rehabilitation [180, 181] and as an indicator for musculoskeletal strength and motion coordination [131]. Not only can they be used to indicate restrictive and impairing quality of life, they can also be used to infer the status of health and the onset of certain age related musculoskeletal and neuro-degenerative diseases. For example, transitions from sitting to standing have been shown to be indicative of the likelihood of falls [130, 182]. They have also been studied in association with stroke [183, 184], neuromuscular conditions [133], and found to be an indicator of difficulty in movement due to obesity [185].

Thus far, clinical and rehabilitation studies typically focus on Sit-to-Stand transitions conducted in laboratory and hospital settings. These rely on expensive, specialised equipment often with human observations included in the assessment. A goal of pervasive sensing research is to translate such research into home environments, reducing both the cost of sensors and human intervention. In previous work, Allin and Mihailidis [136, 186] developed such a video-based

system using multiple web-cameras to study the kinematics of the Sit-to-Stand transition in ‘natural’ environments. Other systems have also been developed using generic wearable sensors [139, 140, 187]. These studies typically involve multiple sensors, and the analysis focuses on the detection of specific transitions.

In this chapter, we will extend previous research by focusing on transitional activity recognition and present a framework for both detection and analysis of transitional activities. The framework uses a single ear-worn Body Sensor Network (BSN) [115] node, and is applied to tracking post-operative recovery and impairment detection. Figure 4.1 shows data from the BSN node as a participant performs the transition from standing to sitting.

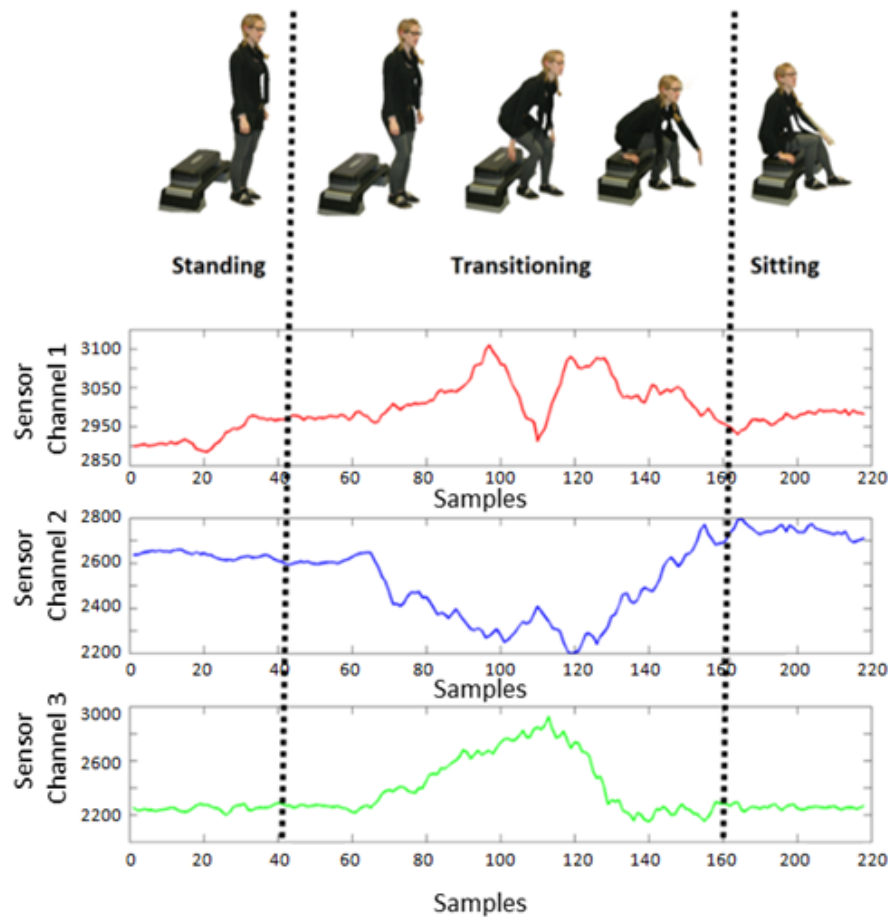


Figure 4.1BSN sensor data from a participant performing the Stand-to-Sit transition. Data morphs from standing to sitting in a distinctive pattern.

It is important to note that transitions between activities in natural behaviour may not be sharp (or instantaneous) with distinctive boundaries. Instead, one activity may morph into another. Typically, tools developed for experimental data collection allow the user to place markers in the data stream as a means of ground truth during the course of the experiment. For prolonged transitions, the marking of the boundary can be subjective and error-prone. One of the goals of this study is therefore to segment activities and transitions automatically, thus reducing subjective error and improving subsequent analysis.

The purpose of this chapter is to describe a semi-supervised approach for flexible yet consistent transitional activity recognition and analysis. The analysis will be applied to the problem of impairment detection and postoperative recovery. Unsupervised and semi-supervised techniques are attractive in that the structure in the data can be effectively utilised, decreasing the reliance on explicit labels. The main contributions of this chapter can be summarised as follows:

- 1) Transitional activity detection using a single ear-worn sensor from un-segmented data.
- 2) The development of a framework for categorisation and analysis of transitions in manifold space.
- 3) Application of framework to impairment and post-operative recovery studies.

4.2 Analysis Framework for Transitional Activity Recognition

Pervasive sensing environments generate a significant volume of data. As described in Section 3.2, a large number of features can be computed from the raw sensor signal. It may not be possible to manually annotate such data, particularly after deployment in homes. With the limited availability of class labels, it is important to maximise what can be learnt from the structure of the data itself, and to perform effective classification it is important to map data into dimensions that reflect its intrinsic variations. In this section we develop such a framework for transitional activities.

Technically, the proposed framework consists of the following main steps:

- 1) Feature Extraction: Features are extracted from fixed size windows of data.

- 2) Transition Detection – Transitions are detected using a spectral bisection based algorithm
- 3) Dimensionality Reduction: Transitional activities are embedded in a lower-dimensional space reflecting each activity's intrinsic structure.
- 4) Transition Categorisation and Analysis: Utilizing supervised learning methods transitions found in step 3 are placed into categories and analysed for specific healthcare applications.

Until Step 4 the analysis performed is without the use of class labels. Our analysis is enabled by the modelling of activity using the geometric concept of manifolds, both for detecting transitions in activity and reducing dimensionality. In the following manifolds are briefly introduced, followed by their use for transition detection and dimensionality reduction.

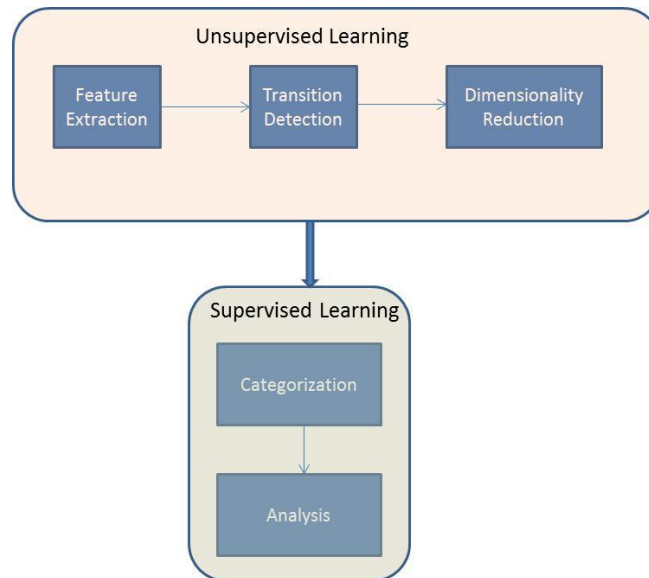


Figure 4.2 Overview of Transition detection and analysis methodology, with unsupervised learning based preprocessing before classification and labelling of transitions is performed.

4.3 Manifold modelling of Transitional Activity

Structure in sensor signals recording human activity may result from constraints on how activities can be performed by participants. Such structures can be modelled using the geometric

concept of manifolds. Points in feature space are mapped to a manifold. The location of each point on this manifold expresses its relationship to the dataset as a whole. Regions of the manifold will be associated with types of activity. Transitions can be detected by segmenting out strongly clustered regions.

The coordinate system of the manifold can be expressed in considerably fewer dimensions than the feature space, depending on the complexity of the dataset. This is referred to as the *intrinsic dimensionality* of the data. Manifold embedding algorithms, such as Isomap [188], map high-dimensional data into this low-dimensional manifold space. We use a manifold representation in order to uncover lower-dimensional, intrinsic structure from features extracted from sensor data, with the coordinates of this structure representing the dimensions of transitional activity.

4.3.1 Representing Activity with Manifolds

Manifolds are topological spaces that can express complex geometrical structure with well-understood mathematical properties. The space near each point is locally Euclidean, referred to as its neighbourhood. The global structure of manifolds comprised of such points can be complicated. For instance, Osinga and Krauskopf [189] represent the structure of chaotic systems using a manifold. Using a manifold model, they were able to construct a three dimensional object useful for understanding and explaining the dynamics of the Lorenz equations modelling weather systems. Manifold representations are useful where complex data is generated by a small number of parameters. For instance, while Lorenz Equations generate chaotic systems, the equations are described by three parameters. The Euclidean neighbourhoods around each point are in the space of these parameters, which comprise the dimensionality of the manifold. A fuller review of manifolds is provided in Appendix A.

Euclidean neighbourhoods in sensor data can be modelled by selecting for each point either a fixed number of points closest to it, or by selecting points within a fixed radius. A distance matrix is computed from the points in feature space. As it is difficult to specify a neighbourhood radius beforehand, the k nearest neighbours around each point are selected. This is a parameter in our framework that needs to be selected and validated empirically.

Points lying close together on an activity manifold represent data-points generated from similar sensor signals. A strongly connected region of the manifold may represent signals characteristic of a particular type of activity. Activity transitions are detected from the movement of participants from one region of the manifold to another. Figure 4.3 (a) shows signals of an accelerometer worn by a participant approaching a bed and lying down. The manifold corresponding to this is shown in Figure 4.3 (b), with distinct regions for each activity, and a transitional region connecting them. These regions of the manifold can be found using the algorithm described in the next section.

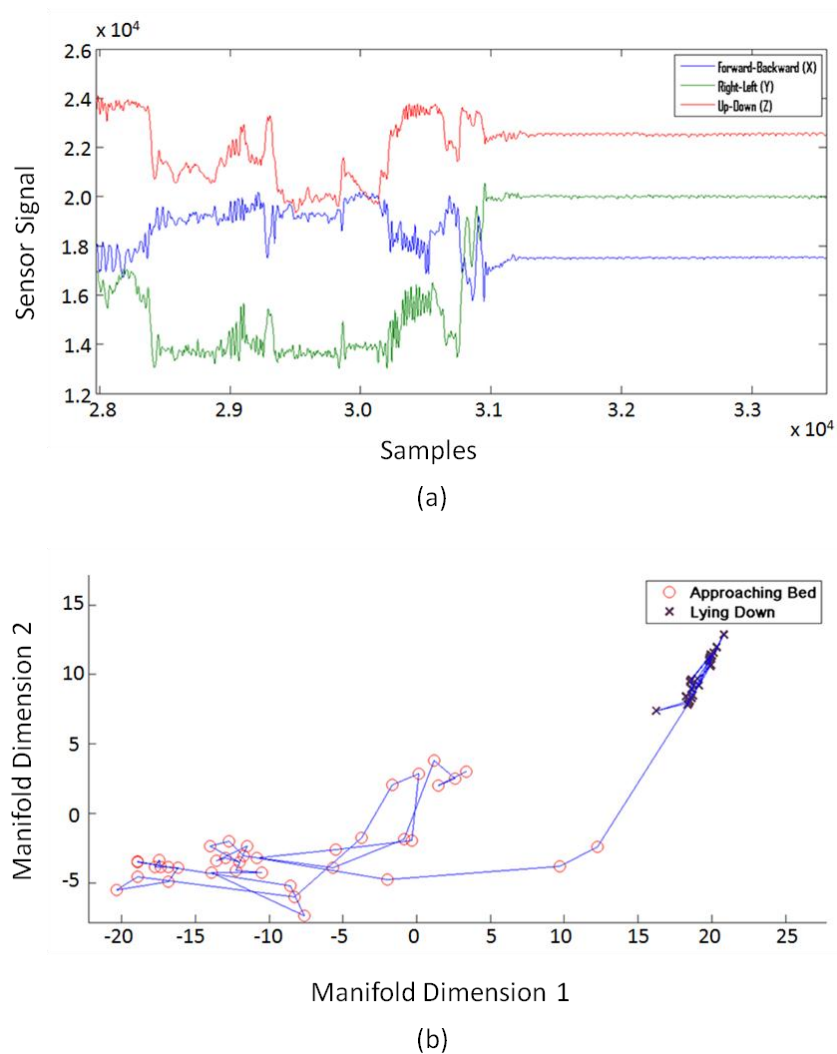


Figure 4.3 Participant approaches bed and lies down: sensor signal (a) and in a manifold visualisation (b). The two activities cluster distinctly in manifold space.

4.4 Detecting Transitional Activity

The neighbourhoods of a manifold can be represented as a graph, where data points correspond to vertices, edges correspond to a neighbourhood relationship, and edge weights correspond to distances. Strongly connected regions of the manifold can be found by partitioning this graph.

Graph Partitioning is the problem of dividing a graph into k parts, such that the parts are close to having the same number of vertices, and there are minimum connections between the pieces. Given a graph with vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$, and an edge set E with w specifying the weight for each edge, find k disjoint subsets of V whose union equals V , minimising the following cost function produces the optimal partition.

$$Cost = \sum_{V_a \neq V_b} \sum_{\substack{i \in V_a, j \in V_b \\ (i, j) \in E}} w_{ij} \quad (4.1)$$

Graph Partitioning is an NP-Complete problem. Heuristic techniques exist for efficiently computing reasonable partitions however, as described below.

Spectral Graph Theory[190] analyses graphs in terms of their eigenvalues and eigenvectors. Given the adjacency matrix A we can compute the degree matrix D , which is a diagonal matrix such that an entry d_{ii} corresponds to the sum of edge weights incident on vertex i . The graph Laplacian is defined as a matrix with entries as per the definition:

$$L_{ij} = \begin{cases} d_{ii} & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

L can be computed as $L = D - A$. The spectral properties of L are well-studied [190-192]. Let $\lambda_1, \lambda_2, \dots, \lambda_n$ be ordered eigenvalues of L . Let the corresponding eigenvectors be $y_1, y_2, \dots, y_n$. For a fully connected graph the first eigenvalue λ_1 is zero, with the corresponding eigenvector equal to a vector of ones. The second eigenvalue λ_2 , and corresponding eigenvector y_2 have been studied by Fiedler [191] and have found to have properties useful for studying the graph's connectivity. y_2 is sometimes referred to as the Fiedler vector. Fiedler proved that sets V_1 and V_2 defined as

$$\begin{aligned} V_1 &= \{v \in V, y_2(v) > 0\} \\ V_2 &= \{v \in V, y_2(v) < 0\} \end{aligned} \quad (4.3)$$

are both connected subgraphs. Therefore using the Fiedler vector we can partition the graph into components that will be connected themselves. Furthermore, it is however possible to demonstrate that these components are also good approximations of the optimal bisection.

Assuming we have two partitions p_1 and p_2 let p be a vector such that

$$p_i = \begin{cases} +1 & \text{if } v_i \in V_1 \\ -1 & \text{if } v_i \in V_2 \end{cases} \quad (4.4)$$

For two partitions, equation 4.1 can be minimised by minimizing the size of cut-set, defined as

$$C = \frac{1}{4} \sum_{(i,j) \in E} (p_i - p_j)^2 w_{ij} \quad (4.5)$$

It can be shown that this is equal to $p^T L p$ by the following derivation.

$$\begin{aligned} & \sum_{(i,j) \in E} (p_i - p_j)^2 w_{ij} \\ &= \sum_{(i,j) \in E} (p_i^2 + p_j^2 - 2 p_i p_j) w_{ij} \\ &= \sum_i p_i^2 \sum_j w_{ij} + \sum_j p_j^2 \sum_i w_{ij} - 2 * \sum_{(i,j) \in E} p_i p_j w_{ij} \end{aligned}$$

Given that $\sum_j w_{ij} = d_{ii}$

$$= \sum_i p_i^2 d_{ii} + \sum_j p_j^2 d_{jj} - 2 * \sum_{(i,j) \in E} p_i p_j w_{ij}$$

By the sum of components and the definition of L in equation 2 this is equal to $p^T L p$. Therefore

$$p^T L p = \sum_{(i,j) \in E} (p_i - p_j)^2 w_{ij} \quad (4.6)$$

C is minimised by finding the eigenvector associated with the minimum eigenvalue of L . The trivial solution, associated with λ_1 places all the vertices into the same partition, as the associated vector is a vector of ones. The smallest non-zero eigenvalue is λ_2 and the associated Fiedler vector y_2 minimises C .

It remains to discretise y_2 to compute p . This can be done by thresholding on a cut-point. This is referred to as spectral bisection. Labels are assigned based on a threshold value. This threshold can be zero as suggested by Fiedler himself [191], which is guaranteed to generate partitions that are themselves connected graphs. Other choices exist for the threshold, for instance for load balancing applications typically the median value is used, to get balanced partitions. Our approach is to select the cut-point automatically using clustering, as in [192]. This leads to strongly connected components selected as activities.

A bisection on an activity manifold partitions it into two ‘types’ of activity. A transition is specified as when a participant moves from activity in one partition to the other. To correct for over-segmentation and incorporate the idea of ‘principal activity’ as discussed in section 3.1, an activity segment must be of a minimum size. This analysis is performed recursively for activities of above the minimum size, detecting progressively smaller partitions. The manifold is recomputed at each level of recursion, ensuring that the graph has sufficient connectivity.

As published in [193] we allow the recursion to segment transitions from activity segments automatically. This however has the disadvantage of not explicitly specifying a segment as a transitional activity. It was shown by Belkin [194] that the Fiedler vector can also be used to map a manifold to a line with relative proximity corresponding to strong connectivity. Therefore the values of the Fiedler vector can be used to find intermediate sections between the transitions. If a participant transitions from an activity with a cluster center c_1 to an activity with a cluster center c_2 a membership function u can be computed as:

$$u_i = \frac{y_2(i) - c_1}{y_2(i) - c_2} \quad (4.7)$$

We refer to using this membership function to identify transitional segments as explicit labelling. The membership function can be seen as an adaptation of the cluster membership function used in Fuzzy C-Means clustering [195].

Algorithm 1 describes the overall methodology.

Algorithm 1: Detect Transitions

Input: Set X of elements indexed by sample number, distance matrix D , neighbourhood size k , cluster membership threshold t , minimum segment threshold s

Output: Set of transitional activity T and principal activities P

1. Compute graph G from D with k nearest neighbourhood function
2. Compute L and the Fiedler eigenvector y_2
3. Find two clusters on y_2 with function $C(x)$ specifying cluster membership of x
4. Compute a set of segments S such that for all $S_i \in S$:

$$S_i = \{x_a, x_b \in X, C(x_a) = C(x_b), b - a = 1\}$$

and

$$\|S_i\| > s$$

5. If transition labeling is explicit:
 - a. For each pair of consecutive segments S_i and S_j
 - i. Compute u . If x_t is the datapoint where cluster membership changes, the transitional activity T_{ij} is defined as

$$T_{ij} = \{x_a, x_b \in X, u_{x_a} < t, u_{x_b} < t, b - a = 1\}$$

and

$$x_t \in T_{ij}$$

$$\text{ii. } T = T \cup T_{ij}, P = P \cup (S_i - T_{ij}) \cup (S_j - T_{ij})$$

- b. Else $P = S$

6. For each $P_i \in P$ where $|P_i| > r$ *DetectTransitions*(P_i)

4.5 Using Manifolds to Capture Intrinsic Dimensionality

As discussed in Section 3.2.1, a large number of features can be extracted from just one channel of a single sensor. As the complexity of the sensors increases along with the number of sensors, the number of dimensions of data needed to be processed by a learning algorithm is very high. It is difficult to know apriori which features will be useful. To enable efficient classification

therefore, it is necessary to project the data into spaces with fewer dimensions using dimensionality reduction.

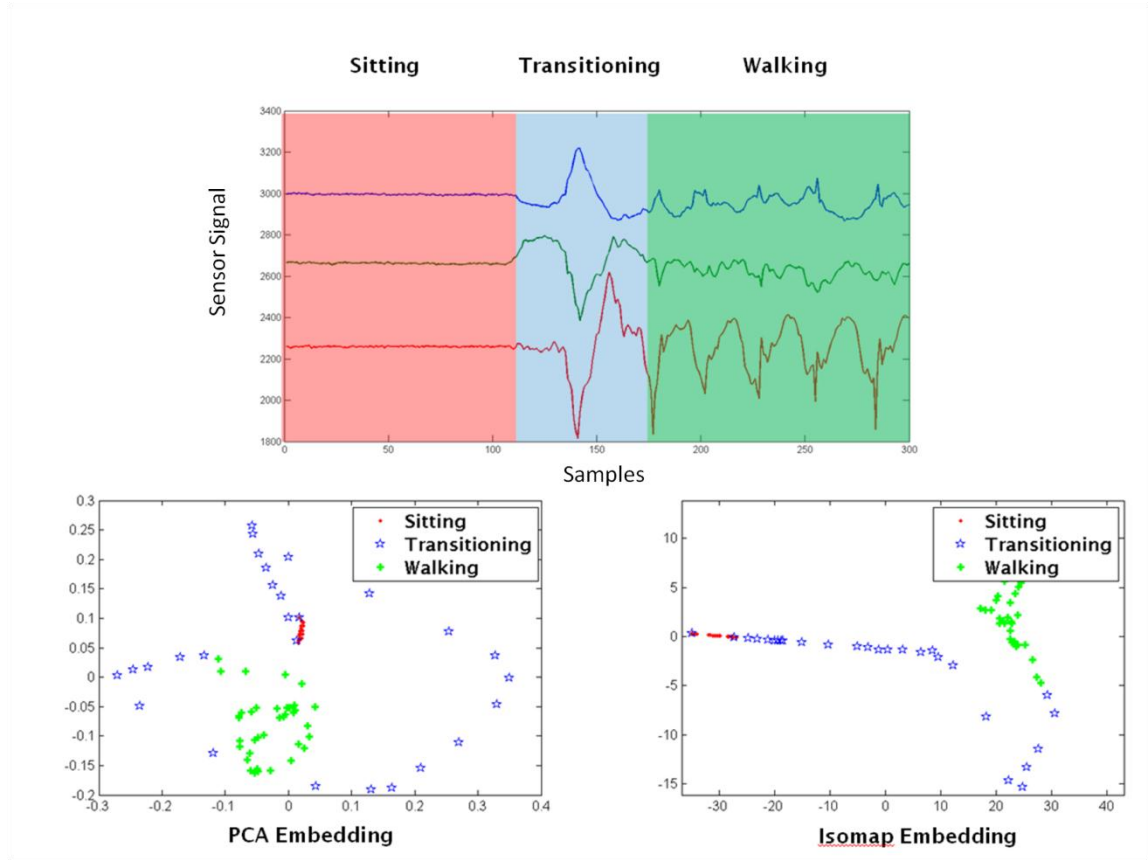


Figure 4.4 Two-dimensional embedding of Sitting to Walking sensor data (top) with first two dimensions of PCA and Isomap (bottom). Isomap preserves transitional activity relationship in embedded space, which is lost in PCA reduction.

Manifold based dimensionality reduction algorithms can preserve the shape of the data better in comparison to linear techniques such as Multidimensional Scaling (MDS) and Principal Components Analysis (PCA). Figure 4.4 shows embeddings for features extracted from sensor data when a participant transitions from sitting to walking. The manifold embedding algorithm Isomap is able to better preserve the morphing of sensor data in the transformed space compared to PCA. In case of PCA, the walking and sitting activities are projected close together, in accordance with straight-line distance. The transitional activity is scattered in the embedding, with its embedded points lying far from both walking and sitting points. In contrast, Isomap captures the transformation of the signals into two clearly separated activities with a transitional

region connecting them, as it incorporates information about the structure of the data distribution.

4.5.1 Isomap

Isomap preserves structural relationships as shown in Figure 4.4 by approximating the geodesic distance in manifold dimensions. Distance along the surface of a manifold is called geodesic distance. This may be significantly different from straight-line distance. As PCA preserves pair-wise Euclidean distances (hence its equivalence to Euclidean MDS), Figure 4.4 can be used to compare Euclidean and geodesic distance relationships. Signal corresponding to sitting can be closer to walking than the transitional activity in unconstrained space. In contrast, on a manifold model of the activity, to go from sitting to walking the transitional region must be encountered. This would result in a higher geodesic distance.

Isomap embeds data in space that preserves the geodesic distance between each pair of points. The manifold neighbourhoods are used to construct a graph representation of the data and graph distance is used to approximate geodesic distance. Given n points, a pairwise distance preserving low-dimensional embedding Y will minimise the cost function

$$E = \|\tau(D_G) - \tau(D_Y)\|_{L^2} \quad (4.8)$$

where D_G represents pair-wise geodesic distance, D_Y represents pair-wise Euclidean distance in the embedded space and τ is an operator designed for efficient optimisation defined for distance matrix D as follows:

$$\tau(D) = -\frac{HSH}{2} \quad (4.9)$$

where :

$$H = I_n - \frac{1}{n}O$$

$$[S]_{ij} = [D]_{ij}^2 \text{ for } 1 \leq i \leq n, 1 \leq j \leq n$$

where $[S]_{ij}$ and $[D]_{ij}$ refer to the i^{th} row, j^{th} column components of matrices S and D respectively, I_n is the identity matrix, O a matrix of $n \times n$ ones. Eq. (8) can be minimised by setting Y as the top eigenvectors of $\tau(D_G)$.

The primary complexity in Isomap lies in estimating the geodesic distances between each pair of points. For large datasets it can be time-consuming to construct a graph representing the data and find shortest paths on it. There are other techniques that do not preserve pairwise distances, such as Locally Linear Embedding (LLE) [196] and Laplacian Eigenmaps [197]. These are described in more detail in Appendix A. While both LLE and Laplacian Eigenmaps are faster than Isomap, both attempt to preserve locality instead of a global property like pairwise distance. As Isomap preserves global properties, under certain theoretical assumptions it is guaranteed to uncover the optimal embedding representative of underlying manifold dimensions. This is important as we intend to analyse transitional activities and differentiate between them both visually and quantitatively. Therefore we use the more computationally intensive algorithm of Isomap in our analysis.

4.5.2 Estimating Intrinsic Dimensionality

Tanenbaum [188] develop the idea of using residual variance to estimate the extent to which a low-dimensional representation captures the high-dimensional data. This is computed using the cross correlation between pair-wise distance matrices in the full feature space D and low-dimensional space D_e for an embedding e . Residual variance is defined as

$$R = 1 - correlation(D, D_e)^2 \quad (4.10)$$

It is demonstrated that residual variance decreases exponentially with increasing dimensions; initially adding dimensions decreases error significantly, with smaller subsequent gains. The ‘elbow’ of such an exponential curve i.e. the point at which further dimensions result in little or no reduction in residual variance, can be regarded as the intrinsic dimensionality of the data. This measure can also be used to compare the performance of dimensionality reduction algorithms.

An alternative is to consider the classification accuracy of algorithms using the manifold embedded data. This may allow us to use fewer dimensions than the intrinsic dimensions of the data for our analysis, as only some of the intrinsic dimensions may be relevant to the classification task.

4.5.3 Representing Learned Manifolds as Models

While an important application of manifold embedding is the visual exploration of a historical dataset in its intrinsic dimensionality, it is important in our application to handle new data without requiring a re-computation of the manifold model. There are two main approaches for this. The first is to learn a manifold from a sample of the data, and map new points to this manifold through its nearest neighbours as in Landmark-Isomap [198] and FR-Isomap [199]. The second approach relies on training regression classifiers such as Generalised Regression Neural Networks (GRNN) [200] and Radial Basis Function (RBF) [201] neural networks with feature vectors as inputs and manifold vectors as outputs. The advantage of this is that once the models are learned, it is faster to map incoming data into manifold space than to map new points to an existing manifold using, for instance, Sammon’s mapping in FR-Isomap. As the intended use of this work is in pervasive sensing environments with continuous data collection, it is infeasible to run an expensive algorithm at runtime. Therefore we adopt the latter approach, and show results for three classifiers in Section 3.4: a linear regression classifier, a GRNN classifier and an RBF classifier.

4.6 Categorisation and Analysis

With segments mapped to low-dimensional space, subsequent processing aims to categorise them into the transition types, and analyse the transitions within their categories. In previous work [136, 139, 140, 186, 187], fixed sized windows have been used for classification of activities that included transitions. In contrast, we will utilise variable length windows comprised of the segmented transition found in the previous step. There are several advantages for taking this approach. Given the high variability in the length of transitions, it is difficult to fix a window size beforehand for the transitions, particularly when dealing with a range of transition types, as involved in this work. Furthermore, utilizing windows found in the previous step directly may potentially remove the need for manually placed markers, allowing a seamless progression from transition to categorisation to analysis.

One solution to this is to learn a mapping from the feature space to a reference manifold space using a subset of the transitions [202]. Features described in Table 4.2 are extracted from these transitions. This is mapped into low dimensionality space using Isomap. For each dimension, a

linear regression model is learnt, mapping from the feature space to the manifold. New data can be mapped to the manifold using these models.

The categorisation problem is the mapping of transitions into broader categories described in Table 4.1 and subsequent analysis is performed within each category. In the following sections, we provide results using the three well-established classification algorithms introduced in Chapter 3: C4.5 based decision trees, a Naive Bayes based probabilistic classifier and a Multilayer Perceptron.

4.7 Experimental Setup

A wireless ear-worn activity recognition sensor incorporating a three-axis accelerometer is used for this study [115]. Previous work has shown that positioning the sensor on the ear provides an effective means of capturing trunk motion, shock wave transmission through the body, and thus allows the classification of ground reaction force and gait [50, 97]. Data is time-stamped and can be visualised in real time. A sampling rate of 50 Hz was used in all experiments. During experiments, the software also allows for both online and offline visualisation and annotation.

Transition Number	Transition Category	Transition
1	Stepping	Step up onto bench
2	Stepping	Step down from bench
3	Sit/Stand	Sit on bench
4	Sit/Stand	Stand from bench
5	Pick Object	Pick up a 4kg weight after walking to it
6	Sit/Stand	Stand up from chair
7	Sit/Stand	Sit down on chair

Table 4.1 Categories of transitions performed by participants in Study I and Study II.

The proposed method is evaluated on transitions selected from the DynaPort KneeTest® [203] protocol. The protocol has been validated for assessing quality of movement of patients recovering from knee replacement surgery [204] through comparison with observations by physiotherapists. Although the protocol consists of 23 activities of daily living in total, transitional activities listed in Table 4.1 were selected for detection and analysis. The DynaPort

system collects data using trunk and leg-worn accelerometers, while our experiments used a single ear-worn sensor.

Two studies were performed, hereon referred to as Studies I and II. In Study I, ten normal participants performed the protocol under controlled laboratory settings. The mean age of the participants was 27.1 years, with a standard deviation of 3 years, seven of whom were male and three were female. Their mean Body Mass Index (BMI) was $22.6 \text{ kg} / \text{m}^2$ with a standard deviation of $3.1 \text{ kg} / \text{m}^2$. Additionally, the transitions listed in Table 4.1 were repeated four times to provide the training data for classification. Impairment was subsequently simulated for the sit/stand and step category transitions using abdominal and knee braces.

In Study II, we evaluate our methods on data collected from five patients undergoing total knee replacement surgery, with ethical approval from St Mary's Research Ethics Committee (reference number: CI/2007/0043). The mean age for the patients was 64.8 years, with a standard deviation of 4.3 years, three of whom were male, and two were female. Their mean BMI was $31.3 \text{ kg} / \text{m}^2$, with a standard deviation of $6.5 \text{ kg} / \text{m}^2$. Data was collected 1 week before the operation, then at 1, 3, 6, 12 and 24 weeks after the operation. The stepping up and down bench transitions were performed for both left and right legs. Similarly, picking up the weight was performed on the left and right side. In the stepping up and down and sitting at the bench transitions, two benches were used, with respective height being 30 cm and 40 cm.

Due to the nature of the operation, patients were not always able to complete all the tasks. During the first few weeks after the operation, patients had very restricted range of motion in the operated knee. Patients usually found it impossible to stand up from benches without assistance. Similarly, it was difficult for patients to complete step up and step down transitions while placing weight on the operated leg.

Figure 4.5 shows some example data from the accelerometer when the step up and down from a bench transitions are performed. The topmost plot shows data collected from a healthy participant, whereas the bottom two plots show data from a patient at 3 and 24 weeks after the surgery. While at 24 weeks the recovering patient's data is similar to the healthy participant's data, at 3 weeks the transitions are extended, and illustrates a different pattern pertaining to the recovery phase of the patient.

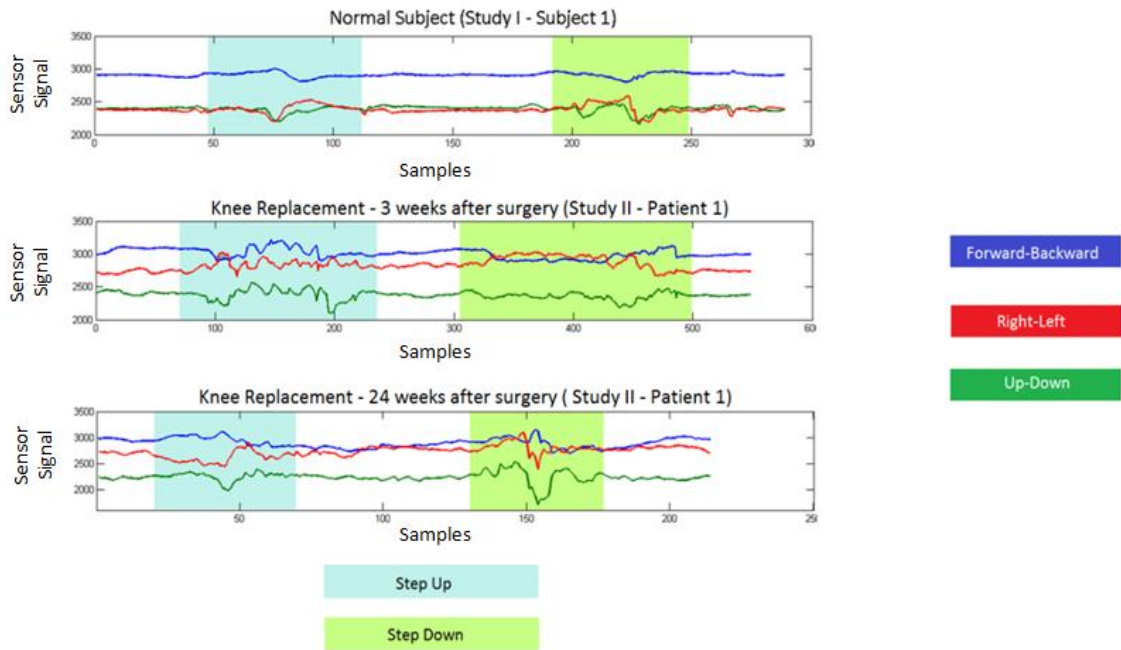


Figure 4.5 Accelerometer data from participants stepping up on, and down from a 20cm high bench. The signals show a difference between healthy participants and participants moving with difficulty after 3 weeks of surgery.

4.8 Results

Results are presented for each step of the framework for Studies I and II. While the same basic protocol is used in both cases, Study I allowed more detailed analysis because it was collected in controlled laboratory environment, with additional steps in the protocol to simulate impairment. Study II is significantly more challenging than Study I, owing to the varied environments of patient homes, and the difficulty of performing activities during post-operative recovery and the variation in patterns of patient recovery. Patients often needed assistance during the protocol to perform activities, or were not able to complete it. In particular, transitions 3 and 4 were difficult for patients to complete. Data was not available for patient 2 during 1 week after the operation. To ensure safety of the patients, a member of staff was always in close proximity to them to prevent falling. Our analysis focuses on the Step Up, Step-Down transitions, which did not require assistance beyond that.

4.8.1 Feature Extraction

Feature extraction from the e-AR sensor consists of the extraction of a set of features from each axis of the accelerometer. A fixed size moving window of 100 samples is used (comprising 2 seconds of sensor data) with an overlap of 30 samples between consecutive windows. Table 4.2 lists features extracted from each axis. These are commonly used features for activity detection [62, 205, 206], including statistical features, wavelet features and frequency features derived from Fast Fourier Transform (FFT) analysis. Chapter 3 discussed statistical and wavelet features for activity recognition. Instead of extracting statistical features from raw data, they are extracted them from normalised data. While these features are going to have the same shape as those extracted from raw data we may correct for baseline changes in the data, which may induce shortcut edges in the manifold approximation. The baseline of the accelerometer channels can shift based on factors such as sensor orientation, placement and battery power. Therefore signal energy, mean and variance are extracted from data normalised with respect to historical mean and standard deviation of the full dataset. In addition to statistics from normalised data, we also include variance from raw data, as this statistic has previously been used for characterising recovery post-surgery [115]. The FFT feature has been used in previous studies for sit-to-stand analysis [207] and activity recognition [208]. As in Chapter 3, cross-channel features are extracted in addition to the channel features. These included the channel cross correlation, and ratios of the means of each channel with the other two. The total number of features extracted is 36.

Feature Number	Channel Feature
1	Energy of normalised data
2	Mean of normalised data
3	Variance of normalised data
4	Variance of raw data
5	Mean of wavelet transform at level 1
6	Standard deviation of wavelet transform at level 1
7	Mean of wavelet transform at level 2
8	Standard deviation of wavelet transform at level 2
9	FFT feature – maximum signal power

Table 4.2 Features used for Impairment Detection from Activities of Daily Living.

4.8.2 Transition Detection

Transitions detected automatically are compared against labels placed manually. Annotations were done in parallel to data collection for each activity in the protocol. Transitions within activities were labelled offline. Transitions detected by the algorithm are labelled according to the ground truth label with the most overlap. Tables 4.3 and 4.4 show results for this analysis on the two datasets. Performance is sensitive to the recursion level to which the algorithm is executed. Figure 4.6 shows aggregate accuracy values for each level.

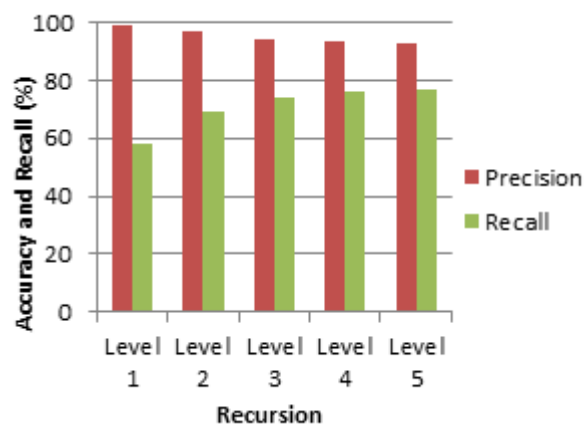


Figure 4.6 Levels of Accuracy, Precision and Recall against recursion level in Transition Detection Algorithm for Study I. Both statistics are influenced by level of recursion.

It can be seen that the recall metric is sensitive to the recursion level in particular. This is due to the fact that with increasing recursion depth, the segment boundaries grow tighter. There is a smaller false-positive rate compared to the manual annotations, and therefore the rate of recall improves. This corresponds to a decrease in under-segmentation. Conversely, precision decreases with recursion; this reflects over-segmentation, where a transition may be split into more than one segment. An example of this is given in Figure 4.7, showing a participant transitioning between standing to sitting and then standing again. The first transition is segmented completely at two levels of recursion. The second transition is under-segmented at the second application of recursion, but over-segmented at the next level.

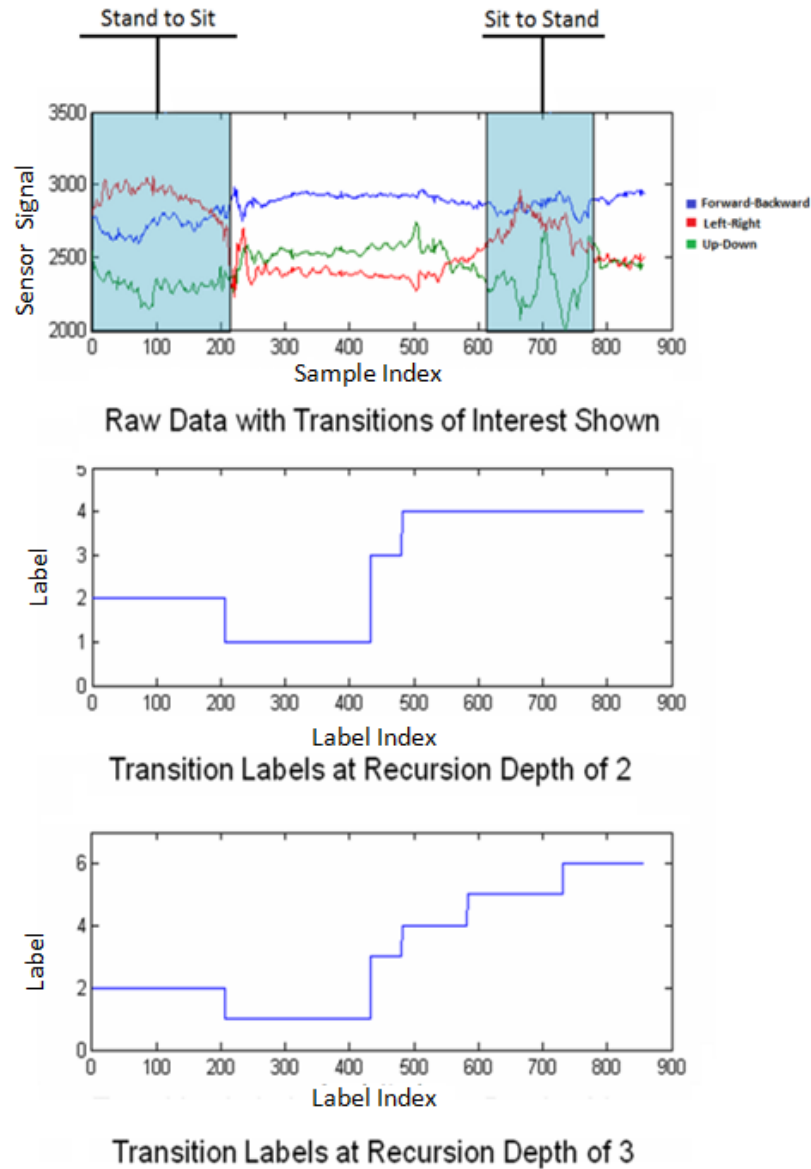


Figure 4.7 Transitions detected using recursive partitioning method at different levels of recursion. The Sit-to-Stand transition is under-segmented when the recursion depth is 2, and over-segmented at the next level.

It can be seen in Figure 4.7 that there is a trade-off between precision and recall with increasing level of recursion in the first three levels. However there is not a significant difference between Level-4 and Level-5 recursion statistics. Since this offers the highest average precision and recall, Level-5 segments are used in to compute accuracy in Tables 4.3 and 4.4, and used for subsequent analysis.

Transition Number	Transition	Precision (%)	Recall (%)
1	Step up onto bench	99.1	74.2
2	Step down from bench	99.6	72.6
3	Sit on bench	92.8	80.3
4	Stand from bench	77.7	80.0
5	Pick up a 4kg weight after walking to it	92.9	78.6
6	Stand up from chair	96.5	78.6
7	Sit down on chair	85.7	77.2

Table 4.3 Accuracy of Transition Detection Algorithm for Study I.

Transition Number	Transition	Precision (%)	Recall (%)
1	Step up onto bench	79.7	71.9
2	Step down from bench	78.6	71.6
3	Sit on bench	86.3	74.3
4	Stand from bench	88.4	73.9
5	Pick up a 4kg weight after walking to it	87.9	72.1
6	Stand up from chair	89.5	72.3
7	Sit down on chair	86.2	73.3

Table 4.4 Accuracy of Transition Detection Algorithm for Study II.

4.8.3 Dimensionality Reduction

The manifold assumption can be validated empirically by comparing performance with a popular linear algorithm Principal Components Analysis (PCA). The residual variance metric is used as a measure of the error. Averaged error for separate embeddings of the seven transitional activities is plotted in Figure 4.8. The Isomap algorithm reduces residual variance significantly within four dimensions. As discussed in 4.5.2, the ‘elbow’ of the residual variance indicates the intrinsic dimensionality of the data. Isomap is able to represent the essential variations in the data with fewer dimensions, and lower residual variance.

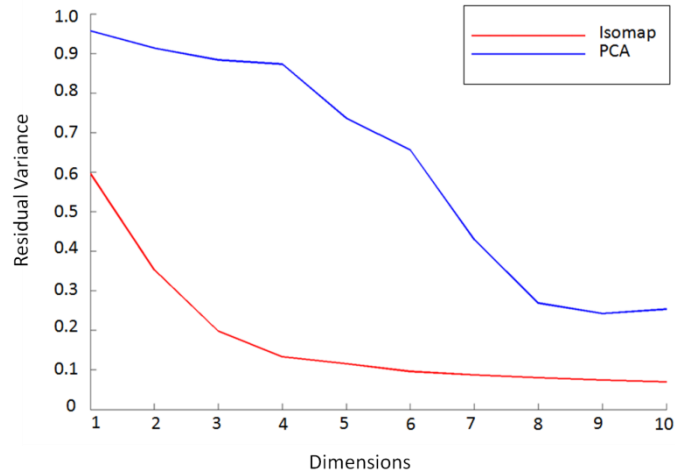


Figure 4.8 Comparison of the performance of Isomap against PCA using residual variance. Isomap achieves low residual variance with fewer dimensions compared to PCA.

In order to map new data to the manifold we train regression classifiers to learn the manifold dimensions from input feature vectors. Linear regression models are computationally the simplest, followed by GRNN and RBF.

Figure 4.9 shows the residual variance errors for the three learned models for each activity. Classifiers were trained for ten dimensions of the manifold, and the residual variance of the model output is compared. While all the classifiers considered achieve low residual variance, Surprisingly however, the linear models achieve comparable performance to the more complex classifiers, outperforming the radial basis function classifiers for all the transitions. Generally the RBF models achieve good quality approximations of the manifold. Given the high accuracy as well as their efficient implementation it may be more feasible to use linear models in scenarios such as continuous home monitoring. This would allow fast online approximation into manifold space of streaming data.

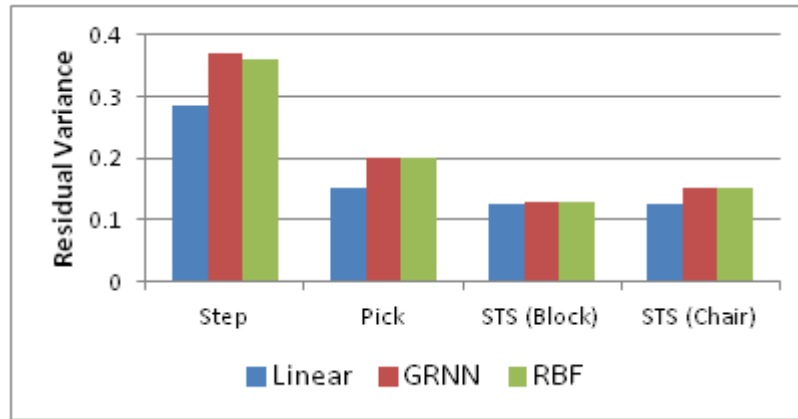


Figure 4.9 Residual Variance of classifiers learning embedding for different activities. Linear algorithms are used based on this analysis as they are low in computational cost and achieve low residual variance.

4.8.4 Transition Categorisation

Transitions are divided into three categories shown in Table 4.1. Data was sampled from each participant in Study I to construct the reference manifold from which models were trained to embed the remaining data. Each classifier was validated with ten-fold cross validation, results for which are summarised in Table 4.5. All three classifiers obtain results with comparable accuracy.

	C4.5	Neural Networks	Naïve Bayes
Accuracy	79.6	77.3	72.7
Precision	70.5	75.8	71.0
Recall	79.6	77.3	72.7

Table 4.5 Accuracy of Transition Categorisation for Study I.

4.8.5 Transition Analysis

A goal of our analysis is to analyse differences within transitions of the same type. In our protocol, the sit/stand transitions are performed at different sites, at benches of different heights (20 cm and 30 cm) and a chair. Chair height has been observed to be a significant influence in

performance of sit/stand transitions [209], particularly for elderly participants [210]. Table 4.6 shows classification results for distinguishing between transitions at the benches and those at the chair. C4.5, Neural Networks and Naive Bayes are trained with ten-fold cross validation, with C4.5 outperforming the other classifiers. We can therefore estimate where the participant is sitting based on the transitions type. An application of this would be the ability to determine whether the living environment of elderly or recovering patients is suited to their condition.

Impairment was simulated on participants using abdominal and knee braces. Participants performed the step on and off bench transition, repeating it five times from each bench leading with the left and right legs alternately, with and without impairment. The transition was therefore performed up to forty times for each participant. A sample of 250 transitions was used to train classifiers to detect impairment. It was observed during data analysis that transitions performed by participants were distinctive, and furthermore participants adapted to the simulated impairment in different ways. Figure 4.10 illustrates this, showing transitions performed by three participants with and without impairment. In both embeddings transitions performed by a participant cluster together, suggesting the feasibility of participant-specific classifiers.

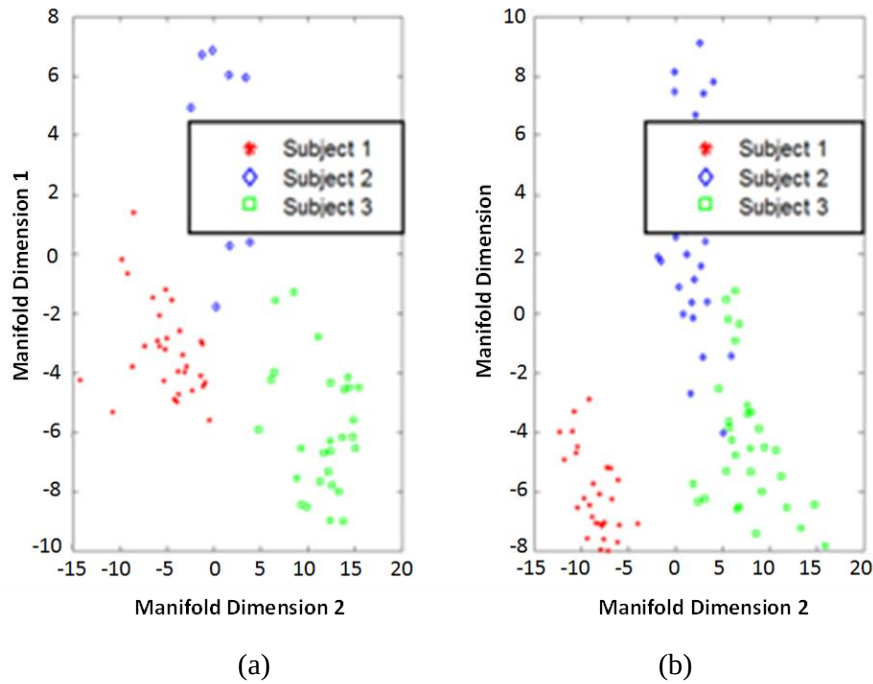


Figure 4.10 Manifold embedding of three participants performing the step transition naturally (a) and with impairment (b). The clearly separated clusters correspond to each participant.

Table 4.6 shows accuracy measures for participant-specific classifiers for five participants. Decision tree, neural network and Naive Bayes classifiers were trained for each participant, resulting in fifteen classifiers in total. Accuracy statistics in Table 4.6 have been aggregated by classifier type. Though the classifier trained on the complete dataset achieves acceptable accuracy, participant-specific classifiers perform better.

	C4.5	Neural Networks	Naive Bayes
Classifiers trained with the complete dataset			
Accuracy	76.0	76.7	76.3
Precision	76.9	77.4	76.4
Recall	76.0	76.8	76.4
Participant Specific Classifiers			
Accuracy	82.7	85.4	77.6
Precision	83.6	85.6	77.2
Recall	82.8	85.5	77.5

Table 4.6 Impairment detection using the Step Transition in Study I. Higher accuracy can be achieved by training participant-specific classifiers.

Three stages of mobility were observed in patients during data collection. The mobility of patients decreased significantly in the early stages of recovery, which gradually went back to pre-operative levels. A significant improvement was observed at the 12 and 24 week data collection stages. Figure 4.11 shows the stepping transition in the first two dimensions of the manifold space for patients performing the step-up and step-down transitions. These embedding for this transition mirror some of the patterns observed during data collection. Transitions for each stage of data collection are close to each other. Stages of recovery observed during data collection can be seen; early recovery (1-6 weeks) and late recovery (12-24 weeks) tend to cluster at a distance. The pre-operative data points exhibit more variability but tend to cluster closer to the late recovery stage. This is because patients have different levels of impaired mobility before surgery.

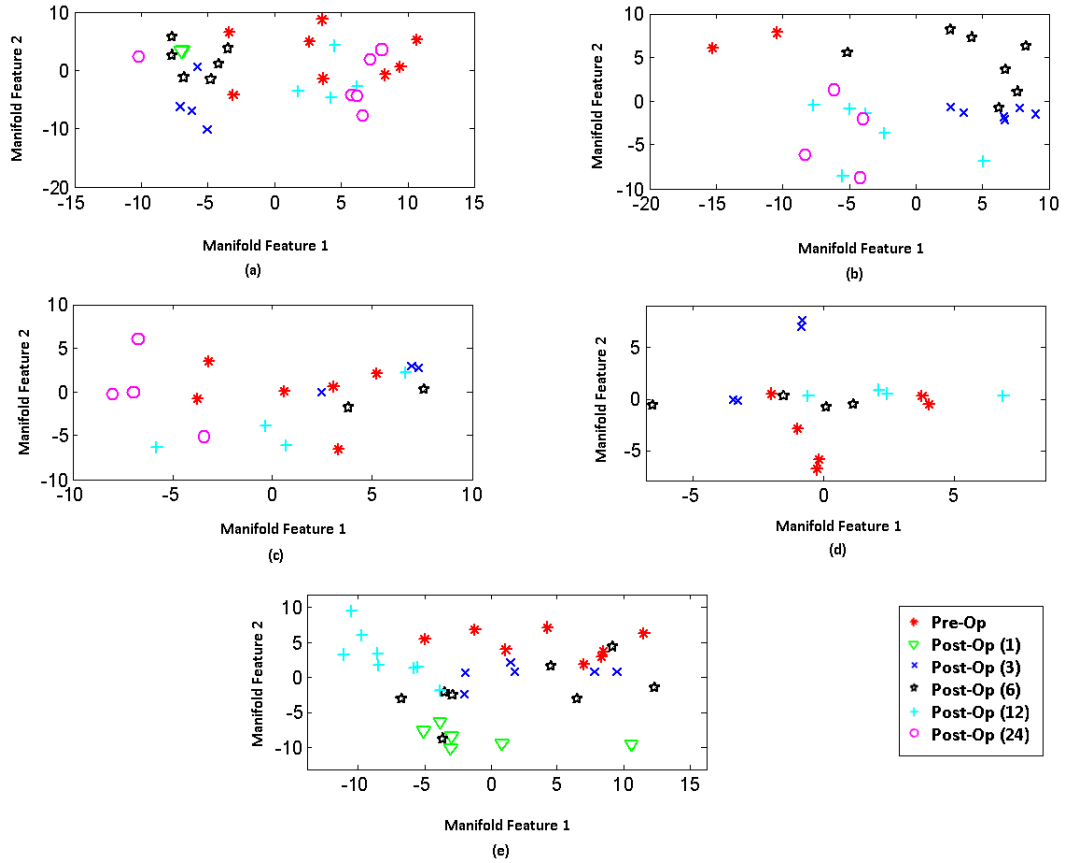


Figure 4.11 Stepping transitions for patients at different weeks of recovery embedded in separate spaces. Figures (a)-(e) show Participants (1-5). Early recovery (1-6 weeks) and Late Recovery (12-24) stages are embedded closer together.

The separation between early and late recovery can be seen in all five patients, as shown in Figure 4.12 (a) and (b) for the step and stand to sit transitions. Patient 5 showed greater progress during early weeks of recovery, which is reflected in both the completion of step activities after just one week, and the proximity of the clusters for each week. It is worth noting that Patient 4 underwent a hip replacement before week 24 of the study, and therefore the data collection was terminated.

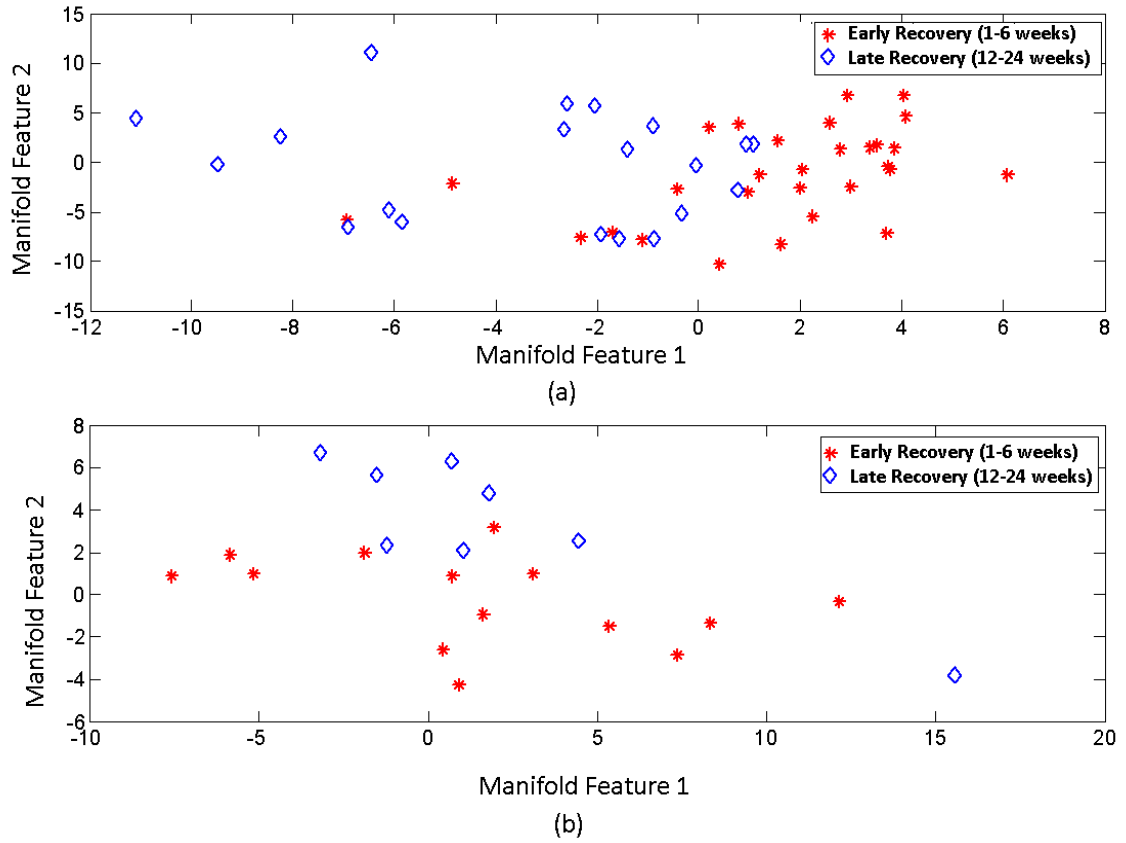


Figure 4.12 Transitions from five patients recovering from surgery. Figures show embedding of all patients together, with Step-up transitions in (a), and Stand-to-Sit transitions (chair) in (b). Clusters correspond to two stages of recovery observed during data collection.

However, care must be taken in interpretation of this data. There are relatively few points for each transition. For instance, in Figure 4.12 (a), the post-op (1-week) transition is relatively close to transitions at six weeks. The patient however was able to perform the transition only once, and was unable to perform it with the impaired leg, or with the higher bench. The distance and the embedding alone do not reflect this difficulty. It is therefore more suitable to see the figures as indicative of changing patterns where small differences in distance should not be seen as significant.

While Sit-to-Stand is the most commonly studied transition, patients were usually unable to perform it from the standard sized benches after the operation. The protocol included an additional transition to and from a chair. This however, introduces variability, given the inability to control chair height and armrest use at patient homes. These patients are often unable to

perform the transition. When the transition is performed, patients may require assistance in performing it, leading to uncertainty in interpretation.

Conversely, these patients were able to perform the step-up and step-down transitions with greater ease. It was possible to visualise the transitions in manifold space, and observe patterns similar to those seen during data collection. If more data is collected for this transition at each week, it may be possible to train impairment detection classifiers similar to those trained for Study I.

4.9 Conclusions

In this chapter, we have presented a transition detection and analysis framework that utilises a single lightweight wearable sensor to detect and analyse transitional activity in a pervasive monitoring context. The framework is designed to reduce human error and involvement, by utilizing unsupervised learning methods for automatic segmentation and dimensionality reduction. The key contribution of this chapter lies in a novel application of manifold modelling, partitioning and embedding to sensor data for transitional activity analysis.

Using data collected in laboratory settings, the framework was evaluated for the detection and categorisation of transitions. Manifold-based dimensional embeddings of these transitions were used to train classifiers for detection of seat type and simulated impairment. It should be noted that the results shown for three classifiers are meant to demonstrate applicability of our approach to generic classifiers, however they are not meant to demonstrate the superiority of a particular type of classifier for this problem. For instance, C4.5 has a performance advantage over Naive Bayes ranging between 4-7%. Decision trees are however, susceptible to overfitting and learning over-complex rules. In contrast Naive Bayes chooses to simplify the classification problem by assuming independence between classes. Therefore, the performance difference does not necessarily mean C4.5 is more suitable for analysis of transitions than Naive Bayes. It was observed that participant specific classifiers are more suitable for the impairment detection problem because participants performed transitions and adapted to impairment differently.

The framework was also applied to a study of patients recovering from knee replacement surgery. The low-dimensional embedding of the automatically detected transitions allowed a

visual analysis of the patient's recovery. While Sit-to-Stand is studied extensively in clinical rehabilitation studies it showed less promise as a recovery measure owing to difficulty in performing the transition from standard sized benches, and variability in chairs at patient homes. Conversely the Step transition was performed with more reliability and ease while also showing patterns of recovery. It is therefore indicated that the Step transition may be a useful metric for post-operative recovery in future research.

The proposed study advances from simple activity categorisation to detailed transitional analysis. We believe this is more meaningful for clinical studies for assessing the efficacy of therapeutic procedures or the gradual deterioration of chronic symptoms. The ability to monitor these changes continuously in a normal environment represents a unique strength of BSN. With such a framework, it is also possible to perform classification of transitions amidst a broader range of confounding activities, such as those performed in home environments.

So far we have considered the analysis of activities at short time-scales, in a scenario where class labels are available. In chapter 5 abstracted pictures of routine are derived from classifications of activity with a view towards mining historical data collected from participants living in pervasive sensing environments.

Chapter 5

Discovering Routines from Activities

5.1 Introduction

As pervasive healthcare technology matures with more real life systems beginning to be deployed, methodologies for profiling user behaviour over the long term have come into sharper focus. While considerable science exists in identifying what users are presently engaged in, characterising their behaviour over the longer term remains a relatively open area. This chapter describes a system, presented at [211] for visualizing an important aspect of the user's behaviour: their routine.

Though there is considerable variation in a person's daily activities, there may be a repeating temporal structure over the long term in the pattern of activities during the course of a day, which is referred to here as routine. Beginning with knowledge of the user's activities, which can be discriminated in pervasive environments, such as sitting, walking, exercising, sleeping, getting dressed etc, we aim to characterise the patterns of activities that occur during a day in someone's life. Thomas Nagel in a famous essay on consciousness [212] asked, "What is it like to be a Bat?" Although our research aims are by no means comparable to addressing the mind-body problem, we may at least attempt to give care providers, who currently rely on questionnaires, to get a good idea of what a day for one of their care recipients is like. The range of medically relevant information encapsulated in a person's routine includes eating habits, sleep schedules, exercise durations and timings, day of the week effects and much more.

There is a second concern we address in this chapter, that of synopsis structures [71]. The volume of data modern computer systems need to handle is extraordinary. Most of this data however is presently not processed and is stored in large server farms. Though the business value of this data has always been clear, the question has always been one of data volume outstripping analysis tools. Pervasive systems, such as BSN based pervasive healthcare environments, will generate by design continuous, multidimensional data streams. Much of this data will be experimental in nature in the beginning and will therefore be archived for the long term. On the other hand, in order to scale up, these systems will need to employ synopsis structures (see Section 2.1) that can act as surrogates for the data for most queries, only retrieving the raw data when necessary. It is also a good strategy to perform this incrementally, shifting the processing cost over time and over computational resources.

This chapter proposes a system, for constructing a synopsis structure specifically for describing routine. This structure is constructed by mining activity data extracted from the e-AR sensor. It is a multi-resolution picture of the user's routine, showing the patterns of activity at progressively finer time granularities where there is more detail to uncover. In future work this can tie in with the activity transitions work described in the previous chapter. In one sense, activity transitions represent the finest resolution of activity pattern. Since the aim is to develop usable systems that can scale to a large number of users, we consider here design criteria like incremental processing, compression, and visualisations for service users. As a synopsis structure, it is also important to represent the base dataset accurately, as far as queries pertaining to routine are concerned. We demonstrate our work on simulated data constructed from lab collected e-AR sensor data, along with several real datasets.

5.2 Activity Classification

We use a multivariate Gaussian Bayes classifier for classification of activity from e-AR sensor data. Developed by Lo et al. [115], this classifier outputs a level or class of activity, relying on the variance and range of the data. The activity level can take on four values and higher activity levels indicate activities with involving significant movement.

A multivariate Gaussian density models each activity class. Given a feature vector x consisting of the variance across 3 axes, the likelihood of belonging to a class C_j is defined as:

$$P(x|C_j) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \right] \quad (5.1)$$

where n is the dimension of the feature vector, μ_j the mean and Σ_j the covariance of the class C_j . The mean and covariance are derived from the training data. Given a prior over the classes $P(C_j)$ and a normalizing constant α , according to Bayes theorem the posterior probability of C_j for a feature vector x is

$$P(C_j|x) = \alpha P(x|C_j)P(C_j) \quad (5.2)$$

A uniform prior is $P(C_j)$ assumed over the classes. It is argued that this removes bias for more frequent classes.

Figure 5.1 shows an example of the activity level derived from the e-AR sensor signals. The top graph shows the raw sensor signals, the middle graph shows the outputs of the activity classifier, and the bottom graph denotes a coarse segmentation into distinct activities derived by filtering the activity levels. A median-filter is used to remove the noise from the activity classifications.

One rationale for providing a class of activity rather than a more specific label lies in privacy preservation. That said, often the specific nature of the activity given a class label may be reasonably inferred from the time of day. If the user has a persistent low activity level at a late hour of the night, it is suggestive that he or she is sleeping. If the activity persists into a late hour of the day with no change, there may be cause for alarm.

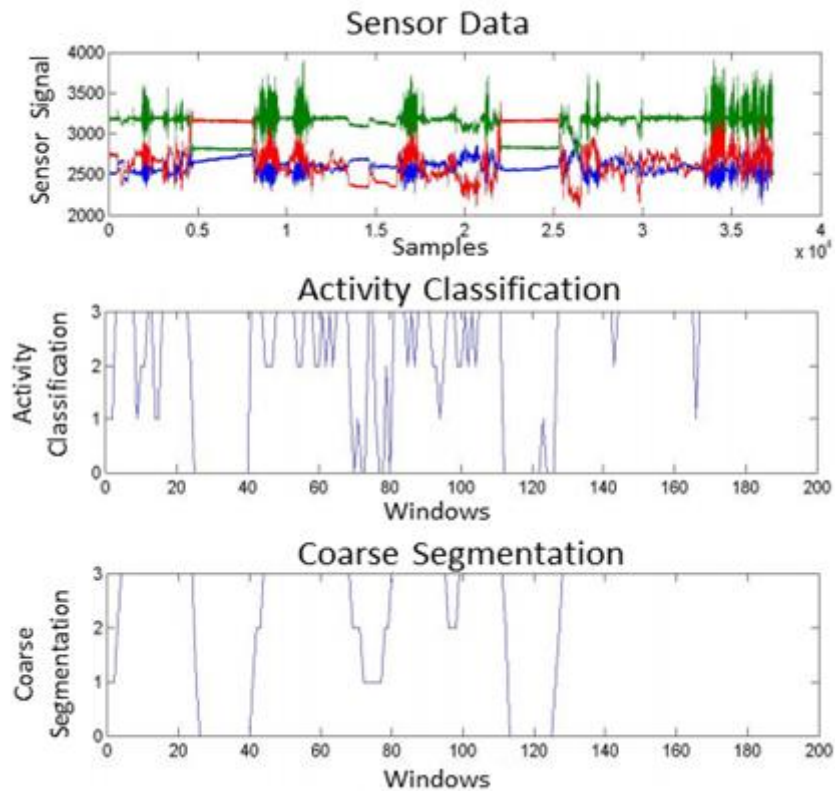


Figure 5.1 Segmentation of raw sensor data into Activities using the Activity Classifier. Each window corresponds to 200 samples. A coarse segmentation can be achieved by filtering the activity classification, removing noise. The classifications can be subsequently mined for routine extraction.

The classifier is treated as a black box in the next steps; more descriptive class labels can be provided without change to the pattern mining procedure. A streaming infrastructure developed during the course of this dissertation enables us to process the data as it arrives. The sensor sends via the gateway a stream of activity levels. These are generated on the gateway or on-board the sensor. The streaming infrastructure routes the activity stream to the data component responsible for producing the routine synopsis.

5.3 Mining Routine Activities

While a few activities may be described by a single activity level, many activities may produce a characteristic combination of activity levels over a period of time, for example, exercising and working. Frequent pattern mining can discover such combinations.

5.3.1 Background

Frequent Pattern Mining [151] is the task of searching for frequently repeating patterns in a database. We introduce some terminology to describe the problem. A transaction database D is a set of transactions. Each transaction T is a subset of items $I = \{i_1, i_2, \dots, i_n\}$. A pattern is a non-empty subset of I . A transaction T is said to contain a pattern P if $P \subseteq T$. The support of a pattern is defined as $\pi(P) = |\{T \in D \mid P \subseteq T\}|$, that is the number of transactions that are a superset of the pattern. A pattern is considered to be frequent if its support is above a user specified threshold. The first application of this was on supermarket data to find what products customers tended to buy together, the goal then was to place these products in close proximity. This problem has exponential complexity if performed with a brute force search, $O(|I|2^{|I|})$. There are ways to prune the search however. The so-called Apriori principle [213] is that if P is frequent, all subsets of P are also frequent. Conversely if P is not frequent, all supersets of P can be discarded.

FP-Tree [214] is a prefix tree structure, which can be constructed from two scans of the databases. This compressed data structure can then be mined, instead of the database itself. A prefix tree is defined as a tree such that each node has its parent node as its prefix. Each node in the FP-Tree is an item-set in the database containing a frequent item, and the support for the item-set. To illustrate the construction of the FP-Tree, consider the simple dataset shown in Figure 5.2 containing three transactions. Each transaction is time-stamped, and consists of the set of activities occurring in the period beginning with the timestamp. The duration of each transaction is assumed to be fixed. This can be obtained by using a fixed-size sliding window over the activity stream generated from the raw sensor signal.

The first pass over the database counts the frequency of each item (sometimes called the 1-length pattern). This can be used to impose an order on the nodes when they are added to the tree. The ordering of the example above can be computed as $\{1, 3, 2, 5, 4\}$. The ordered table associating items with frequencies is referred to as the header. A second pass can then pass through the dataset to produce an initial FP-tree. Figure 5.3 shows the tree constructed by reading the three records in the database. The first record in the database is read and placed in the FP-tree as in Figure 5.2(a). The header contains a link from the node entry to corresponding node in the tree, shown by dotted arrows. When the second transaction is read, the node corresponding to item 1 can be reused, as it has the same prefix, and therefore the pattern is a

sub-tree of the node corresponding to item 1 added for the first transaction. The count of the node is updated to reflect this information. Finally for the third transaction, a similar procedure is followed for adding the prefix 1,2. Note the link from the two nodes corresponding to item 3, linking them to the header table.

Time	Activity Levels
19:00	1,3
19:01	2,5,1
19:03	1,2,3,4

Figure 5.2Example of an activity database where activity levels are aggregated every minute.

Note that this is a representation of the transactions in the database as combinations of items, not an enumeration of patterns in the database, which need to be mined from this data structure. It is however, a highly compressed representation of the database, and has been observed to often be orders of magnitude smaller than the database in practice. A common approach is to construct this tree, then adopt various search strategies for finding frequent patterns within this representation. Frequent patterns can be found by ‘growing’ patterns. This is done by projecting the initial tree along item dimensions. For instance, by projecting along the item 3, the patterns {1, 3} and {1, 2, 3} can be found. The search space can grow very large, but the FP-Tree is a good optimisation, and avoids the need for repeated scans of the database.

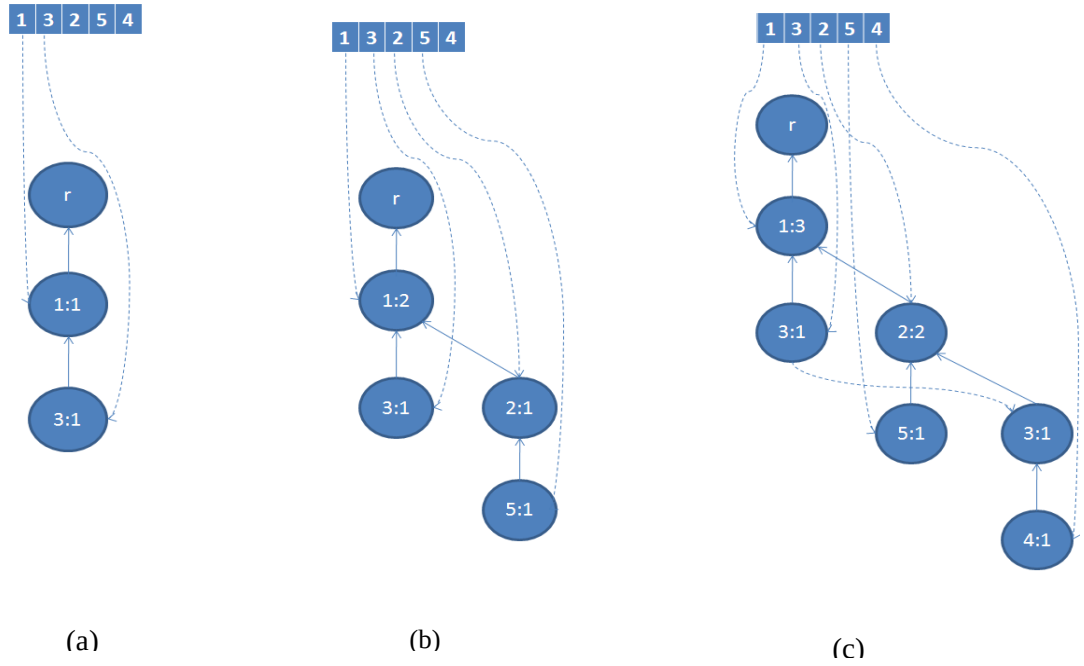


Figure 5.3FP-Tree after scanning the first three transactions respectively (a), (b) and (c). The node is labelled as Item:Count. It is possible to mine frequent itemsets in the database using this data structure without rescanning the database.

Closed Pattern Mining is a related problem, where the goal is to find frequent patterns that have no proper superset with the same support. Closet+ [215] is one recent algorithm for this problem. As before a data structure similar to FP-Tree is constructed, then efficiently mined. In addition to the Apriori pruning, two further search space pruning rules are usually used in closed pattern mining. They are as follows:

- *Item Merging*: If every transaction containing a frequent pattern X also contains pattern Y but not any proper superset of Y , then $X \cup Y$ forms a frequent closed pattern and there is no need to search any pattern containing X but no Y .
- *Sub-Pattern Pruning*: Let X be the frequent pattern currently under consideration. If X is a proper subset of an already found frequent closed pattern Y and $\pi(X) = \pi(Y)$, then X and all of X 's sub-patterns cannot be frequent closed patterns and can be pruned.

In addition to these pruning rules Closet+ proposes a further pruning and search strategies, for example switching between top-down or bottom up search depending on the sparseness of the data, using hashing for efficient subset matching, and memory-efficient implementations of the projections of the database for FP-tree construction.

5.3.2 Routine Tree

Frequent Pattern Mining aims to find patterns in a set of transactions that occur more frequently than a user specified threshold. In this case each transaction is a window over the activity data stream. The goal therefore is to find frequent patterns of activity levels. We call the pattern with the highest support the persistent activity pattern.

The FP-Stream [73] data structure extended the FP-Tree to store information for patterns over multiple time granularities. For each pattern a table is maintained that stores the support for the pattern over progressively larger time-granularities. These time granularities are mathematical progressions of time t such as $t, 2t, 4t, 8t, \dots$, or if considering natural time, then in terms of progressions of quarter hours, hours and days.

We propose a data structure to represent routine in a way that emphasises the temporal composition of a person's day. Time durations also do not need to be in fixed time buckets, thereby allowing for the unpredictability of activity times. The data structure, called Routine Tree is a variable resolution mapping of time periods within a day to frequent patterns of activity levels. Each node in a routine tree is a time-period associated with a table of frequent patterns along with their support. It can be seen as a converse data structure of the FP-Stream, except that we do not require the time periods to be of uniform size. Adjacent nodes in the tree are temporally contiguous (if the data is continuous), and represent a non-overlapping division of time. The persistent activity pattern is the criterion for division of time: adjacent nodes show a change in the persistent pattern. Figure 5.4 shows an example of such a tree.

This data structure presents an intuitive view of the user's routine. Moving down the tree we get more detailed information of the user's activities, whereas nodes near the root contain patterns which persist for longer portions of the day.

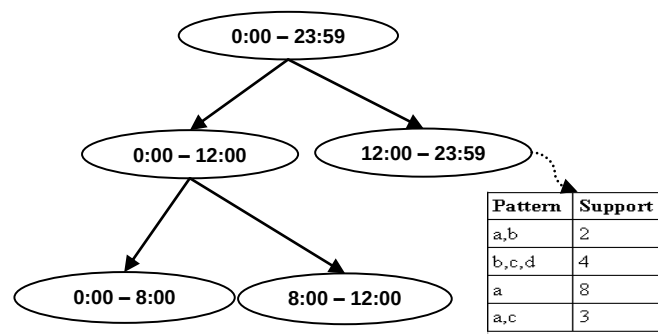


Figure 5.4 Routine Tree with support shown between 12:00 and 23:59. Each node in the tree is associated with a table of activity patterns frequent in that time period.

The tree is generated in two phases. First the tree is constructed by mining activity data in progressively smaller time durations. This tree is then pruned to produce a more compact representation.

5.3.3 Transaction Representation of Sensor Data

Association mining algorithms impose certain assumptions on the data. The transactions generated in our activity database therefore have the following properties:

- **Pattern length:** Pattern lengths are bounded by the alphabet size of the database; the complexity of the patterns found depends on the number of activity classifications possible. A second dimension is introduced in the windowing of the data. The maximum length of the pattern is the number of distinct activity levels that can be generated in the windowing period.
- **Repetition of Items within Transactions:** As association mining algorithms ignore sequences of repeated items within a transaction, the transaction reflects the event of an activity level occurring; instead of recording each activity level in the transaction. It is possible to encode repetitions of activity levels with other symbols, or present the database with segmented data. However this imposes additional processing constraints on the system.

- **Sequence of Items within Transactions:** The sequence of items within transactions is not considered. As for repetition it is possible to encode order of items with a richer alphabet. Alternatively sequence mining algorithms can be used. These are significantly slower. As we aim to uncover the higher-level routine of the individual, in contrast to fine-grained activity classification, the faster pattern mining algorithms are preferred.
- **Handling multiple sensors:** While the activity classification only uses the wearable e-AR sensor, the methodology can be applied to any type of sensor, and multiple sensors. With regards to multiple sensors it is possible to either fuse the sensors before classification (data/feature level fusion). Alternatively, we can take a decision-level fusion approach and incorporate into the itemset alphabet items from each classifier separately. This will be explored further in Chapter 6.

5.3.4 Tree Generation

The first phase is described in Algorithm 5.1, which is close to a preorder traversal. Beginning with the complete day as the root node the binary tree is constructed by recursively executing Closet+ [58] at each node. This yields a table of frequent patterns and their supports, which is associated with that node. The node is split if it meets splitting criteria that takes into account the number of frequent patterns found, and the minimum duration at which the data is to be mined. If the node is split, the new nodes are added as children of this node in the tree, and the procedure is repeated for them.

This simple, recursive algorithm can take advantage of the structure inherent in a person's day by not mining in further detail large periods of time that do not show sufficient variation.

Algorithm 5.1 – Routine Tree Construction

Inputs: Database D of transactions of reduced dimensionality feature windows indexed by time, Node N representing a period $[N_{start}, N_{end}]$, $split\_threshold$ and $min\_duration$

Outputs: A Routine Tree R rooted at N and $Table$ associating nodes to patterns.

1. Apply Association Miner on transactions in D with time in $[N_{start}, N_{end}]$. Store patterns in $Table(N)$.
2. If number of patterns in $Table(N)$ is greater than $split\_threshold$ and duration of N is above $min\_duration$
 - a. Split N into equal duration nodes, N_{Left} and N_{Right} adding them as children of N in Routine Tree R
 - b. Call **Generate-Tree** with N_{Left}
 - c. Call **Generate-Tree** with N_{Right}

5.3.5 Tree Pruning

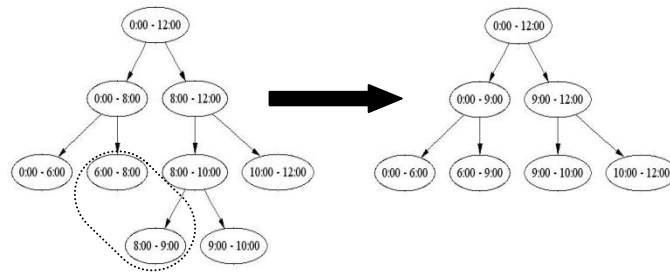


Figure 5.5 Pruning a Routine Tree –an example of a merge operation. Nodes meeting user-specified criteria can be merged to display a more compact tree.

The tree is then made compact by merging nodes where possible. If two adjacent leaves have the same maximal frequency pattern, they are merged. If they are siblings, they can be removed

from the tree. Otherwise they are merged, which involves restructuring the tree, and updating the node periods and pattern tables. Each pruning operation removes two nodes from the tree. The pruning operation is described in Algorithm 5.2. Figure 5.5 shows an example of this operation.

Algorithm 5.2 – Routine Tree Pruning

Inputs: A Routine Tree R . Function $MaxPattern(N)$ returns the maximal frequency pattern of node N , $Add(N_A, N_B)$ adds patterns from N_B to N_A and adjusts time for N_A , $Desc(N)$ returns descendants of N , and $Subtract(N_A, N_B)$ removes patterns of N_B from N_A and adjusts time for N_A , removing the node entirely if it is no longer necessary.

Outputs: A compacted Routine Tree R

3. Compute Leaf nodes L of R
4. For each pair of adjacent nodes N_i and N_j in L , if

$$MaxPattern(N_i) = MaxPattern(N_j)$$

- a. $Add(N_i, N_j)$
- b. For each N_A in R , if $N_i \in Desc(N_A)$ and $N_j \notin Desc(N_A)$, add
 $Add(N_A, N_j)$
- c. For each N_A in R , if $N_i \notin Desc(N_A)$ and $N_j \in Desc(N_A)$, add
 $Subtract(N_A, N_j)$
- d. Remove N_j from R

5.3.6 Aggregating Routine Trees

By combining routine-trees across several days, we can find a structure describing the user's typical routine. We rely on a simple aggregation method for combining multiple trees. As before, a tree is constructed, then pruned. Instead of using Closet+ in the first step to generate the pattern-table, support for patterns occurring during a period are collected from each tree and aggregated. Patterns below the minimum support are then removed. This is a somewhat simplistic procedure however, and may lose some information, for example patterns that occur

only on certain days. On the other hand, the broad, repeating features of the user's day can be captured with this simple aggregation.

5.4 Results

The method described above is applied to two datasets. The first dataset (Simulated Data) is a simulated dataset based on activity data composed to form a typical day. Data was gathered in the lab using the e-AR sensor for different activities, such as sleeping, eating, walking, different types of exercise etc. A day is simulated by concatenating these activities together in a sequence that could be recognised as typical weekdays for an office going person. The second dataset (Real Data) is obtained by asking a normal participant to wear the e-AR sensor and roughly label his activities. We present results from a third dataset where chronically ill participants were monitored over a longer period of time. The participants wore an e-AR sensor, which was supplemented with ambient sensing. In this chapter, the goal of the analysis is the visually present snapshots of the participant's routine. In later chapters we will aim to analyse routines and changes in routines quantitatively.

To obtain a quantitative measure of accuracy, we have compared the base dataset against the tree. Each transaction in the dataset is compared against the patterns stored in the tree for that time at the highest resolution. A transaction and pattern are compared by finding the size of their intersection and scaling it by the size of the larger of the two sets. For example, if the transaction is $\{0, 1, 2, 3\}$ and the pattern is $\{2, 3\}$, the match would be 0.5. Alternatively, if the transaction is $\{2\}$ the match would also be 0.5. The best match is the accuracy of the tree for that transaction. Table 5.1 shows the accuracy of the proposed method averaged over the transactions in each dataset.

We define compression factor to be the ratio of the size of the base dataset to the number of rows needed to represent the tree in a database, which includes the nodes of the tree and the patterns associated with them. Table 5.1 shows compression factors of the two datasets. Trees generated by the algorithm can be substantially smaller than the base data, by up to 12 times. Higher compression is obtained on the simulated data because it is a larger dataset, has extended periods of sleep and work that are suitable for compression, and has less variation in activity levels.

	Simulated Data					Real Data
	Day 1	Day 2	Day 3	Day 4	Day 5	
Accuracy	94.4	99.2	98.1	98.1	97.2	88.9
Compression Ratio	12.2	9.5	9.3	10.8	9.6	2.2

Table 5.1 Accuracy and compression factor for real and simulated data. The routine tree can take upto 12 times less space than the base dataset on simulated home environment data, however we have smaller gains in real data collection due to greater variation in activity patterns.

A software tool-kit has been developed for visualisation of the routine tree. The tree is drawn against a time grid. The **Leaf View** shows only the leaves as in Figure 5.7. This is useful for visualising several trees at once at the highest level of detail. The **Tree View** plots a tree level by level as in Figure 5.8. Here the levels are plotted in increasing level of detail for a single tree. The persistent pattern is shown for each node. Patterns are associated with a bar colour and height, as shown in both figures. Increasing bar size indicates a more strenuous activity. Some subjectivity is introduced here, in defining whether an activity pattern such as $\{1, 2\}$ is more strenuous than $\{0, 3\}$ or $\{0, 1, 2\}$.

Figure 5.6 shows the leaf view of the trees of the 5 simulated days along with the overall week tree. The model used to create the simulated data can be seen. We can get very quickly a bird's eye view of the user's routine. For example, typically the simulated user woke up close to 8 am, although on Friday it was much sooner. The "Week" tree is a result of combining the five trees. The morning exercise, the green bar on Tuesday, Thursday and Friday mornings, does not show in the week tree. Although it occurs three out of five days, it occurs at different times therefore is not captured. On the other hand, evening exercise is captured, because it occurs at similar times each day.



Figure 5.6 Leaf view of routines trees for simulated dataset showing the behaviour of a participant. Six trees are shown at the leaf level, in the highest level of detail.

A set of real data was collected from a 28 years old male participant, during the late-evening hours. During this time he worked, took the train home, rested and finally slept. Figure 5.7 is the tree-visualisation for this. Each phase of the user's day is distinguishable as labelled in the figure. Deeper levels show more detail of these phases. The first level merely summarises the recorded session as one of low intensity activity overall. The second level shows three phases in the day, which we can identify as work, commute, and rest at home. Further detail is visible lower in the tree, for example a period of sleep at the end of the day, a break during work, and the composition of the commute: walk to the station, stand in the train, and walk home.

Next we provide an example of a scenario very relevant for healthcare. Figure 5.8 shows the Routine-Tree for the same participant who wore the e-AR sensor overnight, during a particularly wakeful night. It is apparent that by Level-2, we can see that the user in fact did not sleep uninterrupted. Going further down breaks in sleep can be seen in further detail.

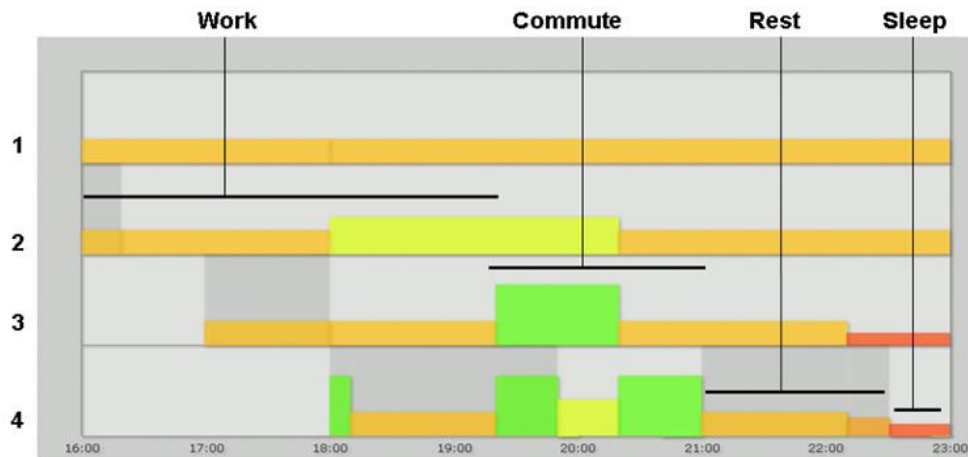


Figure 5.7 Tree view for a participant wearing the sensor during late evening hours of the day. A single tree is shown, in increasing degree of detail. The numbers on the left correspond to levels of the tree. Higher levels show broad trends, while lower levels of the tree show more detail.



Figure 5.8 Tree view for a participant wearing the sensor during a wakeful night.

In the Saphe project, participants were monitored at home, wearing the e-AR sensor during the day. These were older people, receiving long-term care. In Figure 5.9 (a) a participant (50 years old, female) wore the sensor during the day and performed normal activity. This includes increased activity in periods corresponding to meal times (breakfast and lunch, around 9 a.m.

and 2 p.m. respectively). This can be contrasted with 5.9 (b), where the activity patterns are low during lunch time. This may be interpreted in a number of ways: the participant may be asleep, may have taken the sensor off, or may be out of the house. However, as a repeated pattern, for participants requiring assistance with meals, this may provide timely and useful information. In Chapter 7 we will perform more quantitative analysis of this dataset, over longer periods of time.

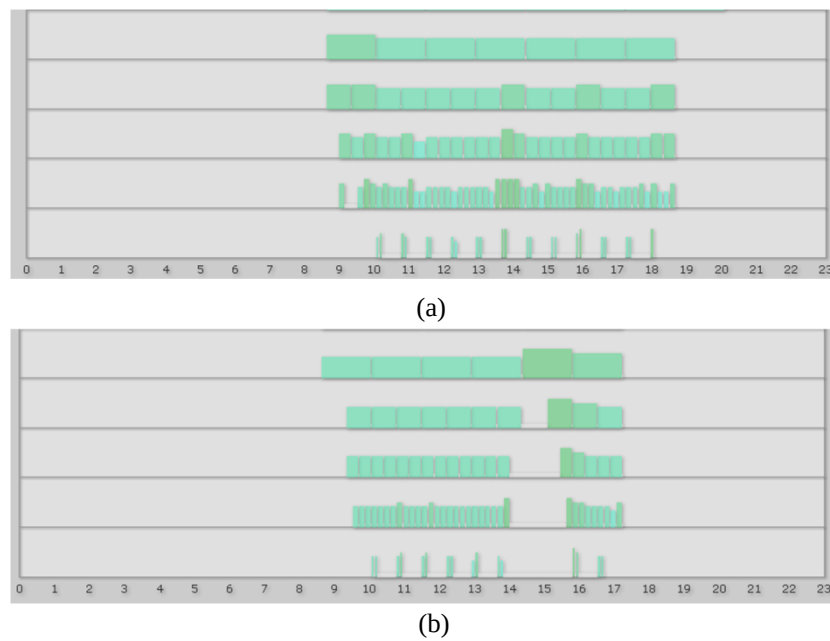


Figure 5.9 The tree view of a participant for activities over a two day period; (a) shows a day with high daytime activity, particularly between close to 14:00, while (b) shows a day where the user had low activity during the afternoon.

5.5 Conclusions

Intuitive, but information rich representations of user data are very important in pervasive healthcare, where care providers often need to care for many patients simultaneously. The Routine-Tree data structure can provide such visualisation for an important indicator of patient health. It is also a significantly compressed representation, which nevertheless represents the base dataset with good accuracy. The technical contribution of this chapter is in the new

application of a standard data mining technique hierarchically to develop a data structure reflective of the temporal structure of daily human routines.

A limitation occurs due to the formulation of windows of activity data as transactions in a Frequent Pattern Mining problem. Since transactions are sets of items, some information is lost in representing a window as a set of activity levels. For example sequence information within the window is lost, as are repetitions of an activity level. We intend to explore alternative coding methods that would minimise the loss of such information.

The top down construction of the tree ensures that data is not mined further if the number of patterns is below a threshold, implying that the user is mostly occupied with one activity. This however, entails buffering a full day's activity stream, which is not desirable for streaming applications. The goal in streaming algorithms is to process data as it arrives, or with very limited buffering. The tree can be built bottom up to utilise statistics generated at lower level nodes at the higher level. This incremental tree construction would free up memory resources required for buffering in top down tree construction, and avoid repeated execution of the pattern mining algorithm.

Algorithms other than Closet+ could also be explored for the data mining step. Frequent Pattern Mining ignores sequence and temporal information. The problem of sequence is addressed by Sequence Mining algorithms such as PrefixSpan [216]. Temporal information can be mined in a pattern mining paradigm using Inter-transaction Pattern Mining, which has been used to mine for activity patterns in [109]. That said, the complexity of Sequence Mining and Inter-transaction Pattern Mining is significantly higher. In our routine tree algorithm the specific mining algorithm can be considered a black-box, and different algorithms can be plugged in, depending on the level of detail desired in the results.

Data mining based algorithms can help in exploratory data analysis, to provide visualisations and discover information from real-world health monitoring scenarios where ground truth annotations are scarce. In chapter 6 we seek to adapt our proposed algorithms to an existing platform where non-specialised, yet effective sensors are already carried on the person of a potentially large user base: smartphones. The challenges on smartphones are similar to those of home-monitoring: a need to process streaming, unlabelled data. In addition there is a need to optimise battery usage and provide interest-sustaining feedback to users

Chapter 6

Mining Activity and Routine using Smartphones

6.1 Introduction

BODY Sensor Networks (BSN) have the potential to transform the delivery of healthcare and how personal well-being is tracked. We anticipate that traditional healthcare systems will over time incorporate activity and behaviour profiling, particularly as relationships clinical measures is concretely established. There is however, an existing base of users with body-worn sensors who can adopt pervasive sensing technology at the touch of a button: smartphone users. In this chapter we propose to adapt activity and behaviour profiling methods developed for home-based healthcare with the aim of profiling health-related parameters on smartphones.

Smartphones represent an unparalleled opportunity for sensor data mining. Not only are these devices widely used and projected to increase in usage in the coming years, they are often equipped with a wide range of sensors. For instance, the iPhone™ and most Android™ phones are equipped with one or more accelerometers, magnetic compasses and proximity sensors. They can also provide the user's location, by either triangulating from nearby cell-towers or using Global Positioning System (GPS). They may also allow audio and video recording. While users need to be motivated to incorporate specialised sensors into their everyday life, phones are on the person of users for most of their day. Their popularity is further enhanced by the

availability of rich-functionality applications, commonly termed Apps. Both iPhone™ and Android™ phones allow apps to run in the background, opening the way for applications that can collect sensor data from the phones. For instance, applications exist to record GPS data as a user exercises, calculating energy expenditure based on the speed and type of activity.

Apps that run in the background need to meet certain critical user requirements. Foremost of these is battery utilisation. Collecting sensor data can be very expensive depending on the sampling frequency and type of sensor. The most expensive sensor in this regard is GPS. The volume of data is also critical. Subsequent storage and transmission of sensor data can also be expensive, depending on the medium used for storage (internal phone memory vs. external memory cards) and the medium used for transmission (mobile internet vs. WIFI). Finally, there may be cost incurred for using the internet. The data transmitted from the phone should therefore be minimised.

The second requirement is minimal processor use. Processors of smartphones have evolved significantly, but are still limited both in terms of speed and memory. Furthermore, apps with high CPU use will consume more battery, and will slow other functionalities of the phone, resulting in poor user experience. Therefore any features extracted from the sensor data need to be inexpensively generated.

The tremendous data collection opportunity offered by smartphones is accompanied by complications in the analysis of this data. The quality and number of sensors on the phone can depend on the price of the phone. For instance, the accelerometer on the Samsung Europa™ has an operating frequency of 20 Hz, while the slightly more expensive Samsung Mini™ operates at 100 Hz. It is not possible, therefore, to assume any particular sensor type. In contrast to specialised sensors, the placement and orientation of the phone is not fixed. The phone may be on the users pockets, may be in their hands, or it may not be on their person at all. It is difficult to collect a representative dataset in controlled settings for how phones will be used in practice. Using just a supervised learning approach taken in Chapter 5 will therefore be insufficient.

Thus far, apps typically tend to require significant user management, and are targeted towards providing workout related metrics [217]. These rely on GPS, and sometimes the accelerometer with manual input of exercise type determining outputs such as calorie expenditure. Research in activity recognition using mobile phones is growing, typically using supervised learning to recognizing specific activities [218-220]. There has been research [221] in developing profiles

that map accelerometer data from phones to calorie expenditure. This work specifically looked at phone placement in order to adjust the mapping functions, with a belt worn holder suggested for the most accurate calorie expenditure output. A representative approach is taken by [222] to train an SVM to recognise seven activities. A larger range of phone orientations is considered by Henpraserttae et al. [218], who conclude that location mapping models need to be trained to achieve satisfactory accuracy.

In this chapter, we describe activity and routine analysis of data collected from smartphones. Accelerometer sensor, orientation sensor and location data is collected. It is partially processed on the phone itself to reduce the volume of the data transmitted. We describe unsupervised methods for extracting activity from the general sensor data, while training classifiers to recognise specific activities. The routine-tree presented in Chapter 5 is adapted to handle both location and activity. Finally results are presented for a participant wearing the smartphones over sixty days.

Our contributions include:

- Development of a software system for low-cost collection of sensor data with a view towards minimizing the data transmitted by shifting processing onto the phone.
- Application of supervised and unsupervised methods to sensor data to generate specific classifications and more abstract clustering that doesn't require annotation/training.
- Recognition and profiling activity over the long-term without specification of location of phone on the participant.
- Incorporation of GPS and accelerometer for a combined analysis.

6.2 ActiveMiles

Research [223, 224] has shown that reducing known lifestyle risk factors could prevent the dominant sources of morbidity and mortality, particularly in the developed world, such as heart disease and diabetes. Increasing activity is important for managing obesity [225], as well as for reducing risk factors for many diseases including heart disease [226], type-II diabetes [227], certain types of cancer [228]. Where low activity lifestyles are likely to lead to health problems, encouraging higher activity is a recommended intervention [229]. We have developed the smartphone-based ActiveMiles application, which is designed to encourage users to be active by

tracking their activity over time and competing for points by incorporating exercise into their lifestyle. This Android™ based application that executes in the background, and records accelerometer, orientation and location sensor data. Feedback to users is provided in the form of ‘Active Miles’ points, which correspond to how much active they have been. The software has been designed to present information intuitively, while also recording more complex features from the data for deeper analysis. Care has been taken to ensure the reliability and power-efficiency of the system. Figure 6.1 shows an example of feedback provided to users, where their daily, weekly and monthly ‘Active Miles’ are compared against other users, and a designated competitor, termed ‘Nemesis’.

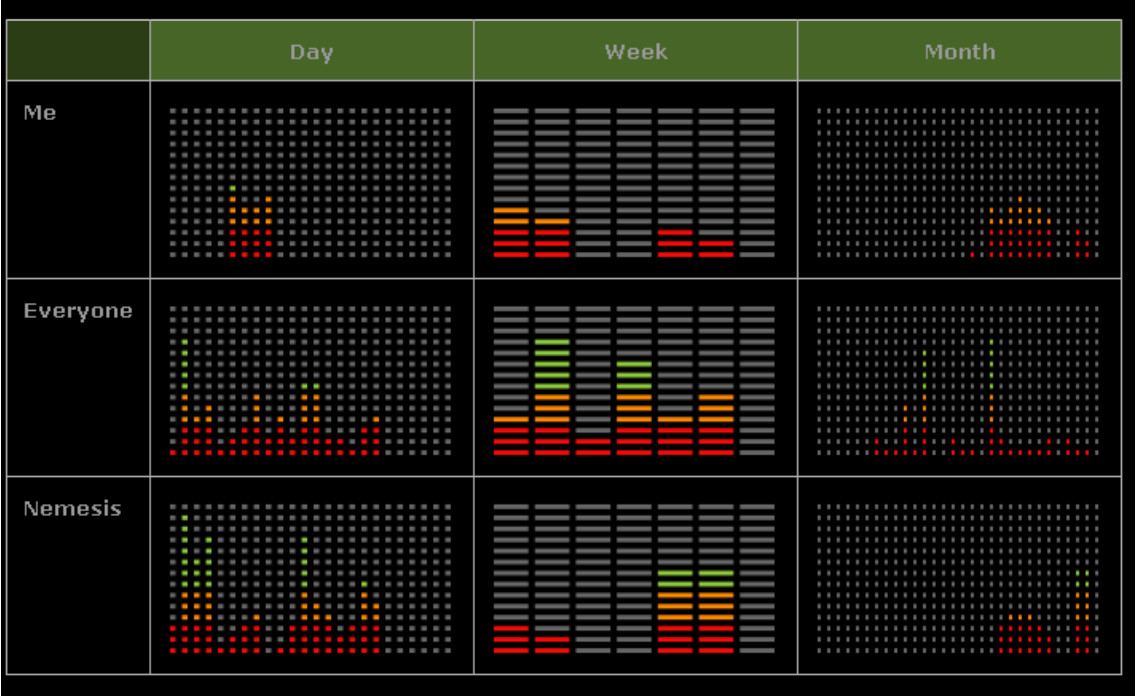


Figure 6.1 Activity feedback shown to users of the Active Miles application: daily, weekly and monthly activity levels are compared against other users of the system, and a designated competitor termed Nemesis. Day plots are over 24 hours, with each bar corresponding to an hour. Week and Month plots are over 7 and 30 days respective, with each bar corresponding to a day. The goal of the system is to encourage users to increase the points they win by being more active.

Figure 6.2 shows accelerometer, orientation and location sensor data for a user during an exercise session. Location can be estimated using GPS or cell-tower triangulation. It can be seen that the cell-tower based triangulation 6.2 (b) is much coarser resolution than the GPS location 6.2 (a), and is prone to errors. To use location for long-term mining however, it is important to

use this inexpensive reading, instead of the more precise GPS location, as it is significantly lower in battery consumption.

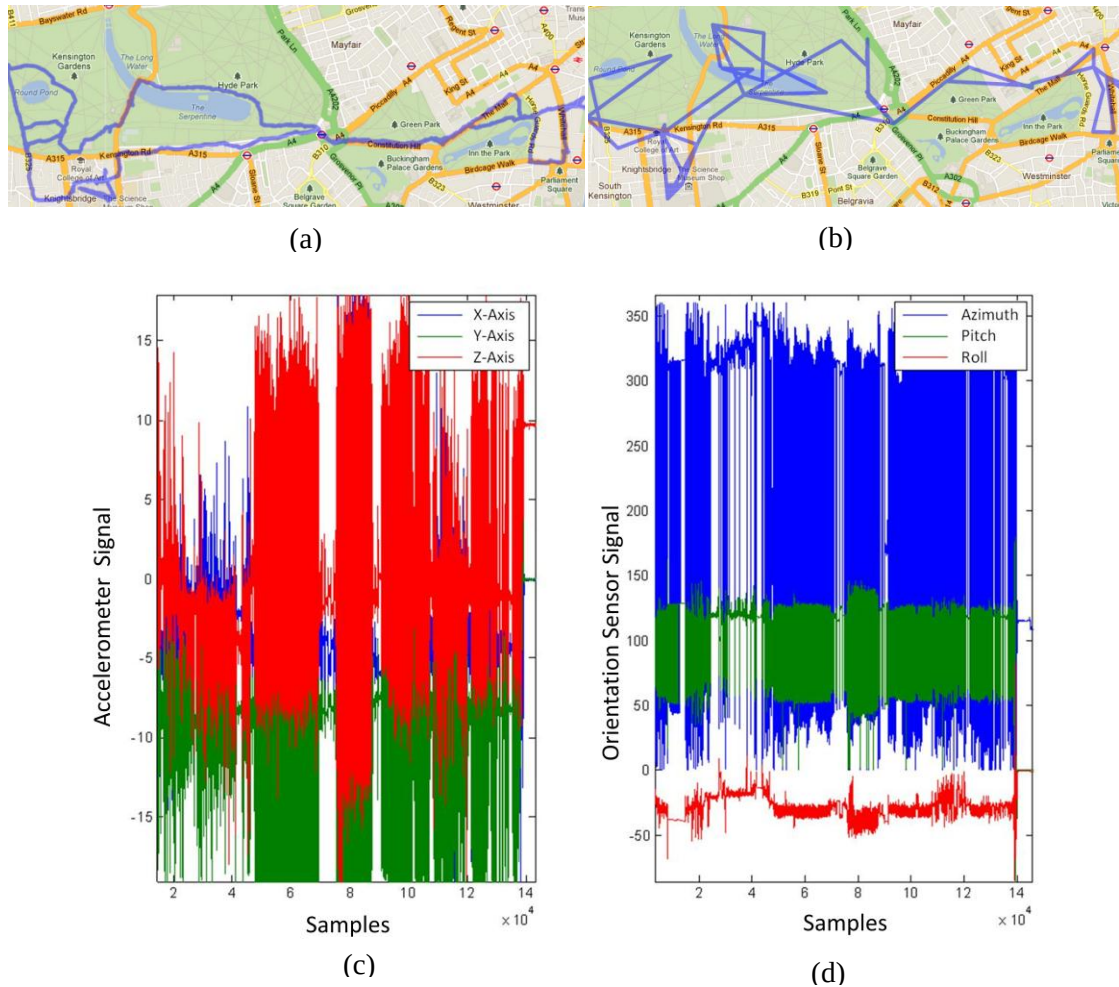


Figure 6.2Data collected from the Android™ based Active Miles application: GPS based location (a), Cell-tower triangulation based location (b), Accelerometer sensor (c) and Orientation sensor (d) . A Samsung Galaxy Mini™ phone was used for this circuit

Data from the phone's accelerometer is similar to the data collected from the e-AR sensor in Chapters 4 and 5. The orientation sensor has three channels: Pitch, Roll and Azimuth. These channels correspond to the orientation of the phone around the X, Y and Z axes respectively.

6.3 Overview of Methodology

The ActiveMiles application is designed to be deployed using the Android Marketplace™, enabling anyone with the smartphone to download the application and start streaming data to the server. This presents significant analysis challenges, amongst which is the lack of control over variables such as phone type, the user's manner and extent of phone usage. Supervised learning techniques can be powerful for activity detection, however given the immense variability in participant and phone types, it is necessary to consider unsupervised, clustering based techniques. As in Chapter 4, it is important to maximise what can be learnt from the structure of the data itself, and perform analysis in the intrinsic dimensions of the data. In addition, to reduce the load on the server, and to minimise the amount of data transferred, it is far more feasible to transmit data in reduced dimensionality, instead of the full feature space. Technically, the proposed framework consists of the following main steps:

- 1) Feature Extraction: Features are extracted from fixed size windows of data.
- 2) Dimensionality Reduction: Feature space data is embedded into a reference manifold reflecting its intrinsic dimensionality.
- 3) Activity Analysis: Depending on the application, manifold embedded data can be analysed using supervised and unsupervised learning methods. If class labels are available then classification algorithms can be used for activity categorisation and analysis of specific activities. In the absence of class labels, we propose using clustering algorithms, with results presented for two well-known algorithms.
- 4) Routine Mining: The activity labels thus generated can be mined by adapting the algorithm proposed in Chapter 5 to incorporate location. Two approaches for this are proposed in this chapter.

Steps 1-3 can be performed on the phone itself, reducing the volume of data transmitted significantly at each step. There is a trade-off, however, between reducing network and CPU usage, and it may be feasible to perform only the first or second steps on the phone.

6.4 Feature Extraction

Feature extraction from the accelerometer and orientation sensors consists of the extraction of a set of features from each channel of the sensors. A moving window of data over two seconds is used. For a three-axis accelerometer and a 3-axis orientation sensor, we extract 9 features in total, listed in Table 5.1. This is a smaller set of features compared to those extracted for the transitional activity recognition. This is due to the design choice of feature extraction on the phone, which limits the complexity of the features that can be extracted without imposing infeasible computational or memory costs. The benefit, in addition to saving on network utilisation (and therefore battery usage), is that we can perform the analysis at high sampling rates. Android™ phones allow sampling at increasingly high rates, with the mid-range Samsung Mini capable of sampling at a maximum of 100 Hz. The most common use case scenario will not permit streaming at such high sampling rates. Our system samples at 100 Hz, but streams features at 0.5 Hz. The mean and variance is extracted from each channel of the accelerometer, with the mean of the magnetic compass features giving the orientation of the phone.

Feature Number	Sensor	Channel Feature
1	Accelerometer	X-Axis Mean
2	Accelerometer	X-Axis Variance
3	Accelerometer	Y-Axis Mean
4	Accelerometer	Y-Axis Variance
5	Accelerometer	Z-Axis Mean
6	Accelerometer	Z-Axis Variance
7	Magnetic Compass	Azimuth Mean
8	Magnetic Compass	Pitch Mean
9	Magnetic Compass	Roll Mean

Table 6.1 Features extracted locally from Accelerometer and Orientation sensors. Six features are extracted from the accelerometer and three features are extracted from the orientation sensor.

Figure 6.3 shows the features extracted from a user who is initially walking, then sits down. For both accelerometer and orientation sensors, there is a significant change when the participant transitions from walking to sitting down. As can be expected, not all features are equally useful for a given activity. For instance, the Azimuth variance is not correlated with change in activity in this instance. To perform subsequent analysis with features in the intrinsic dimensionality of

the dataset, and to potentially further reduce the volume of data transmitted, data is embedded into manifold space.

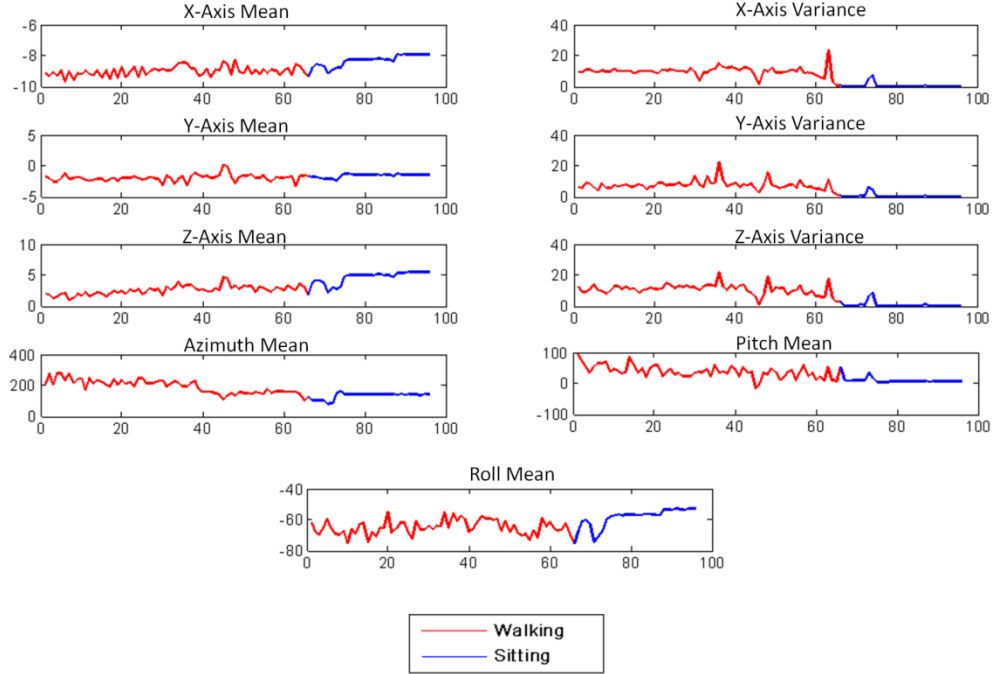


Figure 6.3 Features extracted from accelerometer and orientation sensors, plotted against window index. Each window corresponds to 2 seconds of sensor data. User is initially walking, and then proceeds to sit down.

6.5 Dimensionality Reduction using Manifold Embedding

The advantages of dimensionality reduction for activity classification were described in Section 4.5. In addition to potential gains in classification or clustering accuracy, the dimensionality reduction step can be shifted onto the phone, reducing the volume of the data transmitted several fold. Figure 6.4 shows an Isomap representation of the data shown in 6.2 in two dimensions. While the transmission of features remains at 0.5 Hz, the dimensionality of this data can be decreased by several-fold. In the results section for instance, we demonstrate that it is possible to perform accurate classification with three dimensions, leading to a three-fold decrease in the data transmitted. As described in Chapter 4, a reference manifold can be learned, with regression models to map new points to the manifold.

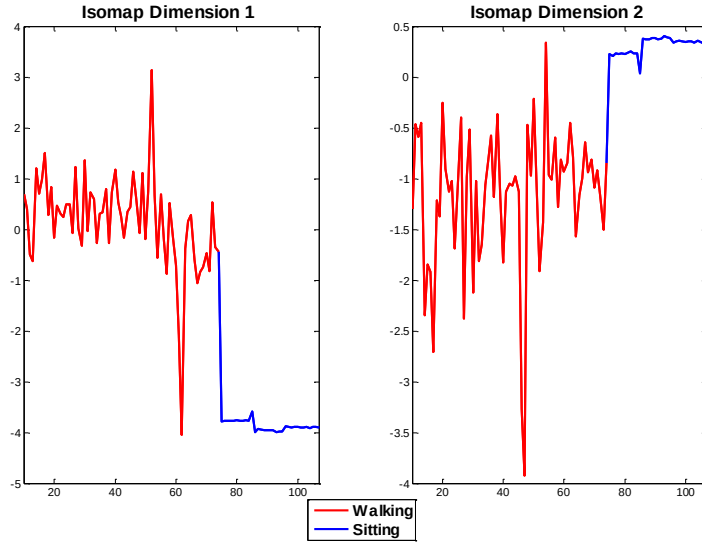


Figure 6.4 First two dimensions of Isomap reduction showing participant transition from Walking to Sitting activities

6.6 Activity Analysis

We consider two use-case scenarios for the activity analysis step. In the first instance, the purpose may be to develop customised deployments for specific applications. For instance, for clinical trials relating to exercise, it may be possible to collect labelled data for the activities of interest before deployment. In this case, a supervised learning approach will be suitable. We will present results using standard classifiers for the categorisation of commonly performed activities and exercises.

The second use case is where we wish to extract activity labels from dimensionality reduced data without any labelled training data. For this case we learn a clustering model. Clustering aims to partition a dataset into k points, in which each point belongs to the cluster with the nearest mean. We assume here that each cluster corresponds to a distinct activity region on the manifold. There exist several well-studied choices for the clustering step. We will compare results for K-means and Gaussian Mixture Model (GMM) based clustering using the Expectation Maximisation (EM) algorithm.

The parameter k specifying the number of clusters in the data can be fixed beforehand if the number of activity types is known, or it can be chosen using cluster size selection heuristics, for instance the Gap statistic [230].

It was noted in [115] that the variances of the accelerometer channels can be used to measure the intensity of activity. For the purposes of presentation, clusters will be ordered by the average activity intensity; i.e. by the average variance of points within the cluster.

6.6.1 K-means Clustering

Given a set of points $x_1, x_2, \dots, x_n$ k-means clustering [231] finds a set of k partitions such that the sum of squared distances of each point from the centre of its assigned cluster. The objective function therefore is to produce a set of partitions s such that

$$\sum_{i=1}^k \sum_{x_j \in s_i} \|x_j - \mu_i\|^2 \quad (6.1)$$

is minimised, where μ_i is the cluster centre of segment s_i .

The objective function can be optimised by choosing cluster centres randomly initially, then running an iterative procedure where in each iteration distances to cluster centres are updated, and cluster centres are recomputed. The algorithm is terminated when the assignments no longer change.

6.6.2 Gaussian Mixture Model based Clustering

Mixture models define a set of probability distribution functions that is likely to generate the dataset. Gaussian Mixture Models (GMM) [232] model the mixture models using Gaussian functions, where each component Gaussian corresponds to a cluster in the data. Formally, a Gaussian mixture model is a weighted sum of k component Gaussian densities as given by the equation

$$p(x | \theta) = \sum_{i=1}^k w_i g(x | \mu_i, \sigma_i) \quad (6.2)$$

where w_i is the weight assigned to the mixture i and, and given mean μ and variance σ $g(x | \mu_i, \sigma_i)$ is a component Gaussian function of the form

$$g(x | \mu_i, \sigma_i) = \frac{1}{(\sqrt{2\pi}\sigma_i)^D} \exp\left\{-\frac{1}{2}\left(\frac{\|x - \mu_i\|}{\sigma_i}\right)^2\right\} \quad (6.3)$$

The well known Expectation Maximisation (EM) algorithm [233] can be used to find the maximum likelihood estimate of the parameter vector θ .

6.7 Incorporating Location into Routine Tree

In chapter 5 we proposed a technique for abstracting a multi-resolution data structure that associates temporal periods with patterns of activity. This is referred to as routine tree. The application of the routine tree to the activity clusters instead of activity levels is straightforward. In this section we consider approaches for the incorporation of location into the analysis.

6.7.1 Location as Feature

One approach, as taken by Atallah et al. [50] is to regard location as providing context for a more precise or more accurate activity level estimation. For instance, the sitting activity at home and sitting at work could correspond to two different labels in the system. There is a significant difficulty to this when applied to unsupervised analysis of phone data. While activity values can be envisioned to belong to a small number of clusters, it is difficult to determine apriori the locations in which a user is likely to be in. Particularly in the case of GPS or cell-tower based location, the number of locations the user can be in can be very large, with most of these readings being transitional and thereby not useful for context.

To provide meaningful location context, it is useful to add more semantic content to the location measure. Instead of raw longitude, latitude readings for instance, Google Latitude™ outputs location history with more descriptive labels like ‘work’, ‘home’, etc. In Google Latitude [234]

and other recent work [235] these labels are derived from temporal models of user routines. This is infeasible, as instead of imposing a model of routine on users of the system, the goal is to discover it. A simpler approach, as taken by Ashbrook and Starner [236] is to use a clustering algorithm to reduce the longitude and latitude into locations. This requires the selection of a cluster number however, which is infeasible when considering location data which may change significantly. An algorithm based on minimum occupancy time and minimum duration is proposed by Khetarpaul et al. [237], that selects ‘Interesting Locations’ without specifying a fixed number of locations. Algorithm 6.1 describes a simplified version of the procedure based on [237] to select ‘StayPoints’.

Algorithm 6.1 – Stay Point Calculation

Inputs: Longitude, Latitude based locations L , distance threshold D , time threshold t , predicate $dist$ returning distance between two locations, $duration$ returning the time difference between the first and last element of a period P and $Center$ returning the central point of a set of locations.

Outputs: A set of locations s

1. Scan L to select periods P such that for every L_i, L_j in P such that

$$j = i + 1 \quad dist(L_i, L_j) < D \quad \text{and} \quad duration(P) > t$$
2. For each P , $S = S \cup Center(P)$

The parameters of minimum duration and minimum distance still need to be specified. This can be determined empirically, and based on the constraints of the sensors. For instance, GPS readings can be of high resolution, in contrast to cell-tower based localisations, as shown in Figure 6.1. As cell-tower based localisations are coarse grain to begin with, fewer readings will be close enough to come within a reasonable distance threshold to specify the same location.

6.7.2 Location as Dimension

Phenomenologists such as Heidegger, consider temporality and spatiality as foundational relations for human consciousness [238]. Time is incorporated in the routine tree algorithm by associating patterns of activity with periods of time. Space can be similarly considered a second

dimension of routine, instead of a feature of activity. In the database representation for location as feature the table is indexed by time, and location is appended to activity features as a categorical feature. When location is a dimension, location and time are combined to index activity features. This results in associations of activity patterns with locations and time, instead of just time. Instead of a data structure divided only in terms of time granularity, we seek to extract separate pattern statistics for each location and time granularity. The algorithms for routine tree construction and pruning can be modified by recursing along ‘space’ as well as time, producing a two-dimensional grid structure at each level. Algorithm 6.2 shows the modification proposed for two-dimensional routine-tree construction. It is challenging to produce visualisations for such a grid structure, therefore the preferred method will be to consider location as a feature.

Algorithm 6.2 – Routine Tree (with Location) Construction

Inputs: Database D of transactions of reduced dimensionality feature windows indexed by time and location L , location threshold parameter α Node N representing a period $[N_{Start}, N_{End}]$, $split\_threshold$ and $min\_duration$

Outputs: A set of trees R rooted at N and $Table$ associating nodes to patterns.

1. For each L_i in L
 - a. Apply Association Miner on transactions in D with location within α of L_i and time in $[N_{Start}, N_{End}]$. Store patterns in $Table(N, L)$.
 - b. If number of patterns in $Table(N, L)$ is greater than $split\_threshold$ and duration of N is above $min\_duration$
 - c. Split N into equal duration nodes, N_{Left} and N_{Right} adding them as children of (N, L) in Routine Tree R
 - d. Call **Generate-Tree** with N_{Left}
 - e. Call **Generate-Tree** with N_{Right}

6.8 Results

We present our results from data collected from a single participant (a 30 year old male) over a period of seventy days. A diary was kept for parts of two days comprising approximately four hours, listing the activities performed. This is referred to as Study I, which we will analyse using supervised and unsupervised learning methods to generate activity labels. We will demonstrate the feasibility of training classifiers for specific activities, and relate the clustering algorithm's output to the output of supervised classification. The full dataset, referred to as Study II, will be analysed for routine by mining activity labels and location. The sampling frequency of the sensors is not fixed and is driven by events generated by the operating system. Table 6.2 lists the mean and standard deviation of the data collected from each sensor, while Table 6.3 describes the activities annotated in the diary for Study I, and the duration for which the label was placed. The running activity was performed on a treadmill at three different speeds. The walking activity was performed outdoors, and on a treadmill at a fixed speed of 5.5 km/h.

No restrictions were placed on the orientation or the placement of the phone during data collection. A significant segment of the data may therefore be from periods where the phone is not on the person of the participant. How the phone is carried influences the sensor readings obtained for the same activity. Data for the walking activity from the accelerometer is shown when the participant holds the phone in his jacket in Figure 6.5 (a) compared to when the phone is in the front pocket of the participant's trouser in Figure 6.5 (b). Signal features such as the mean and variance are significantly different for the same activity because of how the phone is carried.

Sensor Type	Mean	Standard Deviation
Accelerometer	20.8 Hz	1.71 Hz
Orientation	5.05 Hz	0.44 Hz
Cell Based Location	0.02 Hz	0.002 Hz

Table 6.2 Mean and Standard Deviation of sampling frequency from sensors on the phone. The phone operating system attempts to deliver sensor data on a 'best effort' basis, and does not guarantee constant sampling frequencies.

Activity Number	Annotation	Duration (in seconds)
1	Phone not on person	967
2	Sitting Down	523
3	Lying Down	44
4	Standing	33
5	Walking Upstairs	82
6	Walking (Outside)	512
7	Walking(Treadmill)	232
8	Running	1058
9	Cycling	126
10	Rowing (Machine)	306

Table 6.3 Activities corresponding to annotations placed by smartphone user, along with the duration of the label.

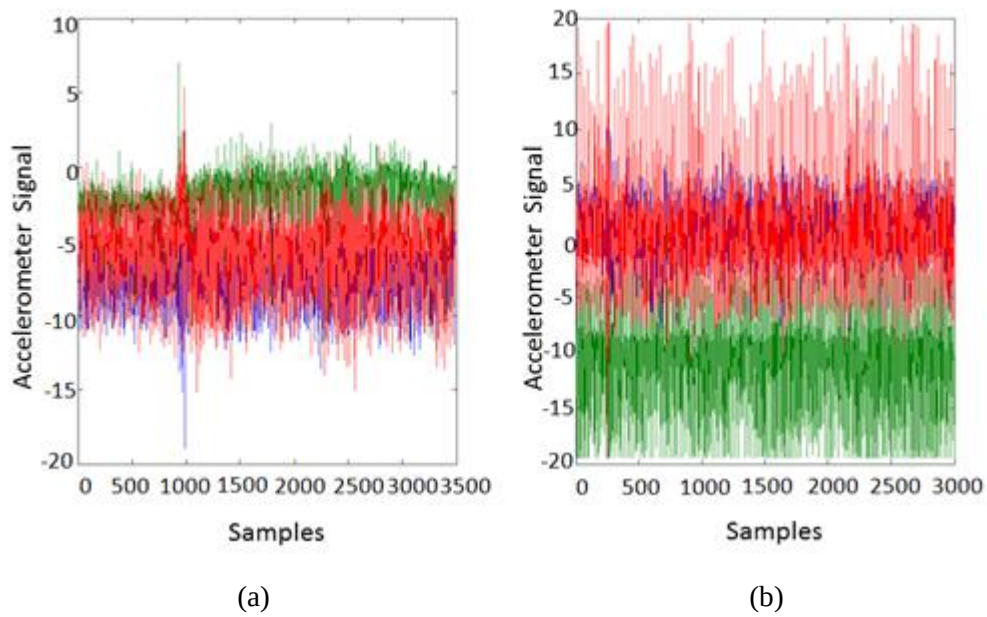


Figure 6.5 Accelerometer data showing the walking activity when the phone is in the jacket (a) and trouser front-pocket (b). The signal features can be significantly different for the same activity depending on the position of the data.

6.8.1 Analysis of Study I - Labelled Activity Dataset

A reference manifold was trained from 500 points selected using the MaxMin method proposed in [239]. Figure 6.6 shows the visually separable clusters of activities performed in Study I when the data is embedded in this reference manifold. The walking outside activity has the most variability, as this was collected with the participant moving the phone (after talking on it). This

suggests that the reference dataset must be representative of the diverse ways in which mobile phones are used and carried, if a supervised classifier is to be trained on this data.

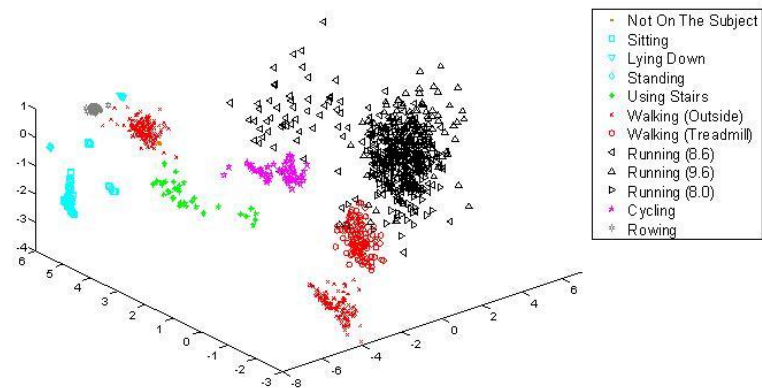


Figure 6.6 First three dimensions of Isomap embedding of activities performed in Study I. Data corresponding to each activity clusters together, but may belong in multiple clusters depending on phone location, as in the walking activity.

The first five dimensions of the manifold were empirically found sufficient to represent the intrinsic dimensionality, as shown in Figure 6.7, where the arrow shows the ‘elbow’ of the residual variance curve.

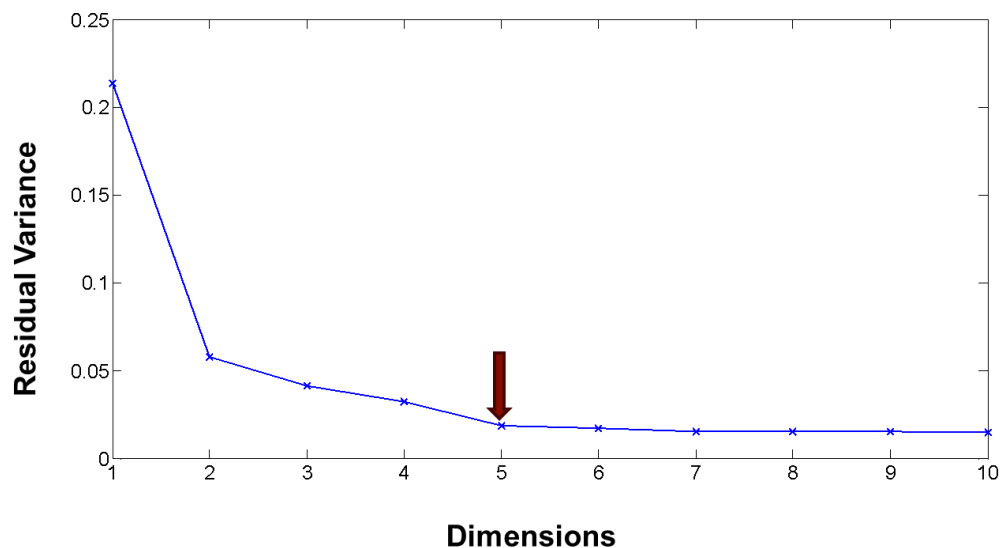


Figure 6.7 Using Residual Variance to estimate the intrinsic dimensionality of activity data. The arrow shows the ‘elbow’ of the residual variance curve. This gives an estimate of the dataset’s true dimensionality.

	C4.5	Neural Network	Naive Bayes
Accuracy	97.4	97.0	95.5
Precision	97.4	97.4	95.8
Recall	97.5	97.0	95.5

Table 6.4 Accuracy of Activity Detection for Study I using first five dimensions of Isomap embedding.

Given class labels, we can train participant specific classifiers to detect activities. Table 6.4 shows the accuracy of classifiers trained with data embedded in the manifold of its intrinsic dimensionality. Naive Bayes, Neural Network and Decision Tree classifiers were trained and ten-fold cross validation was used to compute classification accuracy. All three classifiers achieve high classification accuracy. It is possible to perform this classification with fewer dimensions, without significantly reducing accuracy. The plot of classifier accuracy versus manifold dimensionality in Figure 6.8 demonstrates that adding dimensions beyond the first three does not significantly increase the classification accuracy. There is a significant advantage to using fewer dimensions in the classification task.

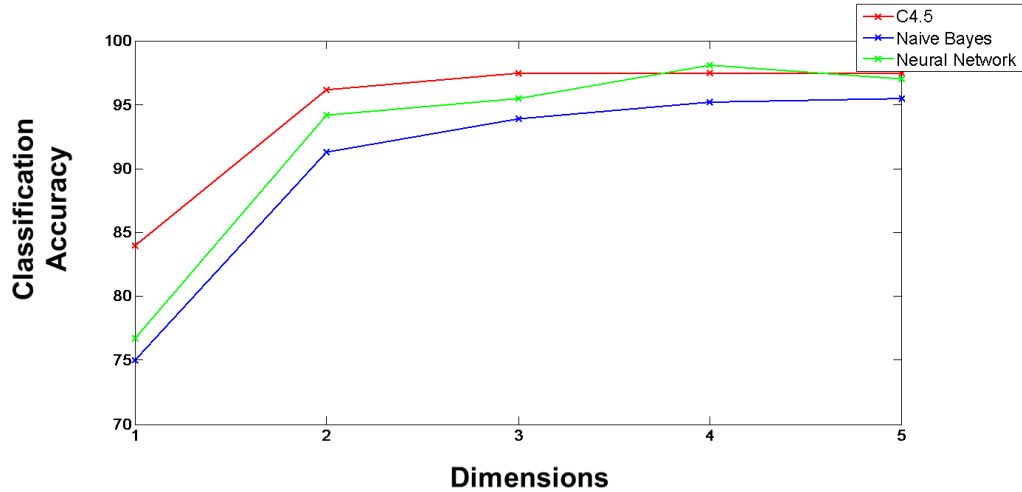


Figure 6.8 Accuracy of Classifiers for Activity Detection versus Manifold Dimensionality. There is diminishing impact of dimensionality on classification accuracy after three dimensions.

As in Chapter 4, we can train reference manifolds for specific activities of interest. For instance, Figure 6.9 shows the manifold embedding of the running activity using the accelerometer data features. The three running speeds cluster in identifiable regions.

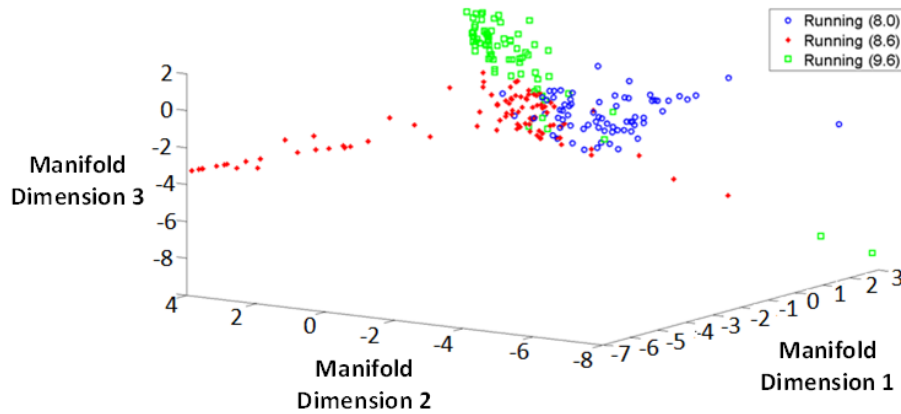


Figure 6.9 Isomap embedding of accelerometer data for the running activity. Clusters correspond to running speed.

Generalised Regression Neural Networks (GRNN) were introduced in Chapter 4 to learn regression classifiers to map from feature to manifold space. Here we propose to use GRNN to estimate running speed using just the accelerometer features. This has a significant advantage over current exercise apps, most of which estimate running speed using GPS. GPS reception is poor indoors. Furthermore, running in place, as on treadmills, does not change the location of the participant, and therefore the application will fail to detect running. As Figure 6.10 shows, the running speed estimated using GRNN approximates the speed as recorded by the participant on the treadmill. While the treadmill moves at a constant speed, the strides of the participant can be variable (referred to as kinematic variability). This can explain the spread of the estimated speed. As Figure 6.9 shows however, the patterns at 9.6 km/h are significantly different from the other two speeds. This results in stronger correspondence of the estimated speed with the treadmill belt speed.

The choice of clustering algorithm can be made based on the performance of the algorithm against labelled data. Figure 6.11 shows the activity labels and the cluster labels generated by the K-means and GMM algorithms. There is noticeably better correspondence to the labels for the GMM clustering results than the K-means clustering results.

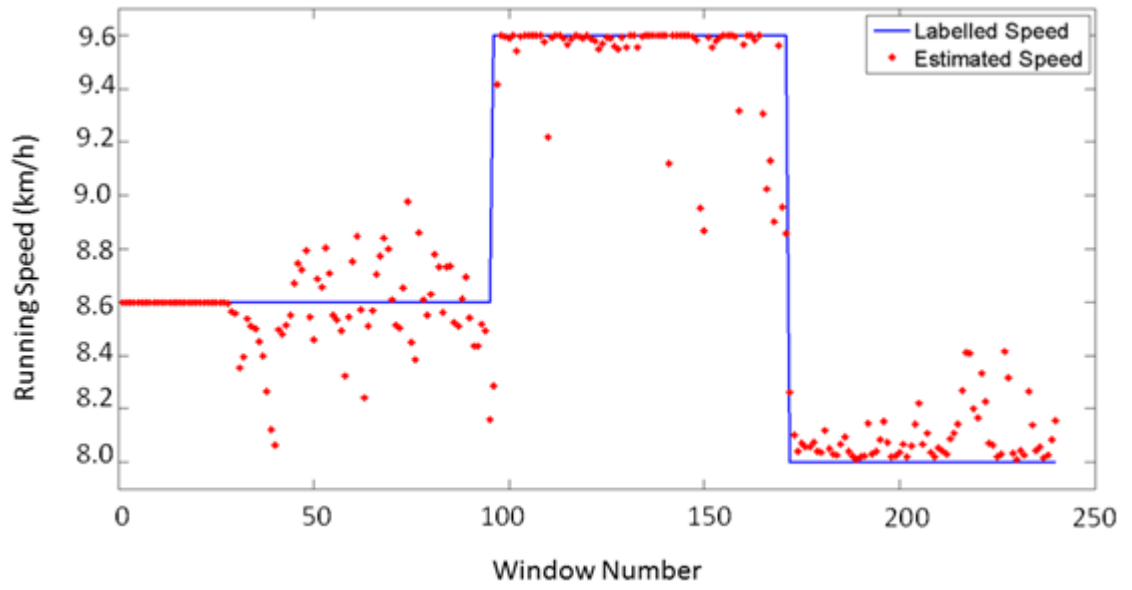


Figure 6.10 Running speed (in km/h) as labelled and as estimated using a GRNN regression classifier.

An objective measure of the correspondence between activity labels and clustering can be obtained using the Adjusted Rand Index (ARI) [240]. Given two partitions U and v of n elements, there are $\binom{n}{2}$ pairs of elements. Let a , b , c and d be defined as the number of pairs of elements such that:

- a - Elements in a pair are placed in the same set in U and in the same set in v
- b - Elements in a pair are placed in the same set in U and different sets in v
- c - Elements in a pair are placed in the different sets in U and same sets in v
- d - Elements in a pair are placed in different sets in U and different sets in v

The Rand Index (RI) is defined as:

$$RI = \frac{a + d}{a + b + c + d} \quad (6.4)$$

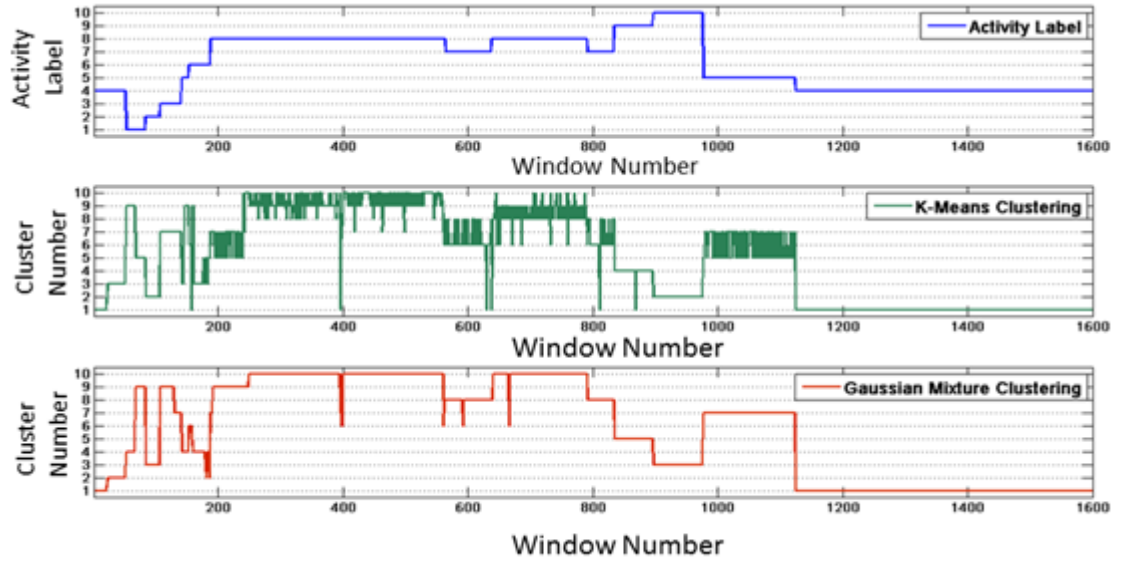


Figure 6.11 Manually placed activity labels versus cluster labels generated by K-Means and GMM Algorithms for Study I. GMM outperforms the K-Means when compared against activity labels.

RI takes values between 0 and 1, with 1 denoting complete correspondence, and 0 denoting complete disagreement. A limitation however, is that the expected RI of random partitions does not take a constant value. The corrected for chance measure, called Adjusted Rand Index (ARI), can be computed as follows:

$$ARI = \frac{\binom{n}{2} (a + d) - [(a + b)(a + c) + (c + d)(b + d)]}{\binom{n}{2} - [(a + b)(a + c) + (c + d)(b + d)]} \quad (6.5)$$

As in RI, ARI ranges from 0 to 1, with 1 denoting complete agreement. Table 6.5 shows the ARI values of the labels generated by the two algorithms against the activity labels and each other. GMM outperforms the K-means algorithm significantly. One reason for this is that K-means tends to produce ‘round’, equal-sized clusters [241] which may not correspond to the actual shape of the clusters in the data. As Table 6.3 demonstrates, there is a significant variation in the activity lengths and types. The application of k-means generally results in one or more activities being split into multiple clusters. Due to the superior performance of GMM clustering, it will be used in our subsequent analysis to generate activity labels.

	Activity Label	K-Means	Gaussian Mixture
Activity Label	1	0.63	0.87
K-Means	0.63	1	0.72
Gaussian Mixture	0.87	0.72	1

Table 6.5 Comparison of Clustering Algorithms with Ground Truth using the Adjusted Rand Index (ARI).
GMM performs better than K-Means clustering and is used in the subsequent analysis.

6.8.2 Analysis of Study II – Routine Mining of Historical Activity Data

The analysis described so far can be considered as ‘online processing’: It can be performed as the data arrives, and can even be shifted onto the phone itself. In this section we present results for the server-side mining of data collected for 60 days. We assume that labels are unavailable, although we will be showing results for some labelled data for validation.

As before, data is embedded into the manifold space, with 1000 landmark points selected using the MaxMin method to train the manifold. A GMM was trained on the manifold using a cluster size of 10. Figure 6.12 (a) shows the cluster sizes for the clusters and 6.12 (b) shows the average activity intensity in each cluster. It can be seen that clusters 1-4 correspond to very low-intensity activities, cluster 5 to low, 6-7 to medium, 8-9 to high and 10 to very high intensity activities.

Instead of setting the number of clusters, the number of clusters can be selected using the Gap statistic. It is based on the within-cluster dispersion error, which decreases monotonically as the number of clusters increases. The gap statistic seeks to find the point at which this decrease flattens significantly. Computation of the statistic involves sampling from a uniform distribution. Furthermore the EM algorithm’s output depends on its starting points, which are selected randomly. The analysis was therefore repeated 50 times. Figure 6.13 shows k versus the number of times it is selected by the Gap statistic. This suggests that $k = 3$ will produce optimal clusters, although $k = 5$ is also suitable, as this offers greater specificity. The cluster size suggested by the gap statistic is necessarily dependent on the sample size used for training the algorithms. In the case of a wide-scale deployment however, it is difficult to determine what the appropriate sample is, and therefore we may need to retrain the clustering algorithms. In this

case, it may be simpler for subsequent analysis and visualisation to choose a fixed number of clusters, simplifying the mining procedure further down the analysis pipeline instead of needing to adjust based on variable number of clusters.

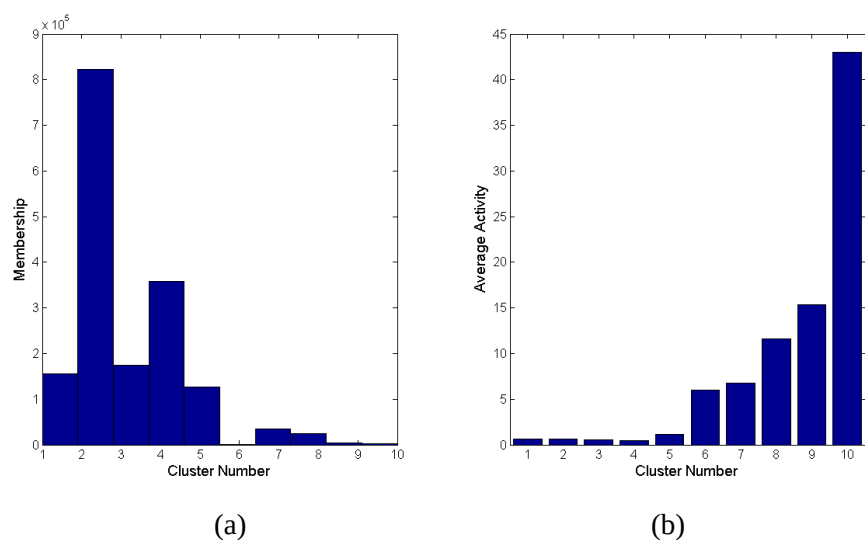


Figure 6.12 Membership sizes of GMM clustering for Study II (a) and average activity intensity for each cluster (b).

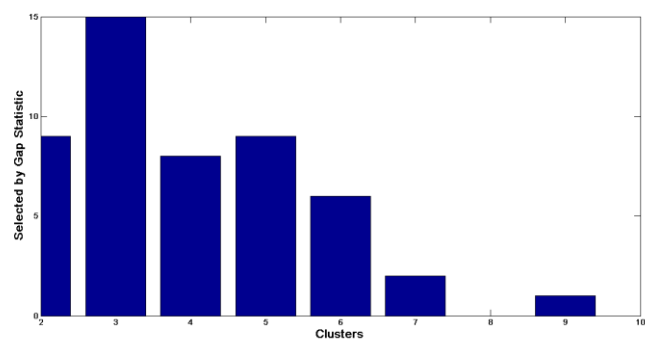


Figure 6.13 Membership sizes of GMM clustering for Study II (a) and average activity intensity for each cluster (b).

Cluster labels produced by the GMM shown in Figure 6.14 are used for generating Routine Trees. Each transaction in the database consists of activity labels over twenty seconds. Figure

6.15 shows the Routine Tree for a day in this database involving office-activity. Periods between 11:00 and 17:00 have high and medium intensity activity patterns. On this day, the participant participates in weekly meetings at his research group, between 11:00 and 13:00. The lower-levels of the tree shows the participant's arrival at the meeting and activity during it.

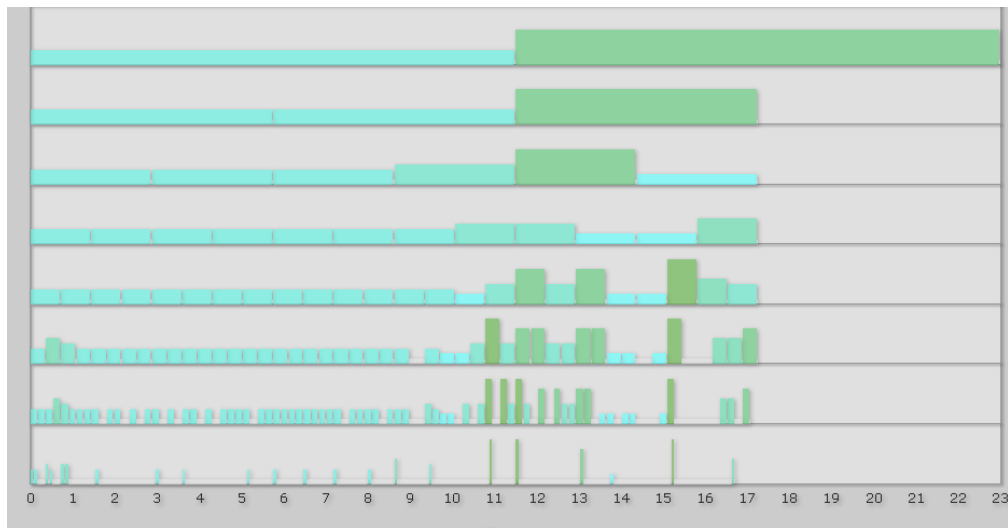


Figure 6.14 Routine Tree of a Participant showing office-activity, including a weekly meeting

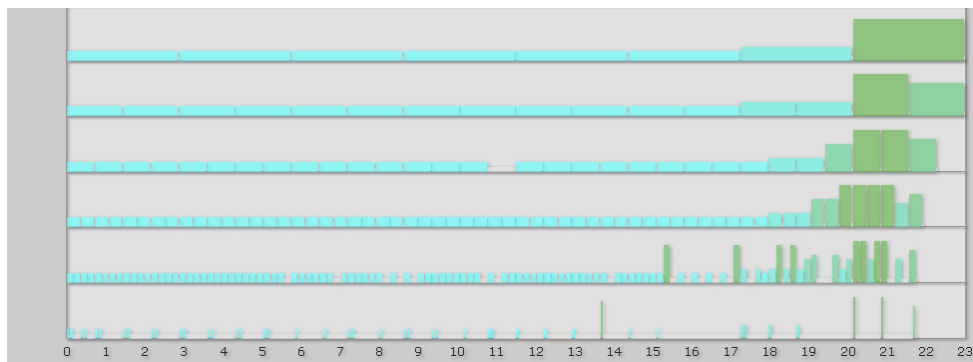


Figure 6.15 Routine Tree of a Participant showing holiday activity, with low activity in the daytime and high-intensity activity patterns in the evening.

Figure 6.14 can be contrasted with the participant's behaviour during an off-day, as in Figure 6.15. In this case the participant has low activity for most of the day. However, in the evening

there are activity patterns associated with high intensity activities. This would suggest activities such as socializing. To be certain however, we must incorporate location into the analysis.

Over the course of three months, the participant was located at 4262 distinct locations. Only a small subset of these is significant however, as shown in Figure 6.16, which shows the average time the participant was present at each location in hours. It can be observed that within the top twenty activities, the occupancy duration reduces to less than 10 minutes. This can be used to select the threshold on the number of ‘significant’ locations, or the minimum duration of occupancy for a location to be considered significant. This is our time threshold in the Stay Points algorithm.

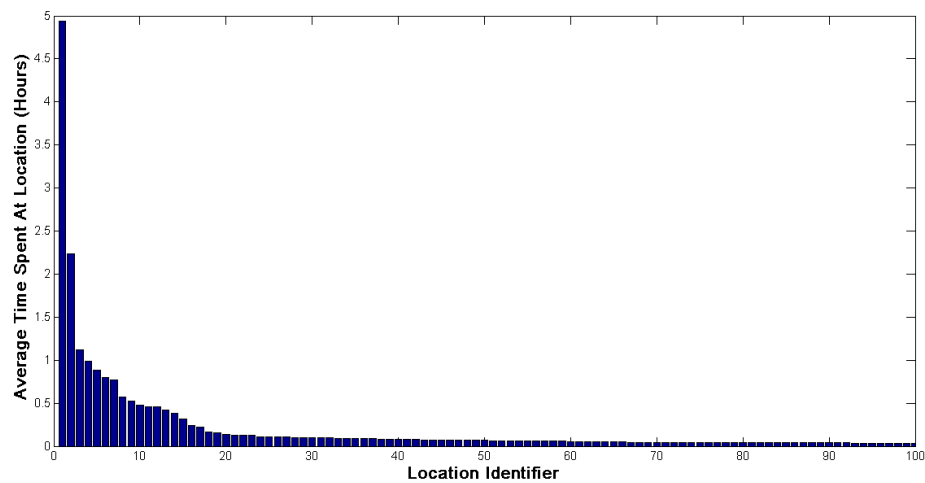
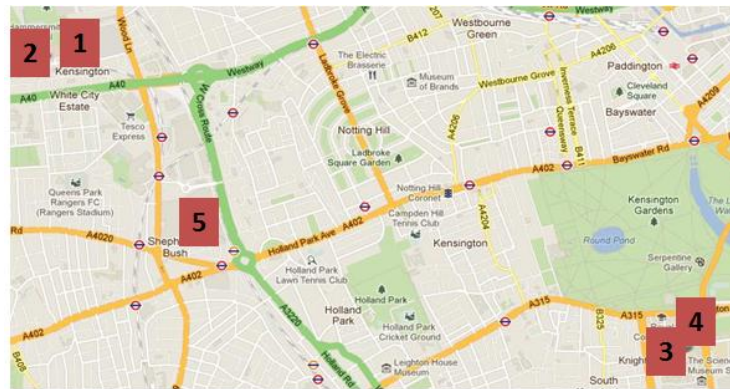


Figure 6.16 Average time spent at top 100 significant location candidates. Typically users spend most of their time in a small number of locations, which are selected by the algorithm.



- 1 - Home
- 2 - Closest bus stop to home
- 3 - Workplace
- 4 - Weekly meeting location
- 5 - Shopping mall

Figure 6.17 Semantic labels attached to significant labels found by the algorithm can be placed by users.

Semantic labels can be assigned to the significant locations using feedback from the user. Figure 6.17 shows some of the significant locations found in our analysis, along with the semantic labels retrospectively added after the locations were computed.



Figure 6.18 Routine tree incorporating location into the visualisation. Location can be used to contextualise the activity patterns.

When significant locations are added to the database transactions of the routine tree, it is possible to associate frequent patterns of activity with the location at which the activity is performed. Transactions in the database comprise of twenty seconds of distinct activities and locations. In addition to the visual output of the routine-trees, the semantic labels manually assigned by the user are shown in Figure 6.18. The patterns are for the day shown in Figure 6.14, however given the semantic labels it is possible to contextualise the user's activity. For instance, the meeting activity can be inferred from the 'Office 2' location, which is distinct from 'Office 1' where the participant is usually situated. Similarly, exercise activity can be inferred based on both higher level activity patterns as well as the user's presence at the gym.

A second method proposed for incorporating location into the mining task is to consider location as a dimension, i.e. group activities by time and location. It may be important to drill down and investigate participant's routine activities in particular locations (for example, unusual kitchen activity may indicate a change in eating patterns). Figure 6.19 shows an example output for this approach, where the location is the participant's home. In this instance, the participant is active in the morning, not-present at home during the daytime, and returns in the evening.

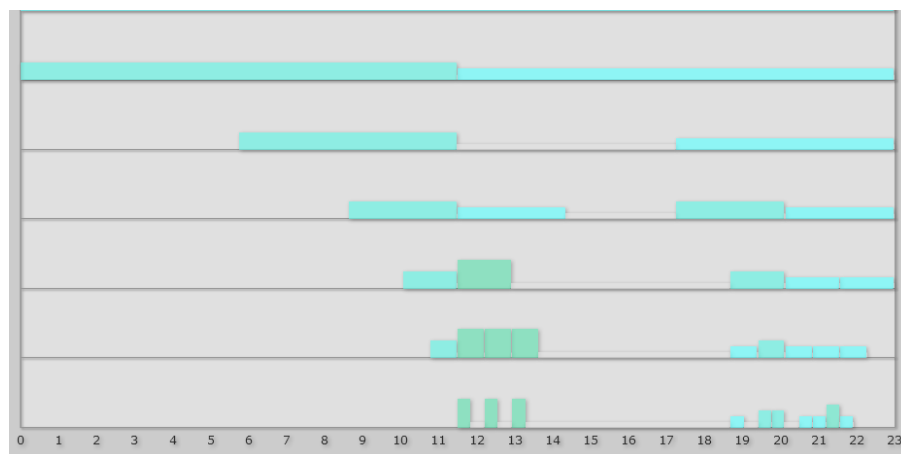


Figure 6.19 Routine tree for the 'Home' location, showing activity in the morning before the participant leaves for work, and in the evening when he returns.

Conclusions

This chapter presented a system for collecting and analysing long-term activity and routine data from the widely used platform of Android™ based smartphones.

While there has been previous work on behaviour profiling using mobile phones, most of this has utilised GPS based location and tended to focus on narrow applications such as those related to exercise. ActiveMiles opens up the possibility of continuous activity monitoring through the development of algorithms that can be deployed on the phone itself in order to reduce the data volume, while extracting features at high sampling rates. The main contribution of this chapter lies in the exploration of strategies for long-term trend data analysis by applying clustering methods with automatic cluster size detection, and adaptation of methods proposed in previous chapters.

Study I suggests the feasibility of developing very specific classification of activity from the features extracted from the phone. It should be noted that it is based on only one participant's data, and classifiers such as the GRNN for detecting running speed must be trained on a large number of participants as well as different phone placements in order to be robust. The analysis opens the way towards clinical studies, such as those for obesity monitoring and a more precise association with clinical measures such as calorie consumption. An on-going study will validate the activity output with clinically standardised caloric expenditure measurements.

In the absence of enough training data to learn participant-specific, precise models of activity, we propose the use of clustering. The GMM clustering algorithm, in particular was found to generate both stable clusters, and was found to correlate with manually placed labels of activity. We use this to generate a label for the participant's activity.

Given this categorisation of 'activity', a methodology was proposed whereby the routine of the participant on any given day could be summarised in terms of activity and location either using the routine tree visualisation or a diary. Key locations were selected from the GPS/cell-tower based locations that corresponded to significant places for the user. These are used to provide more detailed reports on the participant's routine.

One significant challenge for activity profiling on smartphones lies in the need to conserve system resources. Software such as ActiveMiles will consume significant battery power due to

sensor usage. In particular GPS sensors have a high operation cost, and therefore must be used sparingly. In future work we will explore the use of surrogates for expensive sensing, for instance using coarse network location and user activity to estimate GPS location. A further direction for research is to adapt the sensor sampling rates to the activity being performed. Another area for resource optimisation is in data transmission, which impacts not only the phone's battery significantly, but also can be potentially expensive depending on the mobile internet charges. Several measures have been taken to address this. Users can choose to only perform data upload when the phone is being charged. There is a further option to transmit only when connected to a wireless internet network. To minimise network use, the data is stored and transmitted in a compressed, binary format. A direction for research is local processing to filter data that is redundant or can be reconstructed on the server from previous data or other sensors.

In our data-mining based presentation of routines so far, the analysis has been mainly visual and qualitative. While such abstractions may be useful for care-providers and users of the system, it is important to be able to characterise routines and change in routines in a more quantitative framework. In the next chapter, we extend our data-mining based routine algorithm by proposing mechanisms to compute the distance between two daily routines, thereby permitting clustering and classification of routine behaviour and changes in it.

Chapter 7

Long-term Profiling with Pervasive Sensing

7.1 Introduction

IN previous chapters a set of techniques was proposed for the extraction of behaviour related information from ambient and body-worn sensor data. This chapter extends research from quantitative analysis of activity to that of routine. The primary consumers of such healthcare related information are the participants themselves and their care providers. Data from pervasive sensing systems, even abstracted into the daily picture of routine, accumulates very quickly. It is difficult to track changes, and patterns of behaviour. It is important therefore to be able to present the information by categorizing it into groups, and detecting and presenting significant deviations from the normal types of data.

In this chapter, we will propose quantitative measures that can be used to compare an individual's behaviour over the long-term developing on the techniques described in Chapter 6. As in earlier chapters, the goal remains to discover categories and types of behaviour in unlabelled data. In particular, we focus on two use case scenarios. The long-term care of elderly participants in a home based healthcare scenario and the analysis of active miles data to detect changes in activity or lifestyle.

Much of our analysis is focused around the temporal unit of a day. The design choice is driven by the dependence of human routine on the circadian rhythms. These rhythms are influenced by

daylight, having been evolved to regulate human behaviour around the day-night schedule of the Earth. It is possible however, that certain periods of the day may offer more structure, and deviation in these periods may be more meaningful. For instance, it is important for care providers to note variability in sleeping patterns [242, 243]. This includes disturbed sleep as well as interruptions in it. It is important therefore to be able to compare specific periods of the day to detect. The multi-resolution temporal model of the routine-tree is well suited for this analysis, as we can bias our analysis to focus on specific portions of the routine-tree.

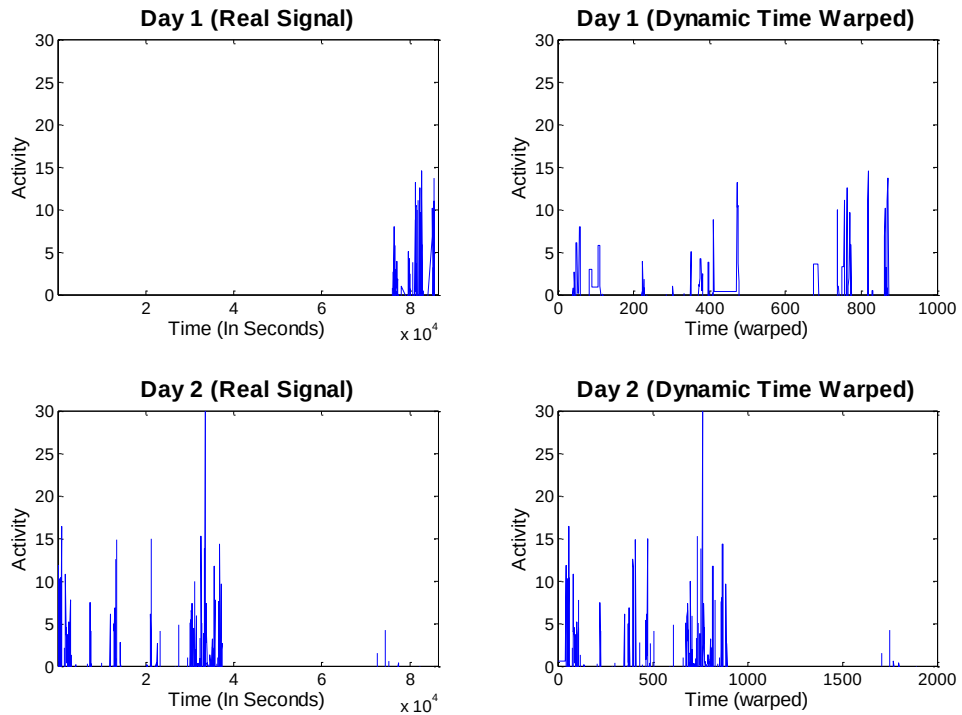


Figure 7.1A A comparison of two days of sensor data using with the real signal on the left and alignment using Dynamic Time Warping (DTW) on the right. The missing data in day 1 results in an incorrect alignment.

It is important to note one significant complication in the comparison of sensor data over different days stemming from how participants use the sensors. It is difficult to predict the times of the day in which the participants have sensors on their person. Much of the data may be missing, with arbitrary starts and stops to the data stream. Typically analysis of behaviour does not take time-of-day into account. For instance, using Zhou and Torre [244] propose an algorithm based on Dynamic Time Warping (DTW) for comparing behaviour, and Atallah et al.

[50] compare sequences of activity using Hidden Markov Models (HMM). Since neither HMM nor DTW have an explicit model of time, the alignment can result in comparisons that are not meaningful, for example by aligning late-night activity with daytime activity. This is the case in Figure 7.1, where the participant used the Active Miles program only in the evening period of day 1, but throughout the day on day 32. The DTW alignment results in the evening activity stretched over the entire day. The distance produced from this alignment will not be correct. Even though the aligned signals look relatively similar visually, the participant was active on day 1 and inactive on day 32 in the evening time period. Our analysis compares like-with-like by comparing the same periods of the day.

An attractive property of both DTW and HMM however is in the ability to align sequences with inter and intra participant variability. Sequences beginning at different times of the day, with small deviations can be mapped. Likewise, if a participant has the exactly the same routine, but begins his day earlier or later than usual, both DTW and HMM can cater for these conditions. If the model of time is too rigid, minor variations in routine may be exaggerated. For instance, Virone et al. [155, 245] associate circadian activity statistics of elderly participants in a smart home with every hour of the day, while Barger et al. [156], while not associating explicit statistics with every hour, also begin their analysis with clustering of hourly firings of ambient sensor data. While our data structure associates activity patterns with time, the multi-resolution nature of the data structure allows for analysis at different time scales. Therefore days with similar activity with small variations in time will be grouped together at the high-levels of the tree.

Our analysis begins with a method for specify the distance between two routine trees. As the routine-tree is a multi-resolution dataset, this requires finding the distance between each overlapping node in the trees, which is used to construct a similarity matrix representing distance at multiple timescales. This similarity matrix can then be used for visual presentation of the complete dataset, or to cluster it into groups. Anomaly detection is then performed to find days where the user's behaviour is significantly different.

Results will be presented for two datasets: a dataset collected from one chronically ill participant over six months in a home monitoring study, and an ActiveMiles dataset from a healthy participant consisting of 83 days of data.

7.2 Methodology

The starting point of analysis in this chapter is a database containing historical activity and routine data. The routine data is comprised of a routine tree mined from historical activity; this can include location. Location may be using ambient sensors such as Passive Infra-Red (PIR) sensors or global positioning using GPS or cell tower triangulation. Technically, the proposed framework consists of the following main steps:

- 1) Activity Labelling: Activities are extracted from wearable sensors. For the activity data this corresponds to clustering using GMM as described in Chapter 6. The clusters may include location as a feature, or location can be mined separately.
- 2) Routine Mining: Routine trees are mined from databases where each transaction is comprised of a window of ten activity samples and any associated locations.
- 3) Computing a Distance Matrix from Routine Trees: Distance matrices are computed for every node in the routine tree dataset. Each entry in these matrices corresponds to the distance between the corresponding trees at that time period. A combined distance matrix is finally generated by combining these matrices.
- 4) Analysis of Distance Matrices: The distance matrices thus derived are used for clustering and anomaly detection. Dimensionality reduction mechanisms such as Isomap can be used to cluster and visualise the data. Mechanisms are included in the analysis for the biasing towards temporal periods specified by the care providers as most relevant.

The main contribution of this chapter is in proposing a method for generating the distance matrix of routine trees. This is discussed further in the next section.

7.3 Generating a Distance Matrix from Routine Trees

The challenge of processing a routine dataset is to preserve the time-sensitive nature of human behaviour while also taking into account variability in how people perform their activities, and use the data collection hardware. To avoid misalignment as in Figure 7.1, we compute separate

distance matrices for each time period in the routine trees. These matrices are referred to as Node Distance Matrices (NDM). The matrices cannot be simply added together however, as this would not take into account the duration of the node, and the extent to which it reflects interesting variation. Instead we take the approach of clustering each distance matrix separately, and then combining the cluster labels in a distance matrix that reflects how often different days are clustered together. The workflow is illustrated in Figure 7.2.

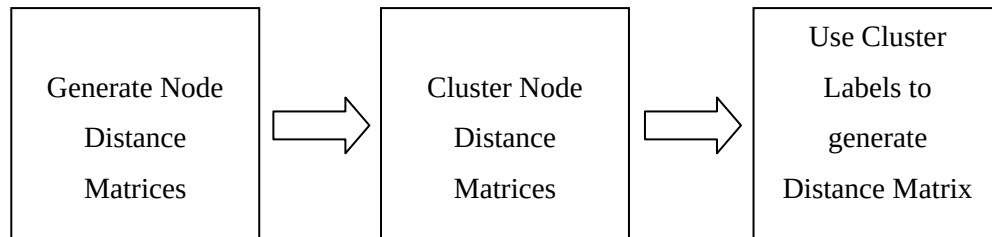


Figure 7.2Methodology for generating a composite distance matrix from a set of routine trees.

7.3.1 Computing Node Distance Matrices

Routine Trees are associations of a table of activity patterns with time periods, referred to as nodes, which are of successively smaller duration at each level of the tree. We compute distance matrices for each node possible in a routine tree. The number of matrices is limited by the recursion level specified during routine tree construction. The distance between two routine trees at a given node is called the Node Distance (ND). Figure 7.3 shows three routine trees, and three node distance matrices that can be generated from the overlapping nodes. It can be seen that there may be fewer matrices than possible, depending on the amount of overlap in the trees, and many entries in the matrices may be null.

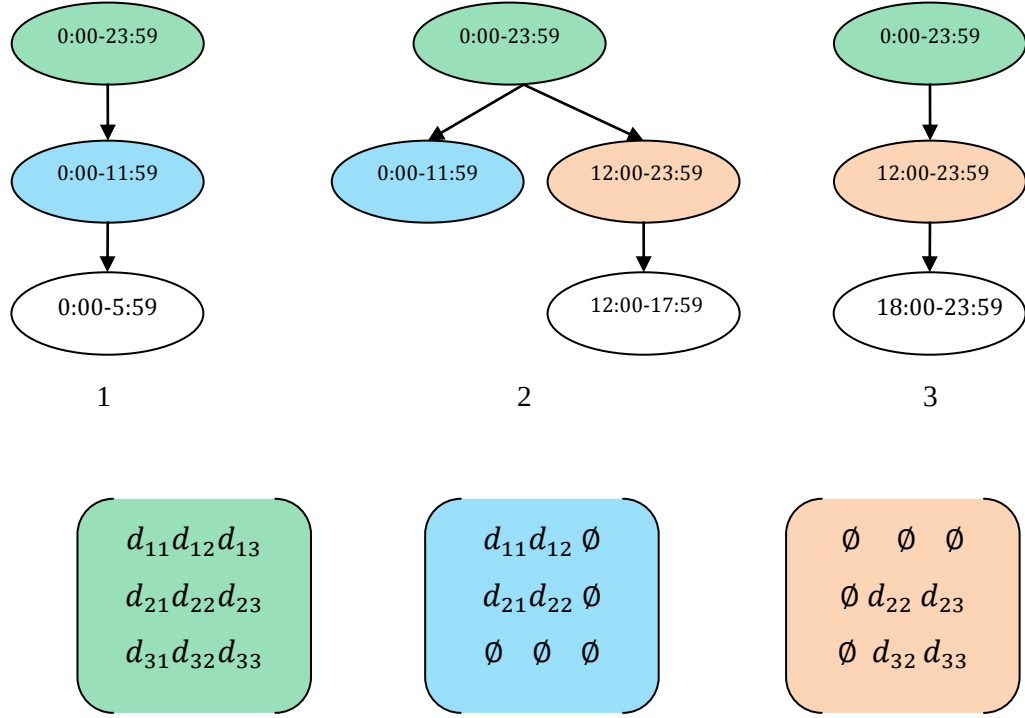


Figure 7.3 Three Node Distance Matrices that can be generated from three routine trees, with nodes colour coded according to their associated distance matrix, where $\emptyset$ denotes a null entry.

To compute ND, we use a similarity measure based on one proposed by Li et al. [246] for testing the similarity of datasets using maximal frequent patterns. Let there be two nodes A and B , and $\text{sup}(P)$ be the associated frequency with the pattern P as a proportion of the dataset, so that

$$\begin{aligned} A &= \{X_i, \text{sup}(X_i)\} \\ B &= \{Y_j, \text{sup}(Y_j)\} \end{aligned} \quad (7.1)$$

where X_i and Y_j are maximally frequent patterns in A and B respectively.

ND can be found by the following equation:

$$ND(A, B) = 1 - \frac{2I_3}{I_1 + I_2} \quad (7.2)$$

Where

$$I_1 = \sum_{X_i, X_j \in A} d(X_i, X_j), \quad I_2 = \sum_{Y_i, Y_j \in B} d(Y_i, Y_j), \quad I_3 = \sum_{X_i \in A, Y_j \in B} d(X_i, Y_j) \quad (7.3)$$

$$d(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} * \log\left(1 + \frac{|X \cap Y|}{|X \cup Y|}\right) * \min(\sup(X), \sup(Y)) \quad (7.4)$$

It can be seen that $0 \leq ND \leq 1$ with 0 denoting no difference between the activity patterns and 1 denoting maximum difference. The measure captures both differences in the frequent patterns as well as the frequency of the patterns in the dataset.

7.3.2 Generating Composite Distance Matrix

Our next step is to generate a Composite Distance Matrix (CDM) that incorporates information from all the NDM. One method is to simply add together the different matrices. This is problematic for several reasons. Firstly, the matrices are over different durations, and therefore the addition would need to be weighted by a parameter reflecting scale. However, the addition should also incorporate the significance of differences within the NDM; interesting detail in small time periods may be lost because of longer time periods. Furthermore, as noted in the previous section, many of the entries in the NDM may be null, requiring the substitution of default values in place of the null entries.

Due to these difficulties we take a clustering approach, by generating cluster labels for each NDM. These labels are then composed together by using a cluster ensemble algorithm. We begin by clustering the subset of the NDM for which there are no null values using GMM clustering. The number of clusters can be user specified, or can be found automatically using the gap statistic. Days missing from the clustering are assigned a null cluster label.

Given m cluster groupings, a cluster ensemble algorithm finds an integrated clustering that shares the most information with the original clusterings. One approach, referred to as Cluster Based Similarity Partitioning Algorithm (CSPA) [247] finds a similarity matrix representing information from all the clustering. This is attractive, as the matrix can be used with the algorithms described in Chapter 4 for visualizing data in manifold. Cluster ensemble algorithms were found by Strehl and Ghosh [247] to produce high quality clustering in noisy data. Given a set of cluster labels, if two objects are placed in the same cluster, they are considered to be fully similar, and if not they are considered to be fully dissimilar. This results in a set of $n \times n$ binary similarity matrices, with 1 representing similarity and 0 representing dissimilarity. The

modification we make is to set as 0 any entry where one or both days have a null cluster label. The entry-wise average of m matrices yields a composite similarity matrix, which subtracted from 1 gives the CDM.

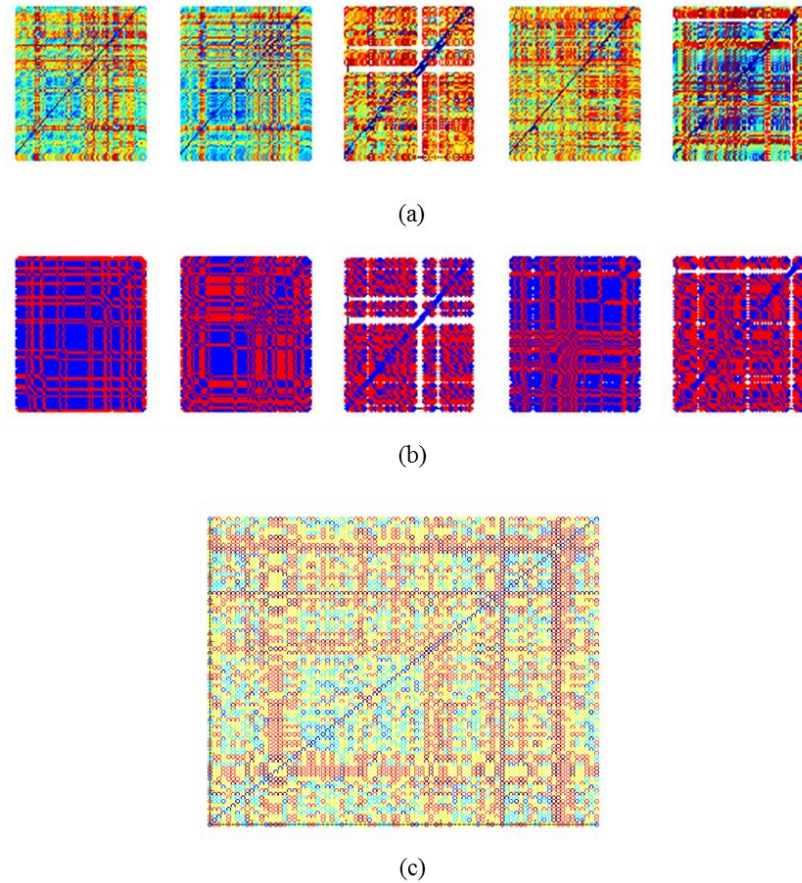


Figure 7.4 Five Node Distance Matrices (a) , their corresponding similarity matrices (b) and the Composite Distance Matrix (c) containing information from all the NDM. Blue corresponds to low distance, yellow to medium and red to high.

Five NDM are shown in Figure 7.4(a) with blue representing small distances and red indicating high distances. Empty rows and columns indicate missing values. The similarity matrices generated from the NDM are shown in Figure 7.4(b). The composite distance matrix shown in Figure 7.4(c) contains information from all five NDM.

One consideration made is to take into account the size of the node corresponding to the NDM. The size of nodes is smaller lower in the tree, and therefore lower levels of the tree will contribute more to the CDM. Instead of introducing scaling, we take a cluster ensemble approach here. The number of clusterings generated from a single NDM corresponds to the height of the node in the tree. If the height of the tree is h and the height of the node n is h_n , 2^{h-h_n} labels are generated from it. In this way, the same number of clusterings is generated from each level of the tree.

7.4 Analysis of Distance Matrices

The typical structure of a participant's daily routine, as well as variations over time can be analysed using dimensionality reduction and clustering techniques discussed in previous chapters. In brief, we apply the following analysis on the NDM and CDM distance matrices:

- **Manifold Embedding:** Manifold embedding techniques, such as Isomap, are useful for presenting data in a small number of dimensions that capture the intrinsic structure of the dataset. The intrinsic dimensionality can be estimated using the residual variance of the data. Given a distance matrix, we represent it in its estimated intrinsic dimensionality using Isomap. Chapter 4 describes dimensionality reduction in more detail.
- **Clustering:** The distance matrices are clustered using Gaussian Mixture Model (GMM) clustering, with the number of clusters chosen automatically using the gap statistic. Chapter 6 describes GMM and the gap statistic in further detail.
- **Qualitative analysis:** By finding points close to and distant from centroids in the dataset we can visualise typical and anomalous days.

7.5 Results

We will present results for two datasets. The first dataset has been collected using the ActiveMiles application described in Chapter 6. The second is from a deployment of a home-based activity monitoring project, SAPHE. The strength of the data mining approach lies in the

applicability of the algorithm to both scenarios where data is streamed continuously, and without annotation.

7.5.1 ActiveMiles Dataset

We introduced the ActiveMiles dataset in Section 6.8. This is comprised of accelerometer, orientation and location data from the ActiveMiles phone application. The participant (30 years old, male) used the application over a period of 83 days between October 2011 and February 2012. Chapter 6 presents the clustering of activities in this dataset, and demonstrated the generation of Routine Trees using the cluster labels. In this section we present analysis of the complete dataset.

To understand the ‘typical’ day out of a collection of routine data structures, we find the centroid using NDM, and display the node from the day closest to the centroid. An example of this can be seen in Figure 7.5, a composite tree formed by selecting nodes from trees in each DNM closest to the centroid. It can be observed that the typical day of the participant has low-level activity till 7:00, with an increase between 7:30 and 19:00 corresponding to the active period of the day when the participant is at work, followed by a decline in the activity. There is higher level activity during the day around lunch time, beginning around noon, and ending close to 14:00. The highest activity in the day is observed between 13:00 and 14:00.

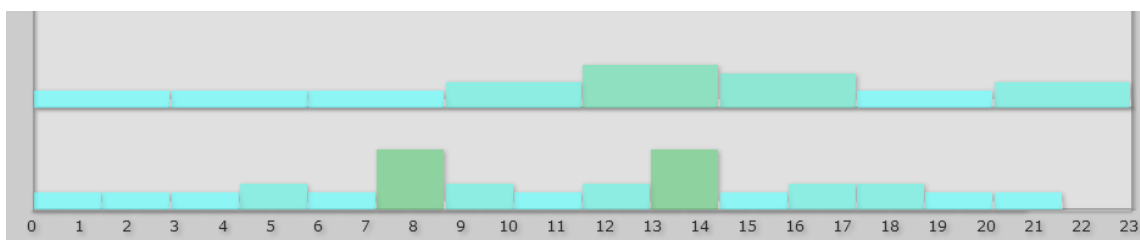


Figure 7.5 Activity patterns corresponding to the centroid node of the NDM. This corresponds to a ‘typical’ routine of the user.

Figure 7.6 shows the variations in the routine-trees over time when the dataset is visualised in three dimensions using Isomap. While the association of the data with the month is not consistent, some patterns can be seen in the data. There is a significant change in routine

between December and January. February points are closer to December than January. This change corresponded to an increase in the participant's workload resulting in unusually low activity and less physical activity during the waking hours.

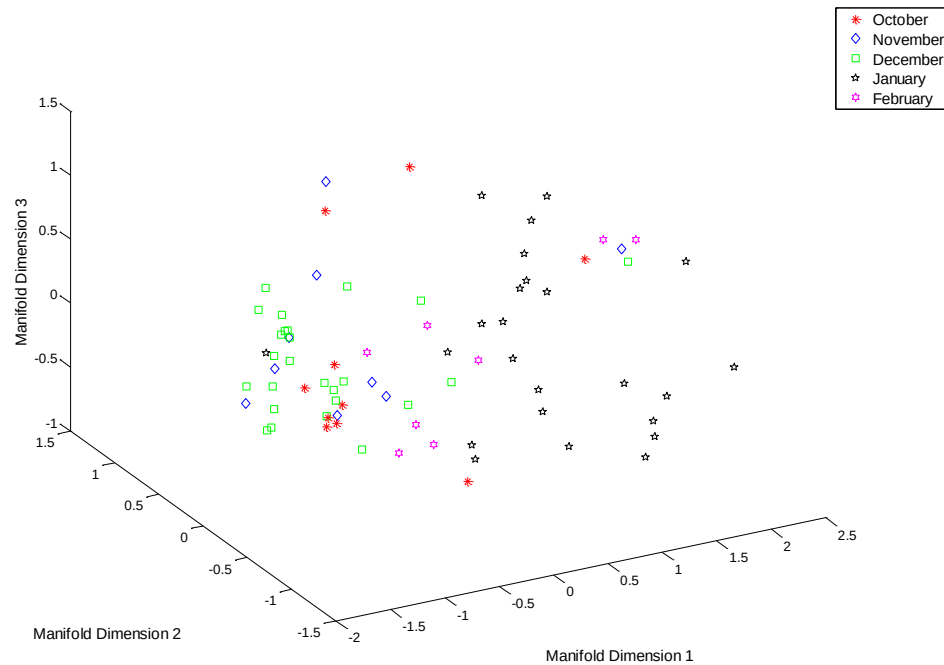


Figure 7.6 Isomap Embedding of Composite Distance Matrix, with each day coloured according to the month.

The pronounced difference in behaviour can be classified. We divided the data into 2011 and 2012 months. Using the Simba algorithm was used to find manifold dimensions most discriminative between the two classes. Using a Neural Network classifier a classification accuracy of 86.7% is achieved, with manifold dimensions 1, 2, 5 and 7 selected by Simba.

The residual variance of the Isomap embedding is plotted in Figure 7.7. It can be observed that even for a relatively small dataset, the variation is high as the elbow in the curve only appears after the 7th dimension.

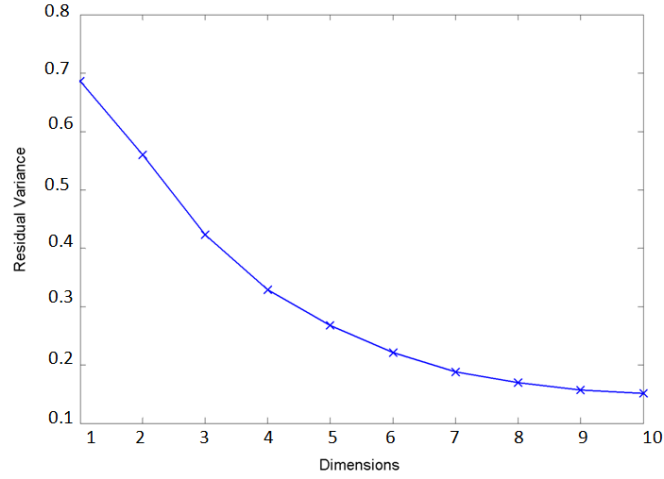


Figure 7.7 Residual Variance of the Isomap reduction of the CDM. The elbow of the curve occurs at the seventh dimension.

Outlier points can be visually observed in the embedding. Outliers are computed based on the Mahalanobis distance from the dataset. Formally for a group of size p with mean $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_p)^T$ and covariance matrix Σ the Mahalanobis distance D_M is defined for a data point $x = (x_1, x_2, x_3, \dots, x_p)^T$ as:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (7.5)$$

Figure 7.8 shows four outliers based on thresholding the Mahalanobis distance at 95%. Trees shown in 8(a) and 8(b) were found using the manifold embedding of the CDM, while 8(c) and (d) is for NDM associated with the a.m. and p.m. hours of the day respectively. 8(a) is unusual for the medium-high activity patterns during the night time and close to 6 a.m., while 8(b) shows a day when the participant performs low intensity, sporadic activities during the day. 8(c) corresponds to a day with more than usual activity patterns during the early morning, close to 2 a.m. 8(d) has low activity for most of the day, with the participant remaining inactive between 2p.m. and 11p.m. This may correspond to a long working day.

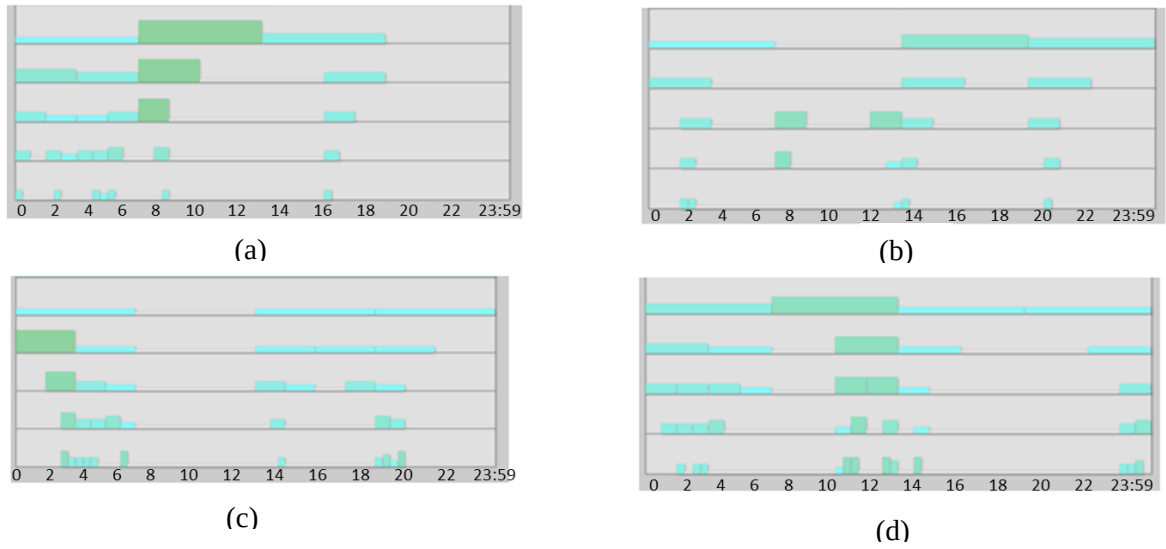


Figure 7.8 Routine Tree Outliers computed using Mahalanobis distance on embedding of the CDM (a) and (b), using NDM for times between 00:00 and 11:59 (c) and using NDM for times between 12:00 and 23:59

To find the number of clusters best suited for the dataset, we performed a gap-statistic analysis on the dataset. Figure 7.9 shows the results when the analysis is repeated 50 times. The most commonly selected number of clusters is 3. Therefore this value is used in subsequent analysis.

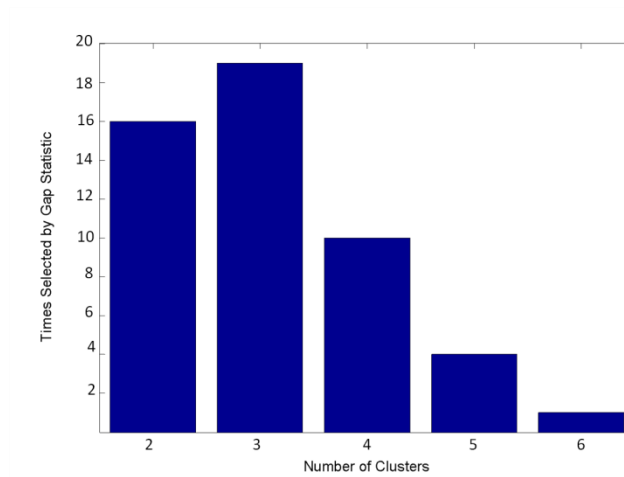
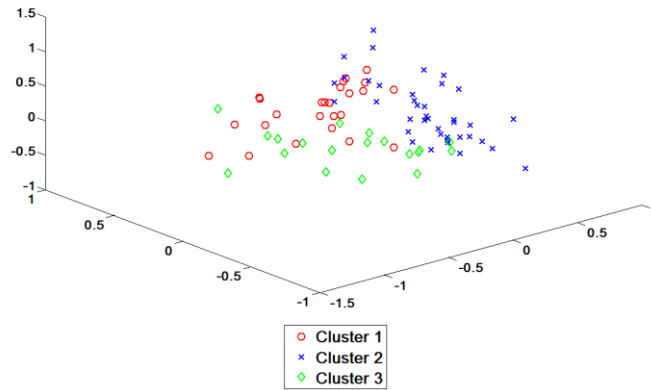
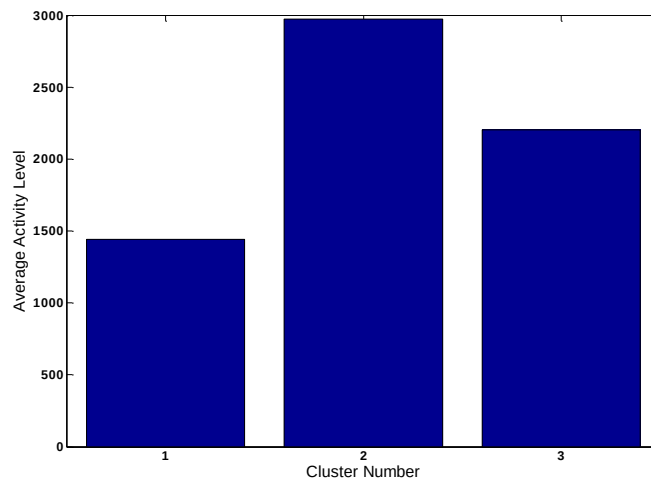


Figure 7.9 Gap Statistic analysis on the CDM of the Active Miles dataset. Based on the analysis the number of clusters is set at 3.

Figure 7.10(a) shows the clustering results on Isomap embedded CDM. While there is some overlap between the clusters in three dimensions, further dimensions will further separate the clusters. Figure 7.10(b) shows the average activity level aggregated for the routine trees in each cluster. Based on 7.10(b) we can estimate the nature of the routines in each cluster.



(a)

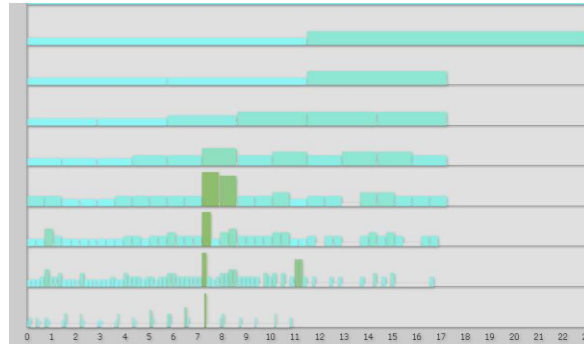


(b)

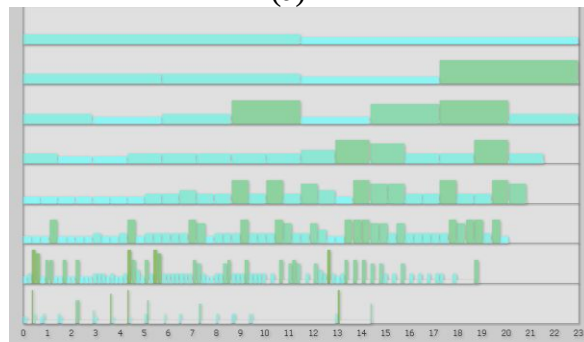
Figure 7.10 GMM clustering of CDM shown in manifold space (a), and average activity level of each cluster.

Figure 7.11 shows representative days from each of the three clusters. The day corresponding to cluster 1 in 7.11(a) clearly has lower level activity during the waking hours. In contrast, the day

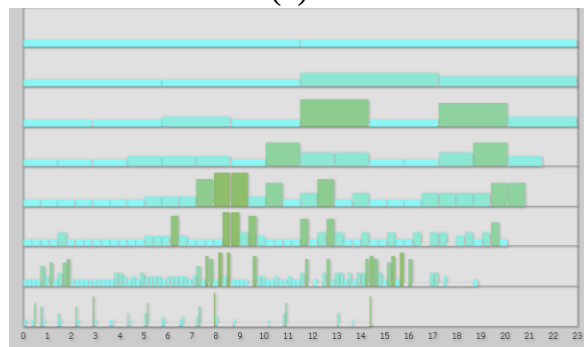
corresponding to cluster 2 in 7.11(b) has activities spread over the entire day. This cluster is the largest in the size, and may be considered as the most ‘typical day. Finally, cluster 3 in 10(c) has a mixed nature. While activity is spread over the day, it is lower intensity with periods of activity corresponding to the commute. It can be hypothesised that these are ‘normal’ days with relatively lower physical activity over the course of the day.



(a)



(b)



(c)

Figure 7.11 Representative Routine Trees corresponding to the three clusters respectively.

The clusters can be further investigated by a day of the week analysis. Routines can often have a strong dependency on events taking place in the week. For instance, the participant attends weekly meetings every Wednesday. Figure 7.12 shows the membership of days in clusters for each weekday.

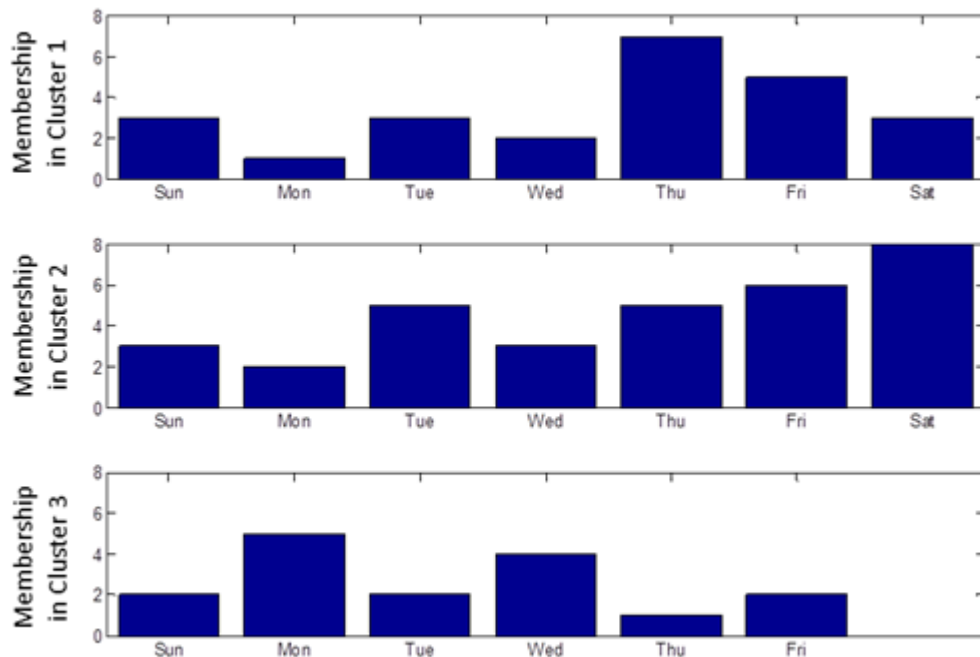


Figure 7.12 Membership of days of the week in the three clusters. Cluster 3 has low weekend representation, in contrast to Clusters 1 and 2.

While not consistent, it can be observed that cluster 3 has relatively low membership after the Wednesday meetings. In contrast clusters 1 and 2 have high memberships from Thursday-Saturday. The highest memberships in cluster 3 are Monday-Wednesday. This suggests that before the Wednesday meeting, non-work related activities that would result in higher level activity in the evening are curtailed. Immediately after the meeting however, the participant either returns to a normal schedule (cluster 2) or has a more relaxed schedule (cluster 1). Saturday in particular is either associated with cluster 2 or cluster 1.

7.5.2 SAPHE Dataset

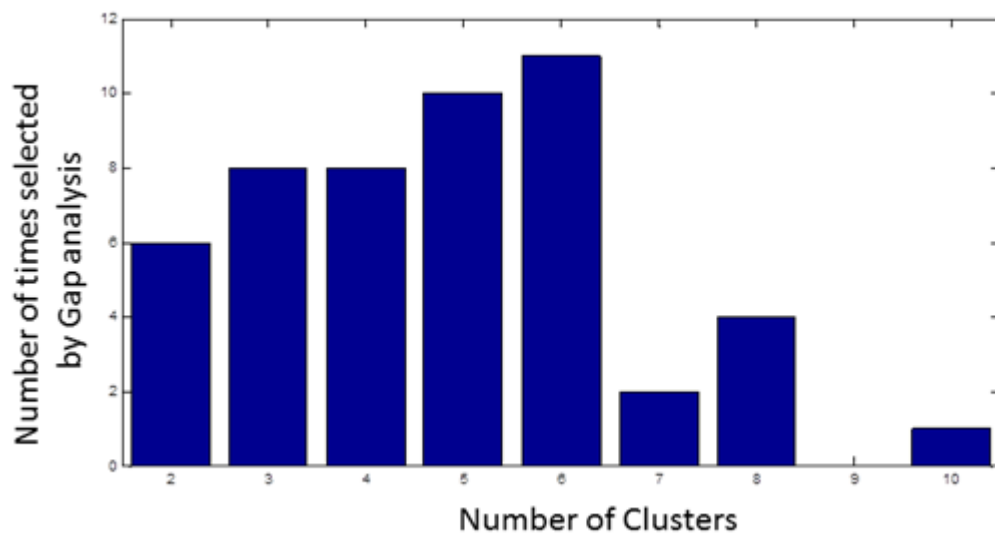


Figure 7.13 Gap analysis of SAPHE dataset. Plot shows number of times cluster size is selected by the Gap criteria, with six cluster sizes selected most often.

The second dataset is from the SAPHE project, introduced in Chapter 5. Chronic patients requiring care from community matrons were supplied with e-AR sensor, and ambient sensors at their homes. The system allowed care-providers to visualise the health status of their patients remotely. As this was a project focused on the development and reception of such systems with the National Healthcare Service (NHS), overriding concerns included privacy and ease of use. To this end, participants were anonymous, and there is little metadata available. Compliance with the system was also left to users; therefore over six months between February-August 2009, the participant (50 years old, female) wore the e-AR sensor for 44 days. Table 7.1 shows the number of days of data available for each month.

	Days
February	4
March	4
April	0
May	4
June	3
July	21
August	7

Table 7.1 Months for which routine data is available in the SAPHE dataset and the number of days the user wore the e-AR sensor. The user wore the sensor most regularly in July.

Figure 7.13 shows results for the gap analysis of the SAPHE dataset. It is indicated that the routines can be distinguished into six clusters. However, a limitation of this dataset is the absence of ground truth. A cluster analysis can only offer limited information about the structure present in the data. It was difficult to visually see the variation in the dataset corresponding to the clusters, suggesting that the differences may be subtle. In this case the conservative approach is interaction with the participants, to verify the variation in their behaviour. As this is not possible for the SAPHE dataset, we shift our focus to parameters that can be substantiated, one of which is daily and seasonal variations.

It is evident from Table 7.1 that the participant wore the sensor most regularly during the month of July. This can be due to an intervention from the community care provider, prompting increased use. A change in routine is also visible in July, as shown in Figure 7.14 where the manifold embedding of the CDM is shown.

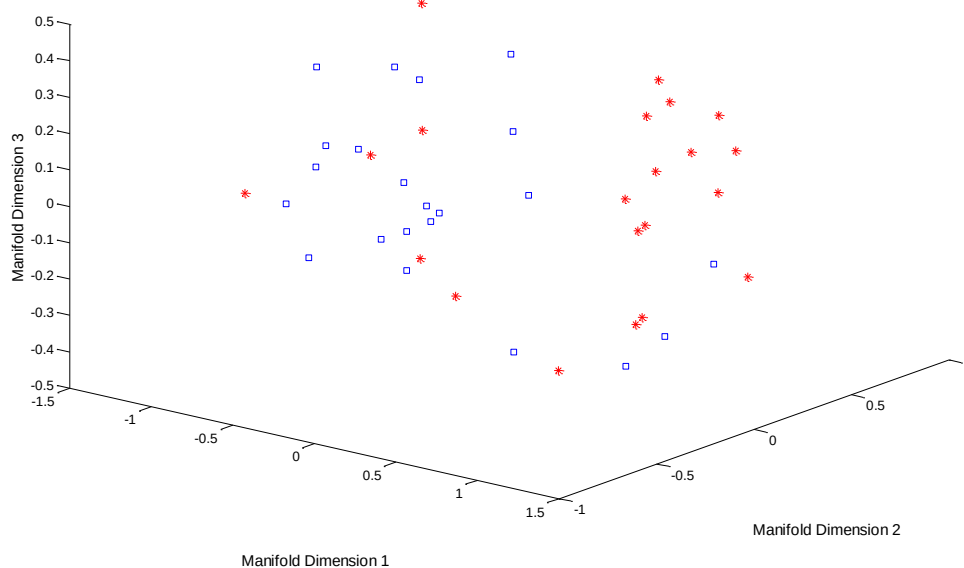


Figure 7.14 The participant's routine changed during the month of July (red) compared to the routine in other months (blue)

Day of the week variations in an individual's routine may be due to social connectedness [248], a key element in the well-being of patients requiring chronic care [249]. One such variation in the manifold embedding may be a change in routine on Saturday. Figure 7.15 shows the participant's routines, with Saturday shown in red. There are relatively few points to perform this type of analysis. Six Saturday routines out of nine lie close to each other in the manifold. In a real deployment a weak clustering pattern such as this could be the seed for further investigation, particularly as a target for community based care. With more data and further research we may classify healthy 'social' routines versus routines suggesting social alienation.

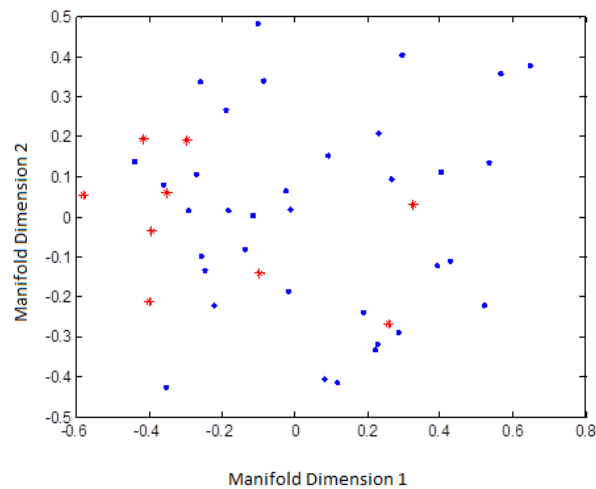


Figure 7.15A manifold embedding of the CDM with Saturday routines shown in red and remaining weekday routines in blue. Although there are too few points to make a strong claim, a weak grouping of red points could indicate change in behaviour associated with the weekend.

It can be seen that specific regions of the routine in manifold space can be associated with real world parameters such as day of the week and month. The SAPHE project purposively withheld metadata in the interest of participant privacy. Our work is promising in indicating the feasibility of categorizing routine types, and under more controlled conditions it may be possible to learn to identify medically relevant changes in routine.

7.6 Conclusions

Chapters 2-6 developed a framework for the analysis of activity, and the abstraction of routine from activity. In this chapter this was extended into the quantitative analysis of routine. The key contribution was the development of a set of distance matrices based on frequent patterns associated with tree nodes. Subsequently these distance matrices were aggregated using cluster ensemble methods.

The strength of our framework lies in the meaningful profiling of activity at the small temporal resolution, and then extending that to longer time periods, with a view to mining of historical data with minimal input from domain experts or the requirement of any manual logging.

One strength of our algorithms lies on the compression achieved from representing routine using activity patterns. The volume of sensor data that is used to generate figures such as Figure 7.15 is very large, constituting six months of accelerometer data. The analysis owes its efficiency to the abstraction of routine tree and the compression it achieves, as was demonstrated in Chapter 5.

Routine-trees are similar to wavelets in their multi-resolution scaling of activity patterns. As in wavelets, higher-level nodes represent overall trends, while lower-level nodes capture local variations. By analysing with this multi-resolution data structure our distance measures capture similarity in both terms: global and local. This can control for normal variations in routine activities, for instance it is not necessary for a participant to eat lunch at a precise time each day in order for it to be reflected as an activity usually performed in the early afternoon in the participants routine-tree profiles.

The generality of the algorithm was demonstrated by application on two datasets collected in very different contexts: a smartphone application and a home-healthcare environment. Very little metadata was assumed, or even available. Despite this, the exploratory data analysis approach allows us to discover interesting variations in the routine behaviours of the participants, particularly variations in routines over weekdays and over longer periods of time.

Characterizing regularity of routine and changes in it has applications in the treatment and prevention of neurological illnesses such as insomnia [250] and depression [251], particularly in

elderly groups. Further research is needed to detect routines associated with socialising [252], an important lifestyle recommendation to avoid dementia. However, in the SAPHE dataset it was possible to see a noticeable clustering of weekend activity, which has been suggested as strongly influenced by social activity [248]. In the ActiveMiles dataset it was possible to see a decrease in exercise regularity, which is of interest in obesity management interventions [253].

Our methodology allows for analysis beginning at the raw activity stream and progressively derives human activity and behaviour related meaning. Through feature selection and dimensionality reduction, informative and succinct representations of the signal are generated. It is then possible to classify activities and transitional activities from this feature space, using the techniques described in Chapters 3 and 4. The clinical applications of such activity analysis were demonstrated through application to impairment detection and the tracking of recovery from knee-replacement surgery. Once sensor signals are transformed into activity streams, the routine-tree algorithms allowed for a summarisation of routine-behaviour related activity patterns, so that a participant's routine could be visually explored without necessitating the processing of sizeable activity databases. In addition to qualitative analysis, in this chapter, we described mechanisms for the quantitative analysis and comparison of routines, enabling an efficient analysis of entire sensor databases in terms of routine behaviour.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Driven by economic demands and technological opportunities, health and well-being applications key areas of research for pervasive sensing [8]. This includes activity and behaviour parameters that could inform about the user's health through inexpensive and unobtrusive sensing. Given the vast array of healthcare applications and the evolving technology, there remain significant areas that are essential to address to make pervasive sensing systems effective and widely used. These were put forward in Chapters 1 and 2.

We have proposed a set of techniques that can be used for mining wearable sensor data for health and well-being related applications. The techniques build on well-established machine learning and data mining algorithms applied in novel ways on sensor data. Several new algorithms and data structures building upon existing techniques have been proposed. The transition detection algorithm applied spectral bisection in a recursive way, detecting transitions using a technique based on Fuzzy C-Means clustering. Routine-tree construction algorithm similarly relied on a well-known algorithm for pattern mining, applying progressively to construct a novel data structure for summarising routine. The algorithm for quantifying change in routines relied on techniques developed for comparing databases using frequent patterns to generate a set of distance matrices, which were combined using a well-known cluster ensemble algorithm.

Validation was performed through application on laboratory collected data with simulated impairment, clinical studies as well as data from healthy participants. Key advances in this dissertation are data-driven methodologies for transitional activity analysis and routine behaviour visualisation and analysis.

8.2 Summary of Methodologies Developed

This thesis was motivated by the practical need of exploiting the intrinsic structure of the data because of the limited availability of annotated data in practice, and the impracticality of retrospective labelling of streamed sensor data. Given this paradigm, our focus was on incorporating unsupervised and semi-supervised approaches wherever possible with the aim of discovering information instead of requiring expert direction. Therefore we rely on clustering, manifold embedding and data mining to process the data before any supervised learning based algorithms are applied to obtain semantic labels. The imperative to process real-world data obtained in scenarios without explicit labelling, as in Chapters 5-7, can conflict with the need to provide precise, medically relevant output, as in Chapters 3-4. We have attempted to develop a methodology that progresses from fine-grained analysis of transient data with expert labelling, to profiling of the activity and behaviour parameters over extended periods without expert annotation. Even with expert annotation however, we aim to reduce dependence on manual labelling by providing a method for automatic detection of transitions from sensor streams.

In Chapter 3, we demonstrated the potential of pervasive sensing for healthcare applications through detection of simulated limb and torso impairment through Activities of Daily Living (ADL) performed in simulated home environments. The experiment allowed the highlighting of the types of features that are suitable for impairment detection. As a result a large set of statistical, frequency and wavelet features could be pared down to a small feature-set optimised for each activity type. Impairment could be generally detected with high accuracy.

ADL detection and the detection of impairment in specific types of activities such as walking is a widely researched area. Relatively less research focus has been placed at how people transition between such activities. This is in particular relevant as a specific transitional activity, sit-to-stand (STS), is widely used in clinical and biomechanical assessments. We advanced the state of the art in Chapter 4 by studying the detection and analysis of a wider range of transitional activities than previously studied, applying our methodology to a real-world application of post-operative recovery. One key finding is that for certain patient types, STS may not be the ideal transition to study despite its widespread use, because patients were usually unable or unwilling to perform the activity. In contrast, stepping transitions were performed by participants generally with much more ease, and were still found to be promising for demonstrating recovery from surgery. Our methodology relied on manifold based techniques

applied in a novel way for transitional activity analysis. An algorithm was proposed relying on recursive spectral bisection to automatically segment transitions. Manifold embedding was performed to generate a concise representation of transition data in its intrinsic dimensionality, allowing subsequent analysis using supervised learning methods. While the manifold space analysis is promising, one disadvantage lies in the explaining results from manifold space to clinical end-users. Manifold embedding is a so-called ‘black-box’ technique. That is to say it is agnostic of the meaning of the features used. The space engendered as a result of the analysis only represents the structure of the data, instead of representing clinically meaningful dimensions. While the intrinsic dimensionality of the data may well correspond to real clinical measures of relevance, visual exploration and depiction of the data may not be meaningful to doctors and nurses. This highlights the necessity of incorporating domain knowledge in the analysis. After unsupervised/semi-supervised learning techniques bring data into a form suitable for visual and quantitative analysis, it is necessary to correlate the abstract dimensions of clustering to labels with real-world relevance. In the absence of this, it can be erroneous to extrapolate specific measures of recovery from the abstract dimensions of manifolds, as the intrinsic variation in the data can be due to factors not relevant to recovery, such as participant, age, gender specific variations. In Chapter 4, we have aimed to provide limited quantitative analysis consistent with the size of our study; however the research opens the way towards a more comprehensive study.

As we move from controlled experiments to home-based long-term monitoring, the size of the data explodes by many orders of magnitude. From the perspective of computational resources as well as for purposes of convenient visualisation it is necessary to abstract out raw sensor data. Synopsis structures can act in place of the real data for specific data analysis purposes. We proposed a multi-resolution synopsis structure in Chapter 5 for activity patterns corresponding to routine. The structure is constructed by mining activity patterns at recursively smaller levels using standard pattern mining algorithms. The key departure from existing stream mining methods is the explicit incorporation of the notion of a ‘day’, a key organizing feature of human activity. In this way we have made a widely used and well-understood data mining technique specific to the field of understanding and representing human behaviour, which is a contribution of this dissertation. The data structure was shown to be able to represent routine information concisely. Similar to the ‘black-box’ concern in data-mining is the well-known problem of ‘interestingness’ of patterns found using data-mining algorithms. Previous research has proposed requiring domain-experts to provide limited feedback on the relevance of patterns

found by the algorithms. It is necessary to provide more semantic content to the patterns found by data mining algorithms for the adoption of the technology by care-providers.

Building on the techniques applied in Chapters 3-5 we aimed to exploit a growing opportunity for activity and behaviour profiling: smartphones. Usually equipped with multiple sensors that can provide activity and location information, these phones represent a ready deployment platform for pervasive sensing. Sensing on phone presents its own challenges however. Users of smartphones are particularly sensitive to the battery life of phones. The manifold embedding approach offers the opportunity to reduce the size of the data on the phone itself, transmitting only the most informative data. In Chapter 6 we presented one such system that relied on manifold embedding and clustering to profile activity over long periods of time. While there has been some research on studying specific ADL with smartphones, we consider both supervised learning approach to recognise specific activities but also clustering based profiling and data-mining over extended periods of time. To this extent, we modified the algorithms presented in Chapter 5 to also include information about significant locations. While we would advocate participant-specific clustering, it is necessary to correlate the clustered and mined information to objective data. Our limited experiment showed that it is possible to cap data manifold space which clustered in distinct regions, with ground truth information obtained from a treadmill.

While routine trees provide output useful for compression and visual exploration of the data, our goal is to be able to detect types of routine and changes in routine through the data mining approach. To this end we propose a framework for quantitative analysis of routine through the composition of two existing techniques: similarity testing of databases through frequent patterns and ensemble clustering. The former allows us to compute the distance between corresponding time-periods in two routine-trees. The latter allows the merging of information from different time-periods into a holistic picture. This picture captures both the broad differences between days, but also fine-grained differences between when the same activity is performed in the day. We applied the proposed technique to two databases. The first was a smartphone based database of everyday activity of a healthy participant. For this database we were able to demonstrate specific changes in the participant's routine by showing a temporal evolution in the routine. The second application was on data collected as part of a home-healthcare pilot application. This dataset illustrates the difficulty of pervasive healthcare system. This real-world deployment lacked labels completely. This lack of annotation resulted in the unsuitability of conventional activity recognition techniques to the data. We were, however able to show specific variations

of the participant's routine, such as day-of-the-week and seasonal dependencies. This is information that was not documented in the study, illustrating the promise of exploratory data analysis for this type of data. This dataset highlights the competing nature of priorities in this domain: for the purposes for privacy the participant details were not recorded. For the purposes of convenience and to avoid intrusion into the participant's private life, records of activity were not kept. This metadata is necessary to add meaning to the analysis. While exploratory data analysis can uncover some knowledge, a thoughtful privacy framework can help contextualise this knowledge to the participant's circumstances, and produce explanations for what factors may be triggering changes in routine. This may guide us in the design of future data collection studies as to what metadata should be recorded.

8.3 Future Work

Our research opens the way towards further research in a number of key areas:

- **Extension of Knee-Replacement Experiment:** We collected data from participants recovering from knee-replacement surgery, and applied transition data analysis to the data. This study can be extended in a number of ways. Through a larger cohort, we can aim to arrive at objective, quantitative scores of recovery instead of visual indication. Furthermore, it is necessary to compare with the current standard for such patients, such as the KOOS questionnaire [254]. While participants were asked to fill these questionnaires out, unfortunately the compliance with the questionnaire was very low, resulting in poor quality data. While this indicates the promise of sensing techniques replacing such outmoded and inconvenient mechanisms as questionnaires, it is nevertheless necessary to establish that sensing techniques can adequately replace the current standard. It would be desirable to ensure higher quality questionnaire data from participants in future studies.
- **Application of transitional activity analysis to other medical applications:** In addition to orthopaedic patients, there are other patient groups that are promising for transitional activity analysis. An example of this is tackling obesity, which is a growing problem in the developed world [120] that technology can help tackle. Ergonomics and suitability of living environments is another potential application, as our analysis could distinguish between seating types, which is relevant for rehabilitation. Limb amputation is likely to have similar data characteristics as the knee replacement study performed in

Chapter 4, with important distinctions related to participant demographics and their recovery profiles. Multiple sclerosis, Parkinson's and other neuro-degenerative ailments also produce distinctive movement profiles, which could be studied in terms of transitional activity. Currently clinical studies are planned that will extend the work presented in Chapter 4 focusing on limb amputation.

- **Generating objective scores through analysis of activities:** With both ADL and transitional activities, it may be much more accurate to develop a score of the participant's health state based on the performance of a combination of activities. While we analysed individual activities for impairment in Chapters 3 and 4, it is likely that combining information from multiple activities will reduce error rates. As our manifold framework can be used to analyse ADLs along with transitions, one promising direction of research will be extending our framework to output a single composite score correlated with an objective clinical measurement.
- **Clinician directed visualisations of mined information:** Our proposed techniques can help visualise large quantities of sensor data, which is a requirement of all stakeholders in pervasive healthcare systems, particularly clinicians. The manifold view of transitions, for instance, summarises time-series information from 36 features in a small number of dimensions. Several graphs were presented in Chapter 4 to visualise changes in transition performance. While these graphs are meaningful to engineers, there is a need to bring this information to clinicians directly. One approach for this is developing a better understanding for what the intrinsic dimensions plotted in Chapter 4 correspond to, and explaining the manifold in those terms. The routine tree also helps visualise a large amount of sensor data into a single, multi-resolution structure showing both broad trends, and finer detail. It can characterise behaviour in terms of time and location, which is also information relevant to all stakeholders. There is a need to develop this further into a system that is readily understood by clinicians, possibly through a semi-supervised approach that can associate semantic labels with the mined data. In case of the quantitative analysis of routine, once again while changes in routine can be plotted in graphs, the visualisations will be more useful if they can be associated with actionable intelligence for the stakeholder: why may routine be changing? With more context, it may be possible to uncover associations between such changes and personal, social and environmental factors.
- **Generating probabilistic models from mined data:** In Chapter 2 we surveyed a wide range of probabilistic techniques for modelling behaviour. The scope of these

techniques is typically limited; i.e. they aim to model specific types of behaviour. While our technique is useful for exploratory data analysis, it will be interesting to extend the multi-resolution synopsis into a hierarchical model of routine.

- **Incorporating ‘Big Data’ techniques into routine-tree algorithm:** Increasingly technology is trending away from databases and towards file-based models. Google has led the way in this by presenting a model for processing large quantities of data through the Map-Reduce model [255], that relies on a simple model of computation that lends itself to efficient parallelisation. This class of technology is colloquially known as ‘Big Data’, and there exist a number of powerful languages (such as Hadoop [256]) that can take data in this form and perform the analysis efficiently on a cluster of computers. As our number of users increases, it may be desirable to translate the mining into this paradigm to take advantage of computing efficiencies.
- **Associating smartphone data features with clinical measurements:** To convince smartphone users to use the application it is necessary to provide feedback meaningful to them. At present our application provides an abstract measure of activity by clustering the sensor data. A next step would be correlate phone data to well-understood measures such as calories burned. This can be done by collecting smartphone data simultaneously with ‘gold standard’ energy output measurements such as Doubly Labelled Water (DLW). We are currently conducting such a study on a small cohort of participants and anticipate further research in this direction.
- **‘Refactoring’ analysis:** A software development practice that emerged as a result of decades of practice was a set of techniques designed to re-engineer existing components to improve the quality of the components without necessarily adding functionality. As activity and behaviour profiling research explodes, there is a dire need for research housekeeping that consolidates on the technical gains to map to medical applications, and ensure that the medical applications themselves lend themselves to a pervasive sensing paradigm where the involvement of the medical expert with the participant would be minimal. One goal of our research will be to ‘refactor’ our analysis to ensure the framework meets specific medical needs and can be deployed in real-world pervasive applications. This requires the bringing together of machine learning and software engineering research, and the implementation of algorithms suitable for actual deployment. We performed some of this work in Chapter 6, where the algorithms were geared towards deployment on a realistic, marketable smartphone application that could not a) transmit a lot of data and b) perform CPU intensive computations.

- **Managing system resources based on user state:** Not all states of participants are equally important. As we develop algorithms for more precise labelling of participant states, if utilities of these states are available it is important to optimise the sensing to extract high utility data at high resolution, and either discard or perform lossy compression on unimportant data. We proposed such a system in [17] for a set of sensors. Applying this type of mechanism in the smartphone application would seem to be particularly promising, given the imperative to maximise battery life.
- **Incorporating metadata:** As pervasive sensing systems are mainstreamed into the healthcare delivery services, integration with Electronic Medical Record systems used by the healthcare service may be feasible. This will provide a wealth of metadata for each participant. Exploratory data mining can exploit this metadata to find richer, more meaningful patterns.
- **Ensuring privacy:** Connected to the above concern is that of privacy. While we have certain built-in privacy preserving steps in our framework (such as abstract activity levels instead of specific activity profiling in the routine tree), in practice privacy is preserved in our work so far through anonymisation. If more metadata becomes available, it will be important to assess privacy concerns, and where necessary to build in stronger constraints in the systems and algorithms. For instance, it may be necessary to sacrifice data quality or completeness in the smartphone applications to preserve privacy. In a well-reported case of a similar fitness tracking system Fitbit [257], users were angered by the revelation of sensitive personal activity accessible through search engine. Our similar system for activity profiling will need to learn from such incidents. For each privacy concern, there needs to be a sufficient justification that users understand, and choose to benefit from. As noted in Chapter 1, although privacy remains a factor in adoption of pervasive sensing, people can be amenable to trading some privacy when benefits are apparent [14], particularly to ensure independent living and where there are risks of acute episodes.

8.4 Conclusions

This thesis advanced the state of the art in activity and behaviour profiling in the following areas:

- Development of a methodology for the automatic detection and analysis of transitional activities. Application of the proposed method to the real-world clinical application of post-operative recovery.
- Development of a synopsis structure to represent an individual's routine visually. This has the effect of simplifying sensor data, and reducing the server load. Development of algorithms for quantitative analysis of routines using this mechanism.
- Application of short-term activity analysis, and long-term behaviour profiling to wearable sensors, as well as translating research from sensors to smartphone based apps. Adaptation of analysis algorithms to the phone system, increasing the attractiveness of the software for end-users and therefore potentially leading to wider adoption of pervasive sensing.

8.5 Acknowledgement

Support for this research came primarily from two projects, Smart and Aware Pervasive Environments (SAPHE) [5] and Elite Sport Performance Research in Training with Pervasive Sensing (ESPRIT) [232].

SAPHE brought together research leaders from industry and academia into a consortium for developing and deploying a pilot telemedicine system. The project was funded by the Technology Strategy Board. The system included the wearable e-AR sensor as well as a number of ambient devices, and was deployed at the home of participants requiring care from community matrons. Contributions of this dissertation towards this project include the analysis of SAPHE data through data mining of routine activity.

ESPRIT is a currently ongoing project that brings academic research to elite sports performance, with a view towards improving the performance of athletes. This has importance in particular with respect to the London Olympics of 2012. The remit of the project includes both investigations into improving the performance of professional athletes participating in Olympic sports, but extending the legacy of the Olympics by encouraging the translation of the technology developed for the project to the general public. This includes health and well-being applications. The Active Miles project contributes to this effort through the software and data

analysis tools developed on a platform widely in use. The ESPRIT project is funded by the Engineering and Physical Sciences Research Council.

We are grateful for the support of the sponsors, which enabled our research.

Appendix A

Manifold Learning

A.1 Introduction

Manifold learning is an important class of non-linear dimensionality reduction algorithms that can uncover the intrinsic dimensions of a dataset. This appendix provides background on manifold learning, which was used for transitional activity recognition in Chapter 4. The motivation for modelling with manifolds is reviewed, followed by brief description for some of the widely used manifold embedding algorithms and a discussion of the tradeoffs between them.

A.2 Motivation

Data sources for machine learning and data mining can have large dimensionality. The raw data may be high dimensional itself, or a large number of features (see Table 2.2 in Chapter 2) may have been extracted. This can result in the so-called ‘curse of dimensionality’ [258]. With such datasets machine learning algorithms may fail to converge to an accurate solution and have weaker statistical significance guarantees [259]. Higher dimension data typically need more training data [260] for learning algorithms to generalise. The performance of a machine learning algorithm can be measured through the correspondence of the error obtained during training E_{in} with the error E_{out} of the trained learner on out of sample data. The Vapnik–Chervonenkis (VC) inequality [261] bounds the probability of the difference in errors as

$$P[|E_{in} - E_{out}| > \varepsilon] \leq 4 m_H(2N) e^{-\frac{1}{8} \varepsilon^2 N} \quad (\text{A.1})$$

where

$$m_H(N) = \sum_{i=1}^d \binom{N}{i}$$

where ε is the user-specified error threshold, N is the number of training points, and d is the VC dimension. The last is a measure of the complexity of the learner. This is generally influenced by the dimensionality of the data. For instance, for a simple linear classifier such as a perceptron, the VC dimension is equal to the dimensionality of the data [262]. The linear model is the simplest of those commonly used in machine learning. Higher complexity models have more degrees of freedom, and therefore an even greater VC dimension. Figure A.1 shows the minimum number of training points required to achieve a low bound on error (specifically $\varepsilon = 0.2$ and $P[|E_{in} - E_{out}| > \varepsilon] = 0.2$) as dimensionality increases. It is assumed that a perceptron is being used for learning. It can be seen that increasing dimensionality proportionately increases the number of training points required to guarantee generalisation.

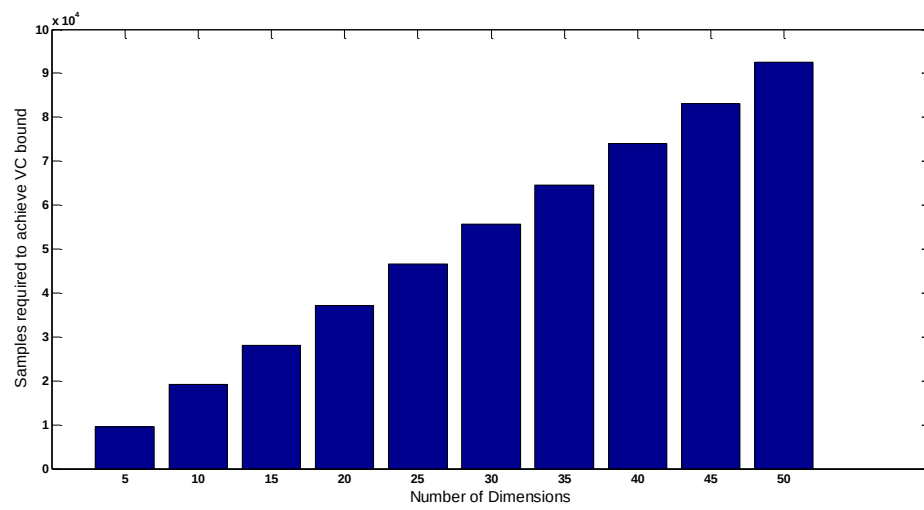


Figure A.1 The number of samples required to achieve a fixed Vapnik–Chervonenkis (VC) generalisation bound, with increasing dimensionality. Higher dimensionality results in significantly increased training data requirements.

There are two main strategies for dealing with high dimensionality: Feature Selection, and Dimensionality Reduction. As the latter is used in important components of this dissertation, it is the focus of this chapter.

A.3 Dimensionality Reduction

Dimensionality reduction seeks to capture the information represented in the complete feature space by mapping to a lower dimensionality space. Commonly used dimensionality techniques such as Principal Components Analysis (PCA) can be effective for activity profiling data (as demonstrated in Chapter 3). A non-linear approach, called manifold learning, can improve on linear techniques for certain applications.

To illustrate the problem, consider the helix dataset shown in Figure A.2. The data can be generated by the equations shown below:

$$\begin{aligned}x(t) &= bt \\y(t) &= a \sin(t) \\z(t) &= a \cos(t)\end{aligned}\tag{A.2}$$

where a and b are constants, and t is a parameter. The helix shape is overlaid with a set of points, $P_1, P_2, \dots, P_{12}$. These points are in order of increasing t . Given t it is possible to determine x , y and z , therefore the dataset has an intrinsic dimensionality of 1.

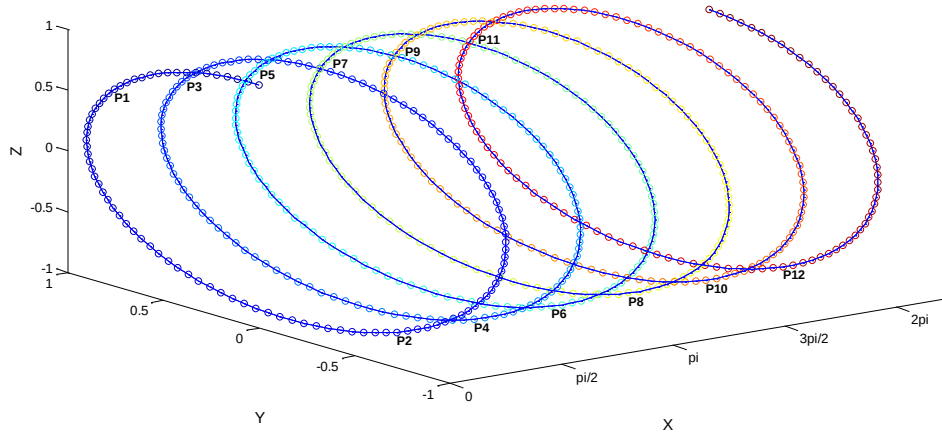


Figure A.2 Helix dataset generated through equations A.2, with a set of points overlaid with increasing t .

We will seek to preserve relationships between each pair of points when representing the data in fewer dimensions.

Several well-known algorithms such as Principal Component Analysis (PCA) [102] and Multidimensional Scaling (MDS) [103] map data into a lower dimensional space while preserving existing linear relationships in the data. MDS for example preserves the pair-wise Euclidean distance between data points. For any two points x_a and x_b defined in n dimensions, the Euclidean distance between them is

$$d_n(x_a, x_b) = \sqrt{\sum_{i=1}^n (x_{a_i} - x_{b_i})^2} \quad (\text{A.3})$$

The lower dimensional space found by MDS reflects pair-wise Euclidean distances in the full feature space. MDS can be expressed as finding a space in m dimensions that minimises a cost function that aggregates the mapping cost c for each pair of points. For any two points x_a and x_b , $c(x_a, x_b)$ can be computed as

$$c(x_a, x_b) = (d_m(x_a, x_b) - d_n(x_a, x_b))^2 \quad (\text{A.4})$$

Equation A.4 can be solved by a Singular Value Decomposition (SVD) analysis of the distance matrix D_n . Figure A.3 shows the helix data embedded in 2 dimensions using MDS. While the shape of the data initially looks similar to the helix in 3 dimensions, on closer scrutiny the distortion is apparent. The linear technique attempts to force the 3-d shape into 2-d, without regard for the structure of the data. Local relationships between pairs of points are not preserved in the 2-d space, e.g. $P1$ is closer to $P3$ than $P2$, and $P9$ is closer to $P6$ than $P10$. The reason for this is that if we disregard the originating equations, and the underlying structure, the MDS mapping reflects the distances between points in free-space. $P1$ is closer to $P3$ than $P2$ in Figure A.2, if distances were measured between each pair of points with straight lines. If these points were presented to us without equations A.2, the linear mapping would be a reasonable first analysis. A line from $P1$ to $P3$ will however, pass through points that cannot satisfy equations A.2. These points are not on the helix, which only comprises of points generated by the helix equations with varying t . This ‘intrinsic dimension’ of the system has not been preserved by MDS.

Consider if we traversed along the helix when considering relationships between points. The distance between $P1$ and $P3$ would not be that of a straight line, but a path along the helix surface. This would correspond to the parameter t . Manifold embedding techniques seek to find such intrinsic dimensionality, by modelling the shape of the dataset, and preserving relationships between points in context of this shape. Figure A.4 is generated using Isomap [188], one of the

best known manifold embedding techniques. Relationships between the labelled points are preserved. The first dimension captures information completely, suggesting there is one ‘intrinsic’ dimension underlying the data.

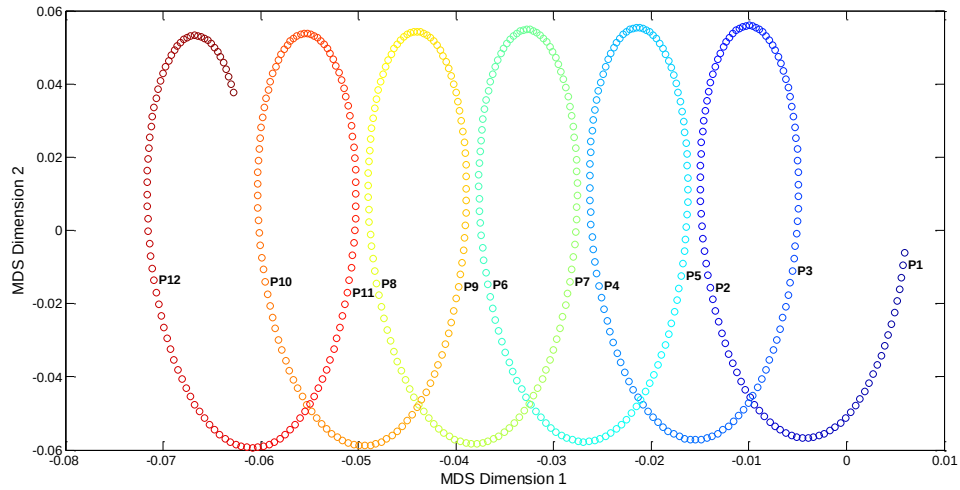


Figure A.3 Helix dataset generated through equations A.2, embedded in two dimensions using Multi-dimensional Scaling (MDS). Linear dimensionality reduction fails to preserve the order of points.

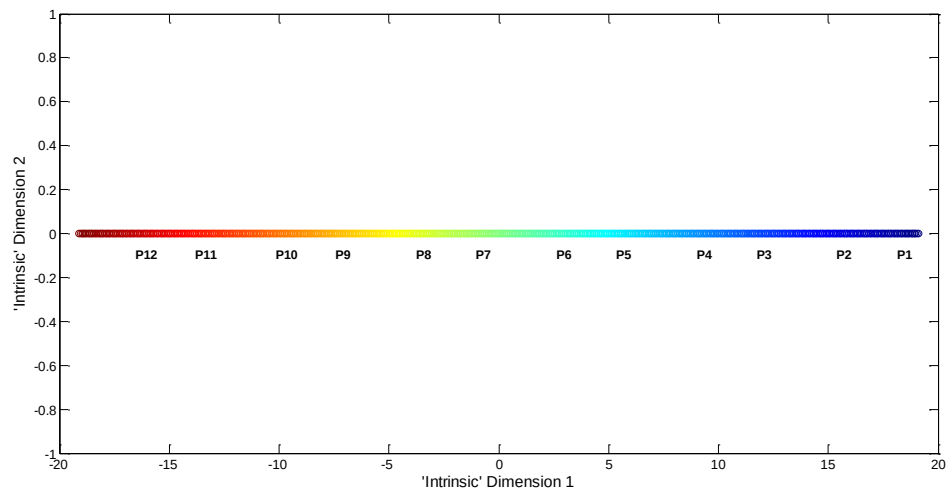


Figure A.4 Helix dataset generated through equations A.2, embedded in its ‘intrinsic’ dimensionality, using Isomap. As the first dimension captures the information for parameter t , the second dimension is redundant.

A.4 What are Manifolds?

Manifolds are a class of geometric objects used to describe complex surfaces through neighbourhood relationships. The crucial property enabling modelling and analysis is a neighbourhood (a subset of proximal points) associated with each point with well defined, typically Euclidean properties. Formally, each such neighbourhood in a m -dimensional manifold is homeomorphic¹ to Euclidean space $\mathbb{R}^m$. This property allows the use of well-studied properties of linear surfaces, but permits a global, non-linear shape. Manifolds are very well studied, with mathematically defined properties of differentiability, orientation, continuousness. The reader is referred to [263] for a comprehensive overview. The relevant property for our analysis is manifold embedding: a mapping of feature-space points to points in a lower dimensionality space that preserves properties of the manifold.

Manifolds are suitable for use where high-dimensional data is generated by variation in a smaller set of parameters. In context of the sensing domain, consider inertial sensor data collected from a human participant. While many features can be extracted from the sensor data, the variation in the features is generated through a small set of parameters pertaining to how people move. This can be considered analogous to the role of the t variable in equations A.2.

Given a dataset D a manifold can be approximated by selecting around each point a neighbourhood. This neighbourhood is typically found by selecting k points closest to each point. Alternatively the neighbourhood can be selected based on a distance threshold. The quality of approximation is influenced by this parameter, and may require empirical validation [264]. Distances between points are along the manifold surface, and are called geodesic distances. These can be significantly different from Euclidean distances, as was seen in the embedding of the helix surface.

Manifold models are often represented using a graph data structure. Vertices correspond to data points, each of which has an edge to points in its neighbourhood. The edge weight is equal to the Euclidean distance.

¹ A homeomorphism between topological spaces is a continuous function between them that has a continuous inverse function.

A.5 Dimensionality Reduction with Manifold Embedding

The challenge in manifold embedding is to find a space that represents a graph representing a manifold. Strategies for this vary in terms of computational complexity, accuracy of representation and sensitivity to parameters. Some of the well known algorithms for manifold embedding are summarised below, with references for further reading.

- **Isomap** approximates geodesic distances on the manifold through shortest paths on a graph representation of the manifold. The matrix of pair-wise graph distances is reduced using MDS. Formally, if $d_g(x_a, x_b)$ specifies the distance of the shortest paths between points x_a and x_b , Isomap solves using MDS the cost function

$$c(x_a, x_b) = (d_m(x_a, x_b) - d_g(x_a, x_b))^2 \quad (\text{A.6})$$

- **Locally Linear Embedding (LLE)** [196] finds in the complete feature space a weight vector for each point that can reconstruct it from its neighbours. A low-dimensional embedding that reflects these weights is then found. The cost function can be specified as finding a set of points Y such that given a weight matrix W specifying the weights needed to reconstruct a point from its neighbours, the following cost function is minimised

$$C(Y) = \sum_i \left| y_i - \sum_j W_{ij} y_j \right|^2 \quad (\text{A.7})$$

The cost function can be computed efficiently by converting into a matrix formulation. Equation A.6 can be rewritten as

$$C(Y) = Y^T M Y \quad (\text{A.8})$$

where

$$M = (I - W)^T (I - W)$$

The cost function is minimised by the eigenvectors of M . As W is mostly zero, sparse matrix computations can be used, making the algorithm memory and computation efficient.

- **Laplacian Eigenmaps** [197] relies on spectral graph theory methods for studying connectivity properties of graphs through matrix decomposition. Based on the graph's degree matrix D and adjacency matrix A , the Laplacian matrix the Laplacian matrix $L = D - A$ is computed. The eigenvectors of this matrix were used for spectral clustering, as in Chapter 4, and are used for embedding in Laplacian Eigenmaps. The adjacency matrix can be constructed in the usual way by setting 1 for neighbouring vertices and 0 otherwise. An alternative formulation is proposed where the

neighbourhood edge weights are specified by a Gaussian heat kernel. In this case the weight between neighbouring nodes x_i and x_j is specified by:

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (\text{A.9})$$

This construction allows a mathematical justification for the embedding, however requires the user to specify σ . The Laplacian Eigenmaps algorithm has been shown to be equivalent to LLE under certain assumptions in [197].

- **Local Tangent Space Alignment (LTSA)** [265] computes at each point a low-dimensional space based on its neighbourhood using PCA. The local spaces are subsequently aligned to compute the global space. The alignment is performed by optimising a cost function that allows any linear transformation of each local space.

In addition to the above, several other approaches have been proposed for manifold embedding [266-268]. Hessian LLE [266] provides stronger theoretical guarantees on global optimality than LLE, although the complexity is also significantly higher. Instead of learning a regression model for each neighbourhood, Hessian LLE generates tangent spaces for them using PCA, subsequently aligning them using the Frobenius norm of the Hessian matrix. Local Multidimensional Scaling [269] performs MDS in local regions, and uses convex optimisation to fit them together. The choice of algorithms can be predicated on a number of factors, described below.

A.5.1 Quality of Embedding

Manifold embeddings can usually be categorised as *isometric* or *conformal*. Isometric embedding algorithms assume a map from manifold space to feature space that preserves inter-point distances, and seek to discover it. Conformal embedding algorithms preserve local neighbourhoods and angles between them (i.e. find conformal maps [270]). These categories specify the mappings the algorithms seek to achieve with varying theoretical assumptions, however performance in practice may not preserve pair-wise distances or angles.

Amongst the algorithms discussed, Isomap and Hessian LLE are isometric. LLE is conformal, as are Laplacian Eigenmaps. Isometric mappings are generally more computationally intensive

than conformal mappings, however they offer the promise of discovering the true parameter space. Algorithms may also be categorised as local or global, based on the information used to reconstruct each point. Isomap is a global technique, as the matrix that is decomposed has distances between each pair of points. In contrast LLE only seeks to reproduce regression parameters for manifold neighbourhoods in a lower dimension. Local algorithms are better at preserving neighbourhood relationships than global algorithms, however may result in distant points on the manifold embedded close together.

The algorithm used in our work, Isomap is isometric and global. Furthermore under certain assumptions, it is also proven to recover the parameterisation of the manifold. The most important assumption is the approximation of geodesic distances with graph distances. This relates to the sampling characteristics of the dataset. If the sampling is sufficiently dense and uniform, graph distances approach geodesic distances. A further advantage of Isomap is that it may allow an estimation of the intrinsic dimensionality based on the eigenvalues computed by MDS.

A.5.2 Complexity

Generally local algorithms have lower computational complexity than global ones. Isomap in particular, has high computational cost because of the need to compute shortest paths between each pair of points, which has a complexity of $O(N^2)$. Subsequently eigenvalue analysis is performed on a full-matrix, which takes a further $O(N^3)$ time. Although most manifold embedding algorithms perform such matrix decompositions, some (including LLE and LTSA), can take advantage of sparse matrix calculations. A sparse representation also reduces memory requirements, which may be prohibitive for large datasets.

Some of the complexity of Isomap can be reduced by computing the manifold embedding with a smaller set of points (called landmarks), and subsequently training a regression algorithm to map the remaining points to the embedding. This approach, called Landmark-Isomap [239] however loses the optimality guarantees of Isomap.

It may also be appropriate here to discuss implementation complexity. While Isomap has high computational cost, it is one of the easier algorithms to implement. It relies on well-understood

algorithms such as shortest-path and MDS, which are already available as well-tested, optimised libraries. Other global, isometric algorithms such as Hessian LLE and Semidefinite Embedding [271] have more complex implementations relying on higher order calculus and semi-definite programming respectively. This may be a reason why Isomap is currently more widely used, and therefore has better studied performance properties.

A.5.3 Sensitivity to Training Conditions

The approximation of a manifold's neighbourhoods is through selecting the k nearest points around each point, from a given sample. There are two factors in this approximation effect the quality of the approximation. The first is application specified: the size and distribution of the sample limits the neighbourhoods that can be constructed, which may not be representative of the true structure. The second is the user specified k neighbourhood parameter. Non-uniform or sparse sampling and unsuitable k can result in 'shortcut edges': edges between points distant on the manifold. This in turn may distort the embedding. Isomap, for instance, has been found to be vulnerable to distortions resulting from shortcut edges [264]. All geodesic distances approximated with paths including a shortcut edge are incorrect. A single incorrect edge propagates errors globally. Detecting and correcting for shortcut edges is an active research area, particularly with respect to the Isomap algorithm [272-274]. Local algorithms tend not to be significantly impacted by shortcut edges [275], as the error does not propagate through the manifold model. This resilience to approximation error is an attractive property of local methods, in addition to their generally faster execution.

A.6 Conclusion

Manifolds are a useful model for high-dimensional data that is generated by a small set of parameters. Examples of applications include image processing and matching [276], text mining [277] and sensor localisation [278]. The choice of manifold embedding algorithm is determined by application requirements. If, for example, global embeddings are too time-consuming or not needed, local approximations such as LLE and Laplacian Eigenmaps are good candidates. Conversely, if stronger guarantees of optimality are desired, Isomap and Hessian Eigenmaps should be considered.

One aspect of manifold embedding and dimensionality reduction in general is that intrinsic dimensions are difficult to explain to end users of machine learning systems. Manifold embedding will render the dataset into a readily visualised form, which may be more suitable for classification and clustering. It is impossible, however, to say what the discovered intrinsic dimensions mean. Where this is important, a feature selection approach may be more feasible.

Manifold learning is popular research area, with several promising aspects of development. Foremost of which is that nearly all algorithms require the specification of a neighbourhood size parameter. There remains a need to develop computationally inexpensive methods for selecting the neighbourhood size. As noted earlier, for global algorithms such as Isomap, shortcut edges are problematic and need to be detected. An important research question is assessing in a principled way when data does lie on a manifold. This is currently largely up to the researcher, and is justified through empirical validation. In our work we relied on demonstrating superior performance for activity transitions when compared to linear methods, through visualisation and residual variance analysis.

Appendix B

Wavelet Analysis

B.1 Introduction

Wavelets [83] allow a means of representing signal in multiple resolutions, incorporating both time and frequency information. They are a widely used signal processing paradigm, and have been shown to be promising features for activity recognition [97, 171, 172]. The activity profiling methods proposed in Chapters 3 and 4 include wavelet feature extraction. This chapter provides background on wavelets and references for further reading.

B.2 Motivation

Signal processing has long relied on mathematical transformations of input signals to uncover information not readily available in the raw form. Signals, loosely defined, are sequences of values of a parameter, often evolving over time. The natural representation of a signal is referred to as its ‘time domain’ representation. In the early part of the 20th century, a mathematical transformation was developed, allowing representation of signals in ‘frequency domain’. This transformation, known as the Fourier Transform, represents a signal as an aggregation of periodic (sin and cosine) signals of varying frequencies, referred to as frequency components. The frequency domain signal can be cast back into the time domain through an inverse transform. Given a signal x comprised of N points (indexed 0 to $N - 1$) the Fast Fourier Transform (FFT) [279] can be computed as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k \frac{n}{N}} \quad (\text{B.1})$$

where $k \in 0, 1, \dots, N - 1$. The FFT can be computed in $O(N \log N)$ time. Figure B.1(a) shows the signal from an accelerometer up-down channel sampled at 50Hz as a participant walks.

Figure B.1(b) shows the power of the FFT components plotted against periods in seconds/cycle. The spike shows associated with the dominant cycle in the data (between periods 1 and 2) corresponds to the pace of walking.

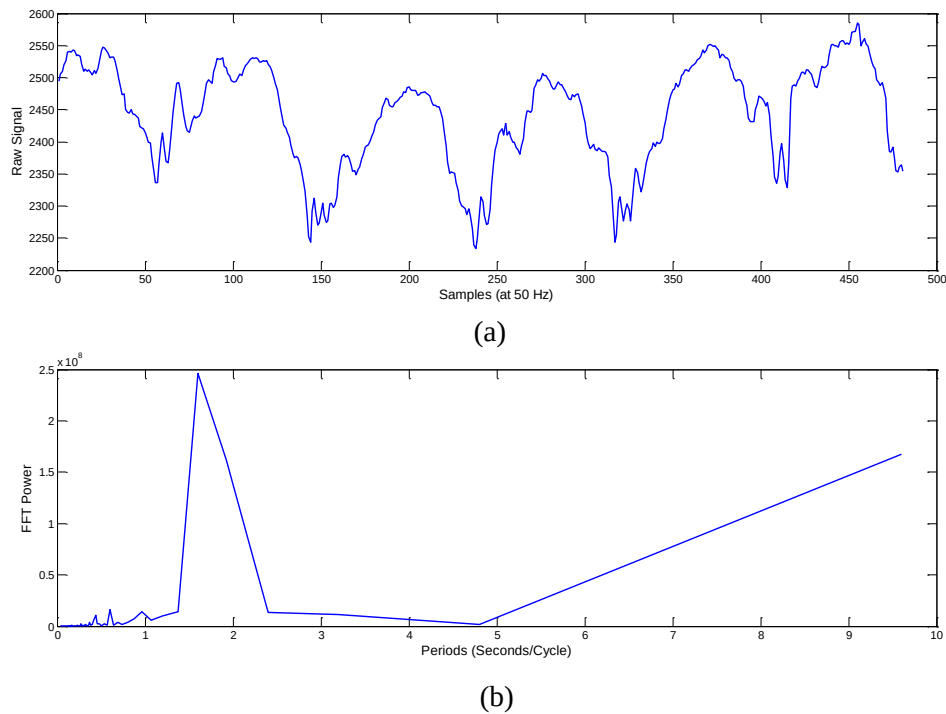


Figure B.1 FFT analysis of signal from accelerometer Up-Down channel (a) sampled at 50Hz worn by a walking participant. The power of the FFT components spikes close to 2 seconds/cycle, indicating the dominant pattern in the data

FFT analysis is very effective for data with significant periodicity throughout the signal. It has limitations however, in representing local frequency content. This is because, there is no representation of time in the frequency domain. If a signal contains cycles in certain places, it is impossible to identify where that occurs in a FFT view. An early solution to this was to apply the transform in fixed sized windows. Wavelets extend this notion, by composing the signal in terms of ‘wavelets’ of varying scale. This allows representation of both time and frequency information.

B.3 Wavelets

Wavelets are curves containing one or more oscillations confined in a finite interval. They are defined by so-called ‘mother functions’ that determine the shape of the curve, and are designed with mathematical properties that permit decomposition of signals. One of the most commonly used wavelets is the Haar wavelet shown in Figure B.2.

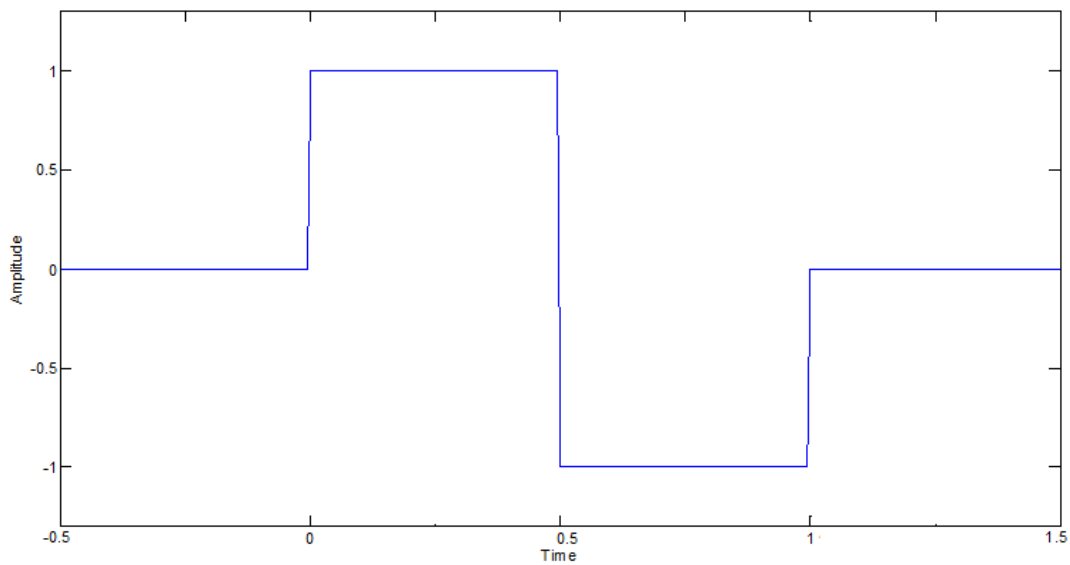


Figure B.2The Haar Wavelet

The Haar wavelet is specified by the equation

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.2})$$

The shape of the wavelet is important to the extent that it can be used to compose the signal. Several other wavelets have been proposed, including Daubechies, Morlet and Mexican Hat wavelets. The shape of the wavelets produces differing properties in the resulting transformations, and is chosen in a domain specific manner [280].

B.4 Wavelet Transform

During a wavelet transformation a windowed basis function (based on the mother function) is progressively applied to achieve coefficients at multiple levels. The data can be reconstructed through an inverse transformation of the wavelet coefficients.

Like FFT, the signal is decomposed into constituents; however wavelets differ by composing the signal in terms of waves with progressively smaller size. In particular, the Discrete Wavelet Transform (DWT) can provide a multi-resolution decomposition of the signal, and can be computed efficiently. The signal is decomposed using a wavelet defined by a ‘mother wavelet’ and a scale. The DWT is defined by the following equation:

$$W(s, i) = \sum_s \sum_i x_i 2^{-s/2} \psi(2^{-s}n - i) \quad (\text{B.3})$$

where s represents the scale, i represents time, and ψ refers to the basis function defining the wavelet, which is called the mother function. The scale varies inversely with the frequency. The mother function for the Haar wavelet transform is given in equation B.2. The choice of mother function is a significant design decision.

The transform can be implemented efficiently through cascading algorithms that reduce the volume of data to be processed iteratively. At each scale, high-pass and low-pass filter is applied. Let H denote high-pass filter and G denote low-pass filter. Figure B.3 shows the Mallat algorithm that can be used for fast computation of the DWT. At each scale s , the high pass filter produces the signal detail d_s , while the low pass filter produces approximation coefficients a_s . After one application of the filtering, half the frequency of the signal has been removed. Therefore, half of the samples can be eliminated according to the Nyquist’s rule. The signal can be subsampled by 2 at each step (denoted in Figure B.3 by $\downarrow 2$). The filtering and decimation process can be continued for a fixed resolution, or based on the length of the signal. There can be at-most $\text{floor}(\log_2(n))$ levels. Lower order coefficients represent the overall shape of the sequence, and higher order coefficients inform about local variations.

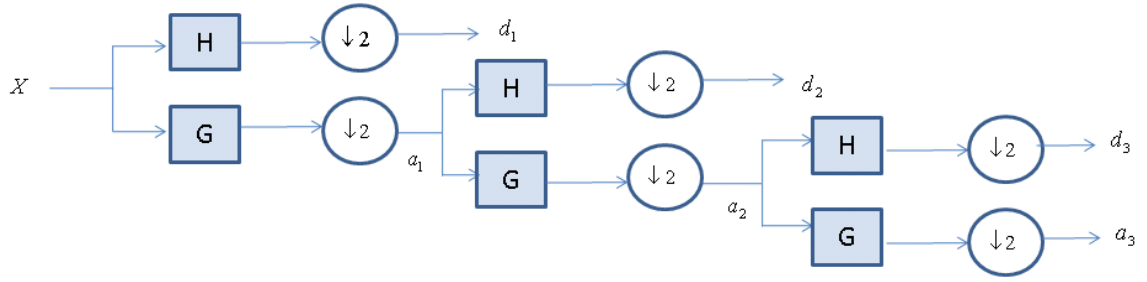


Figure B.3 Filter banks for computing DWT using the Mallat algorithm showing a filter bank of three scales.

The mother function used for computing the DWT determines the filters H and G . The filters for the Haar wavelet are defined as:

$$H = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0 \\ -\frac{1}{\sqrt{2}}, & n = -1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.4})$$

$$G = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0, -1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.5})$$

Figure B.4 shows the result of successive application of the wavelet transform to the signal shown in Figure B.1(a). Wavelets can be used for compressing the data by discarding coefficients with little information (the higher order detail coefficients for example). Furthermore these coefficients also are likely to contain small, high-frequency variations, which can be seen as noise, and can be ignored to smooth the signal.

The size of the data is reduced by half with each successive application of the transform. In Chapters 3 and 4, statistical features of these signals (mean and standard deviation) were used as features for activity profiling.

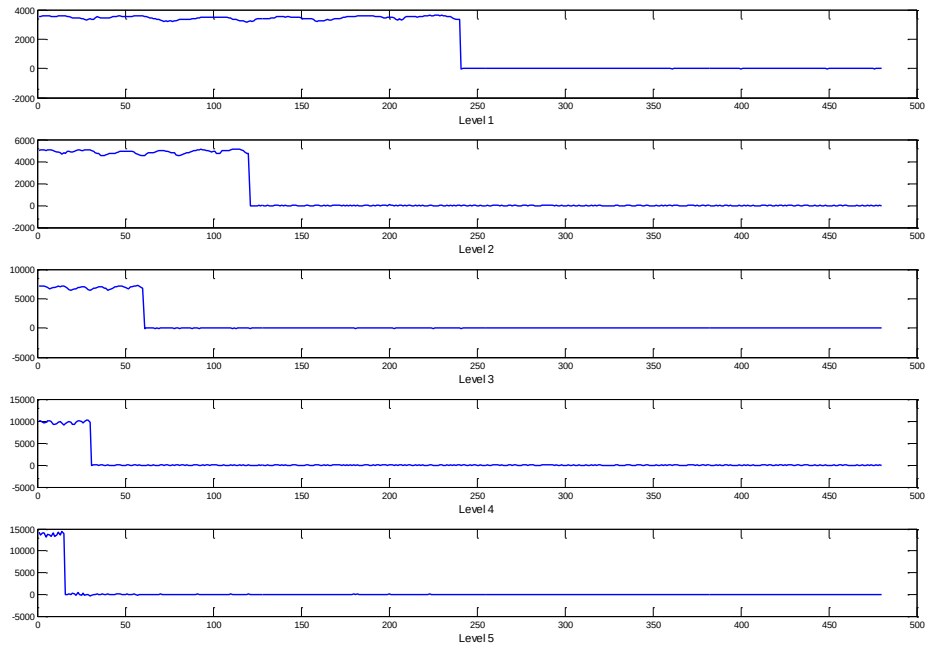


Figure B.4 Successive application of the Haar wavelet transform to the accelerometer signal shown in Figure B.1(a). Each application of the transform decimates information in the resulting signal

B.5 Conclusion

This chapter provides a short background on the wavelet technique used in this thesis. The reader is referred to Burrus et al. [281] for further background.

Wavelets transforms can represent time and frequency information by convoluting a signal with ‘waves’ of small intervals. The theory build on the Fourier transform, which utilises fixed size periodic signals for the transformation. By incorporating temporal resolutions, both time and frequency information can be represented. In using wavelets as features, the aim is to succinctly capture both time and frequency domain information into the analysis.

References

1. Yang, G.-Z., *Body Sensor Networks* 2006: Springer-Verlag New York, Inc.
2. *World health report*, 1998, World Health Organization.
3. Yach, D., et al., *The global burden of chronic diseases: overcoming impediments to prevention and control*. JAMA, 2004. **291**(21): p. 2616-22.
4. Aziz, O., et al., *A Pervasive Body Sensor Network for Measuring Postoperative Recovery at Home*. Surgical Innovation, 2007. **14**(2): p. 83-90.
5. ElHelw, M., et al. *Smart and Aware Pervasive Health Environments (SAPHE)*. 2009; Available from: <http://www3.imperial.ac.uk/pls/portallive/docs/1/35451697.PDF>.
6. Atallah, L. and G.-Z. Yang, *The use of pervasive sensing for behaviour profiling — a survey*. Pervasive and Mobile Computing, 2009. **5**(5): p. 447-464.
7. Verghese, J., et al., *Abnormality of Gait as a Predictor of Non-Alzheimer's Dementia*. New England Journal of Medicine, 2002. **347**(22): p. 1761-1768.
8. Atallah, L., B. Lo, and G.-Z. Yang, *Can pervasive sensing address current challenges in global healthcare?* Journal of Epidemiology and Global Health, 2012.
9. James, P., A. Raza, and Y. Guang-Zhong, *Wireless Sensor Network*, in *Handbook of Healthcare Delivery Systems* 2010, CRC Press. p. 46-1-46-18.
10. Arnrich, B., et al., *Pervasive healthcare: paving the way for a pervasive, user-centered and preventive healthcare model*. Methods Inf Med, 2010. **49**(1): p. 67-73.
11. Murphy, E., *Case management and community matrons for long term conditions*. BMJ, 2004. **329**(7477): p. 1251-1252.
12. Orwell, G., 1984/1983: Houghton Mifflin Harcourt.
13. Hoffman, S., *Electronic Health Records and Research: Privacy Versus Scientific Priorities*. The American Journal of Bioethics, 2010. **10**(9): p. 19-20.
14. Beach, S., et al., *Disability, age, and informational privacy attitudes in quality of life technology applications: Results from a national web survey*. ACM Transactions on Accessible Computing (TACCESS), 2009. **2**(1): p. 5.
15. ElHelw, M., et al. *An integrated multi-sensing framework for pervasive healthcare monitoring*. in *3rd International Conference on PervasiveHealth 2009*. . 2009.
16. Seungwoo, K., *A Scalable and Energy-Efficient Context Monitoring Framework for Mobile Personal Sensor Networks*. IEEE Transactions on Mobile Computing, 2010. **9**: p. 686-702.
17. Panangadan, A., S.M. Ali, and A. Talukder, *Markov decision processes for control of a sensor network-based health monitoring system*, in *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3* 2005, AAAI Press: Pittsburgh, Pennsylvania. p. 1529-1534.
18. Rahman, M.A., A. El Saddik, and W. Gueaieb. *Data visualization: From body sensor network to social networks*. in *Robotic and Sensors Environments, 2009. ROSE 2009. IEEE International Workshop on*. 2009.
19. Mahapatro, A. and P.M. Khilar. *Online fault detection and recovery in body sensor networks*. in *World Congress on Information and Communication Technologies (WICT)*. 2011.
20. Saeed, A., A. Stranieri, and R. Dazeley, *Fault-tolerant data aggregation scheme for monitoring of critical events in grid based healthcare sensor networks*, in *Proceedings*

- of the 19th High Performance Computing Symposia 2011, Society for Computer Simulation International: Boston, Massachusetts. p. 56-64.
21. Ramachandran, U., et al., *MediaBroker: A pervasive computing infrastructure for adaptive transformation and sharing of stream data*. Pervasive Mob. Comput., 2005. **1**(2): p. 257-276.
 22. Atallah, L., et al. *Wirelessly accessible sensor populations (WASP) for elderly care monitoring*. in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*. 2008.
 23. Welbourne, E., et al., *Challenges for Pervasive RFID-Based Infrastructures*, in *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops 2007*, IEEE Computer Society. p. 388-394.
 24. Ellul, J., B. Lo, and G.-Z. Yang, *The BSNOS Platform: A Body Sensor Networks Targeted Operating System and Toolset*, in *The Fifth International Conference on Sensor Technologies and Applications 2011*.
 25. Varshney, U., *Pervasive Healthcare Computing: Emr/Ehr, Wireless and Health Monitoring 2009*: Springer.
 26. Payne, T.H., et al., *National-scale clinical information exchange in the United Kingdom: lessons for the United States*. J Am Med Inform Assoc, 2011. **18**(1): p. 91-8.
 27. Maglogiannis, I., *Introducing Intelligence in Electronic Healthcare Systems: State of the Art and Future Trends*
- Artificial Intelligence An International Perspective*, M. Bramer, Editor 2009, Springer Berlin / Heidelberg. p. 71-90.
28. Selden, C.R. and B.L. Humphreys, *Unified Medical Language System: Current Bibliographies in Medicine, January 1986 - December 1996* 1997: Diane Publishing Company.
 29. Heidegger, M., J. Stambaugh, and D.J. Schmidt, *Being and Time* 2010: State University of New York Press.
 30. Wiener, J.M. and J. Tilly, *Population ageing in the United States of America: implications for public programmes*. International Journal of Epidemiology, 2002. **31**(4): p. 776-781.
 31. Aziz, O., et al., *Ear-worn body sensor network device: an objective tool for functional postoperative home recovery monitoring*. J Am Med Inform Assoc, 2011. **18**(2): p. 156-9.
 32. Atallah, L., et al., *Observing Recovery from Knee-Replacement Surgery by Using Wearable Sensors*, in *Proceedings of the 2011 International Conference on Body Sensor Networks 2011*, IEEE Computer Society. p. 29-34.
 33. Paffenbarger, R.S., Jr., et al., *The association of changes in physical-activity level and other lifestyle characteristics with mortality among men*. N Engl J Med, 1993. **328**(8): p. 538-45.
 34. Reimers, C.D., G. Knapp, and A.K. Reimers, *Does physical activity increase life expectancy? A review of the literature*. J Aging Res, 2012. **2012**: p. 243958.
 35. Inagawa, T., et al., *Decreased activity of daily living produced by the combination of Alzheimer's disease and lower limb fracture in elderly requiring nursing care*. Environ Health Prev Med, 2012.
 36. Salvatore, P., et al., *Circadian activity rhythm abnormalities in ill and recovered bipolar I disorder patients*. Bipolar Disord, 2008. **10**(2): p. 256-65.
 37. Farr, L.A., C. Campbell-Grossman, and J.M. Mack, *Circadian disruption and surgical recovery*. Nursing research, 1988. **37**(3): p. 170-175.
 38. Feldman, L.S., et al., *Validation of a physical activity questionnaire (CHAMPS) as an indicator of postoperative recovery after laparoscopic cholecystectomy*. Surgery, 2009. **146**(1): p. 31-9.

39. Simonsick, E.M., et al., *Just get out the door! Importance of walking outside the home for maintaining mobility: findings from the women's health and aging study*. J Am Geriatr Soc, 2005. **53**(2): p. 198-203.
40. Boyle, M., et al., *Health care providers' perceived role in changing environments to promote healthy eating and physical activity: baseline findings from health care providers participating in the healthy eating, active communities program*. Pediatrics, 2009. **123 Suppl 5**: p. S293-300.
41. Saxe, J.S., *Promoting healthy lifestyles and decreasing childhood obesity: increasing physician effectiveness through advocacy*. Ann Fam Med, 2011. **9**(6): p. 546-8.
42. Asberg, K.H., *Assessment of ADL in home-care for the elderly. Change in ADL and use of short-term hospital care*. Scand J Soc Med, 1986. **14**(2): p. 105-11.
43. Aggarwal, J.K. and M.S. Ryoo, *Human activity analysis: A review*. ACM Comput. Surv., 2011. **43**(3): p. 1-43.
44. Turaga, P., et al., *Machine Recognition of Human Activities: A Survey*. Circuits and Systems for Video Technology, IEEE Transactions on, 2008. **18**(11): p. 1473-1488.
45. Cheung, V.H., L. Gray, and M. Karunanithi, *Review of Accelerometry for Determining Daily Activity Among Elderly Patients*. Arch Phys Med Rehabil, 2011. **92**(6): p. 998-1014.
46. Poppe, R., *Vision-based human motion analysis: An overview*. Computer Vision and Image Understanding, 2007. **108**(1-2): p. 4-18.
47. Panangadan, A., M. Matarić, and G. Sukhatme, *Tracking and Modeling of Human Activity Using Laser Rangefinders*. International Journal of Social Robotics, 2010. **2**(1): p. 95-107.
48. Nawab, S.H., S.H. Roy, and C.J. De Luca, *Functional activity monitoring from wearable sensor data*. Conf Proc IEEE Eng Med Biol Soc, 2004. **2**: p. 979-82.
49. de Bruin, E.D., et al., *Wearable systems for monitoring mobility-related activities in older people: a systematic review*. Clin Rehabil, 2008. **22**(10-11): p. 878-95.
50. Atallah, L., et al., *Real-time activity classification using ambient and wearable sensors*. IEEE Trans Inf Technol Biomed, 2009. **13**(6): p. 1031-9.
51. Biswas, J., et al., *Health and wellness monitoring through wearable and ambient sensors: exemplars from home-based care of elderly with mild dementia*. Annals of Telecommunications, 2010. **65**(9): p. 505-521.
52. Czarlinska, A., W. Huh, and D. Kundur, *On privacy and security in distributed visual sensor networks*. in Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. 2008.
53. Yang, S. and Q. Li, *Inertial Sensor-Based Methods in Walking Speed Estimation: A Systematic Review*. Sensors, 2012. **12**(5): p. 6102-6116.
54. Jin, W., et al. *A Review of Vision-Based Gait Recognition Methods for Human Identification*. in Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on. 2010.
55. Wong, C., et al., *Enhanced Classification of Abnormal Gait Using BSN and Depth*, in 9th International Conference on Body Sensor Networks 2012: London.
56. Oliver, M., et al., *Utility of accelerometer thresholds for classifying sitting in office workers*. Prev Med, 2010. **51**(5): p. 357-60.
57. Singh, S. and J. Wang, *Human Activity Recognition in Videos: A Systematic Approach* Intelligent Data Engineering and Automated Learning – IDEAL 2006, E. Corchado, et al., Editors. 2006, Springer Berlin / Heidelberg. p. 257-264.
58. Pansiot, J., et al., *Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems* 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), S. Leonhardt, T. Falck, and P. Mähönen, Editors. 2007, Springer Berlin Heidelberg. p. 208-212.

59. Logan, B., et al., *A long-term evaluation of sensing modalities for activity recognition*, in *Proceedings of the 9th international conference on Ubiquitous computing*2007, Springer-Verlag: Innsbruck, Austria. p. 483-500.
60. Atallah, L., et al., *Behaviour Profiling with Ambient and Wearable Sensing*, in BSN, S. Leonhardt, T. Falck, and P. Mähönen, Editors. 2007, Springer. p. 133-138.
61. Skubic, M., et al., *A smart home application to eldercare: Current status and lessons learned*. Technology and Health Care, 2009. **17**(3): p. 183-201.
62. Ravi, N., et al., *Activity recognition from accelerometer data*, in *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3*2005, AAAI Press: Pittsburgh, Pennsylvania. p. 1541-1546.
63. Zhang, S., et al., *Detection of Activities by Wireless Sensors for Daily Life Surveillance: Eating and Drinking*. Sensors, 2009. **9**(3): p. 1499-1517.
64. Liu, J., et al., *An Intelligent Food-intake Monitoring System Using Wearable Sensors*, in *International Conference on Wearable and Implantable Body Sensor Networks*2012: London.
65. Snoek, J., et al., *Automated detection of unusual events on stairs*. Image Vision Comput., 2009. **27**(1-2): p. 153-166.
66. Lee, R.Y.W. and A.J. Carlisle, *Detection of falls using accelerometers and mobile phone technology*. Age and Ageing, 2011.
67. Vishwakarma, V., C. Mandal, and S. Sural, *Automatic detection of human fall in video*, in *Proceedings of the 2nd international conference on Pattern recognition and machine intelligence*2007, Springer-Verlag: Kolkata, India. p. 616-623.
68. Lustrek, M., et al., *Detecting Falls with Location Sensors and Accelerometers*, in IAAI, D.G. Shapiro and M.P.J. Fromherz, Editors. 2011, AAAI.
69. Oliver, N. and E. Horvitz, *S-SEER: selective perception in a multimodal office activity recognition system*, in *Proceedings of the First international conference on Machine Learning for Multimodal Interaction*2005, Springer-Verlag: Martigny, Switzerland. p. 122-135.
70. Modahl, M., et al., *MediaBroker: An Architecture for Pervasive Computing*, in *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*2004, IEEE Computer Society. p. 253.
71. Aggarwal, C.C. and P.S. Yu, *A Survey of Synopsis Construction in Data Streams*, in *Data Streams - Models and Algorithms*, C.C. Aggarwal, Editor 2007, Springer. p. 169-207.
72. Gibbons, P.B. and Y. Matias, *Synopsis data structures for massive data sets*, in *External memory algorithms*, M.A. James and V. Jeffrey Scott, Editors. 1999, American Mathematical Society. p. 39-70.
73. Chris, G., et al., *Chapter 3 Mining Frequent Patterns in Data Streams at Multiple Time*, 2007.
74. Atallah, L., et al., *Gaussian process prediction for cross channel consensus in body sensor networks*. Proceedings of 5th International Summer School and Symposium on Medical Devices and Biosensors, 2008: p. 165-168.
75. Atallah, L., et al., *Validation of an ear-worn sensor for gait monitoring using a force-plate instrumented treadmill*. Gait Posture, 2012. **35**(4): p. 674-6.
76. Pansiot, J., et al. *ClimBSN: Climber performance monitoring with BSN*. in *Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on*. 2008.
77. Pansiot, J., B. Lo, and G.-Z. Yang, *Swimming Stroke Kinematic Analysis with BSN*, in *Proceedings of the 2010 International Conference on Body Sensor Networks*2010, IEEE Computer Society. p. 153-158.

78. King, R.C., et al., *Body Sensor Networks for Monitoring Rowing Technique*, in *Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*2009, IEEE Computer Society. p. 251-255.
79. Reynolds, C. and C.R. Wren. *Worse Is Better for Ambient Sensing*. in *Pervasive 2006: Workshop on Privacy Trust and Identity Issues for Ambient Intelligence*. 2006.
80. Caine, K.E., A.D. Fisk, and W.A. Rogers, *Benefits and Privacy Concerns of a Home Equipped with a Visual Sensing System: A Perspective from Older Adults*. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2006. **50**(2): p. 180-184.
81. Longbo, Z., et al., *Construction of Synopsis for Periodically Updating Sliding Windows over Data Streams*, in *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application - Volume 022008*, IEEE Computer Society. p. 588-592.
82. Guha, S., N. Koudas, and K. Shim, *Approximation and streaming algorithms for histogram construction problems*. ACM Trans. Database Syst., 2006. **31**(1): p. 396-438.
83. Guha, S. and B. Harb, *Wavelet synopsis for data streams: minimizing non-euclidean error*, in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*2005, ACM: Chicago, Illinois, USA. p. 88-97.
84. Lin, J., et al., *Experiencing SAX: a novel symbolic representation of time series*. Data Min. Knowl. Discov., 2007. **15**(2): p. 107-144.
85. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection*. The Journal of Machine Learning Research, 2003. **3**: p. 1157-1182.
86. Vafaie, H. and I. Imam, *Feature Selection Methods: Genetic Algorithms vs. Greedy-like Search*, 1994.
87. Zhang, H. and G. Sun, *Feature selection using tabu search method*. Pattern Recognition, 2002. **35**(3): p. 701-711.
88. Das, S.K., *Feature selection with a linear dependence measure*. Computers, IEEE Transactions on, 1971. **100**(9): p. 1106-1109.
89. Peng, H., F. Long, and C. Ding, *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005. **27**(8): p. 1226-1238.
90. Yu, L. and H. Liu. *Feature selection for high-dimensional data: A fast correlation-based filter solution*. in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. 2003.
91. Gilad-Bachrach, R., A. Navot, and N. Tishby. *Margin based feature selection-theory and algorithms*. in *Proceedings of the twenty-first international conference on Machine learning*. 2004. ACM.
92. Kohavi, R. and G.H. John, *Wrappers for feature subset selection*. Artificial intelligence, 1997. **97**(1): p. 273-324.
93. Sebasti, et al., *Simultaneous feature selection and classification using kernel-penalized support vector machines*. Inf. Sci., 2011. **181**(1): p. 115-128.
94. Neumann, J., et al., *Combined SVM-Based Feature Selection and Classification*. Mach. Learn., 2005. **61**(1-3): p. 129-150.
95. Kojadinovic, I. and T. Wotcka. *Comparision between a filter and a wrapper approach to variable subset selection in regression problems*. Citeseer.
96. Talavera, L., *An evaluation of filter and wrapper methods for feature selection in categorical clustering*, in *Proceedings of the 6th international conference on Advances in Intelligent Data Analysis*2005, Springer-Verlag: Madrid, Spain. p. 440-451.
97. Atallah, L., et al., *Detecting Walking Gait Impairment with an Ear-worn Sensor*, in *Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*2009, IEEE Computer Society. p. 175-180.

98. Gilad-Bachrach, R., A. Navot, and N. Tishby, *Margin based feature selection - theory and algorithms*, in *Proceedings of the twenty-first international conference on Machine learning*2004, ACM: Banff, Alberta, Canada. p. 43.
99. Atallah, L., et al., *Sensor Placement for Activity Detection Using Wearable Accelerometers*, in *Proceedings of the 2010 International Conference on Body Sensor Networks*2010, IEEE Computer Society. p. 24-29.
100. Kira, K. and L.A. Rendell, *A practical approach to feature selection*, in *Proceedings of the ninth international workshop on Machine learning*1992, Morgan Kaufmann Publishers Inc.: Aberdeen, Scotland, United Kingdom. p. 249-256.
101. Yang, X., K. Tang, and X. Yao, *The minimum redundancy-maximum relevance approach to building sparse support vector machines*, in *Proceedings of the 10th international conference on Intelligent data engineering and automated learning*2009, Springer-Verlag: Burgos, Spain. p. 184-190.
102. Jolliffe, I.T., *Principal Component Analysis*2002: Springer-Verlag.
103. Borg, I. and P.J.F. Groenen, *Modern Multidimensional Scaling: Theory And Applications*2005: Springer.
104. Lee, J.A. and M. Verleysen, *Nonlinear Dimensionality Reduction*2007: Springer.
105. Demiris, G. and B.K. Hensel, *Technologies for an aging society: a systematic review of "smart home" applications*. Yearb Med Inform, 2008: p. 33-40.
106. Cook, D.J., et al., *MavHome: An Agent-Based Smart Home*, in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*2003, IEEE Computer Society. p. 521.
107. Sohn, T., et al., *Experiences with place lab: an open source toolkit for location-aware computing*, in *Proceedings of the 28th international conference on Software engineering*2006, ACM: Shanghai, China. p. 462-471.
108. Helal, S. and C. Chen, *The Gator Tech Smart House: enabling technologies and lessons learned*, in *Proceedings of the 3rd International Convention on Rehabilitation Engineering \& Assistive Technology*2009, ACM: Singapore. p. 1-4.
109. L, S., et al., *Recognition of emergent human behaviour in a smart home: A data mining approach*. Pervasive Mob. Comput., 2007. **3**(2): p. 95-116.
110. Barnes, N.M., et al., *Lifestyle monitoring-technology for supported independence*. Computing & Control Engineering Journal, 1998. **9**(4): p. 169-174.
111. Park, K., et al., *Human behavioral detection and data cleaning in assisted living environment using wireless sensor networks*, in *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments*2009, ACM: Corfu, Greece. p. 1-8.
112. Purpura, S., et al., *Fit4life: the design of a persuasive technology promoting healthy behavior and ideal weight*, in *Proceedings of the 2011 annual conference on Human factors in computing systems*2011, ACM: Vancouver, BC, Canada. p. 423-432.
113. Yang, C.C. and Y.L. Hsu, *A review of accelerometry-based wearable motion detectors for physical activity monitoring*. Sensors (Basel, Switzerland), 2010. **10**(8): p. 7772-88.
114. Henpraserttae, A., S. Thiemjarus, and S. Marukatat, *Accurate Activity Recognition Using a Mobile Phone Regardless of Device Orientation and Location*, in *Proceedings of the 2011 International Conference on Body Sensor Networks*2011, IEEE Computer Society. p. 41-46.
115. Lo, B., et al., *Real-Time Pervasive Monitoring for Postoperative Care* *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, S. Leonhardt, T. Falck, and P. Mähönen, Editors. 2007, Springer Berlin Heidelberg. p. 122-127.
116. Lin, Z., et al., *Precise process definitions for activities of daily living: a basis for real-time monitoring and hazard detection*, in *Proceedings of the 3rd Workshop on Software Engineering in Health Care*2011, ACM: Waikiki, Honolulu, HI, USA. p. 13-16.

117. Aziz, O., B. Lo, and G.-Z. Yang, *Pervasive healthcare: clinical drive, technological innovations, and socio-economic benefits*. IEE Seminar Digests, 2005. **2005**(1): p. 127-132.
118. Vanhecke, T.E., et al., *Cardiorespiratory fitness and sedentary lifestyle in the morbidly obese*. Clin Cardiol, 2009. **32**(3): p. 121-4.
119. Ali, R., et al., *Detection and analysis of transitional activity in manifold space*. IEEE Trans Inf Technol Biomed, 2012. **16**(1): p. 119-28.
120. Kriska, A.M., et al., *Physical Activity, Obesity, and the Incidence of Type 2 Diabetes in a High-Risk Population*. American Journal of Epidemiology, 2003. **158**(7): p. 669-675.
121. Oliver, N., E. Horvitz, and A. Garg, *Layered Representations for Human Activity Recognition*, in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces2002*, IEEE Computer Society. p. 3.
122. Brand, M., N. Oliver, and A. Pentland, *Coupled hidden Markov models for complex action recognition*, in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)1997*, IEEE Computer Society. p. 994.
123. Dawkins, R., *The Selfish Gene*1989: Oxford University Press.
124. Bui, H.H., S. Venkatesh, and G. West, *Policy recognition in the abstract hidden Markov model*. J. Artif. Int. Res., 2002. **17**(1): p. 451-499.
125. Talukder, A., et al., *Dynamic Control and Power Management Algorithm For Continuous Wireless Monitoring in Sensor Networks*, in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks2004*, IEEE Computer Society. p. 498-505.
126. Talukder, A., et al. *Optimal sensor scheduling and power management in sensor networks*. in *Optical Pattern Recognition, Proceedings of the SPIE*. 2005.
127. Yoshida, T., et al., *Development of a wearable surveillance system using gait analysis*. Telemed J E Health, 2007. **13**(6): p. 703-13.
128. Dawson, D.A.P.D., *Aging in the Eighties : Functional Limitations of Individuals Age 65 Years and Over*.
129. Maki, B.E., P.J. Holliday, and A.K. Topper, *Fear of falling and postural performance in the elderly*. J Gerontol, 1991. **46**(4): p. M123-31.
130. Topper, A.K., B.E. Maki, and P.J. Holliday, *Are activity-based assessments of balance and gait in the elderly predictive of risk of falling and/or type of fall?* J Am Geriatr Soc, 1993. **41**(5): p. 479-87.
131. Janssen, W., et al., *Recovery of the sit-to-stand movement after stroke: a longitudinal cohort study*. Neurorehabil Neural Repair, 2010. **24**(8): p. 763-9.
132. Frykberg, G.E., et al., *Temporal coordination of the sit-to-walk task in subjects with stroke and in controls*. Arch Phys Med Rehabil, 2009. **90**(6): p. 1009-17.
133. Nitz, J.C., Y.R. Burns, and R.V. Jackson, *Sit-to-stand and Walking Ability in Patients with Neuromuscular Conditions*. Physiotherapy, 1997. **83**(5): p. 223-227.
134. Manckoundia, P., et al., *Comparison of motor strategies in sit-to-stand and back-to-sit motions between healthy and Alzheimer's disease elderly subjects*. Neuroscience, 2006. **137**(2): p. 385-92.
135. Gioftsos, G. and D.W. Grieve, *The use of artificial neural networks to identify patients with chronic low-back pain conditions from patterns of sit-to-stand manoeuvres*. Clin Biomech (Bristol, Avon), 1996. **11**(5): p. 275-280.
136. Allin, S. and A. Mihailidis, *Automated sit-to-stand detection and analysis*, in *Symposium on AI in Eldercare: New Solutions to Old Problems*, AAAI, Editor 2008: Arlington.
137. Najafi, B., et al., *Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly*. IEEE Trans Biomed Eng, 2002. **49**(8): p. 843-51.

138. Tinetti, M.E., M. Speechley, and S.F. Ginter, *Risk factors for falls among elderly persons living in the community*. N Engl J Med, 1988. **319**(26): p. 1701-7.
139. Kang, D.W., et al., *Real-time elderly activity monitoring system based on a tri-axial accelerometer*. Disabil Rehabil Assist Technol, 2010. **5**(4): p. 247-53.
140. Zhu, C. and W. Sheng, *Human daily activity recognition in robot-assisted living using multi-sensor fusion*, in *Proceedings of the 2009 IEEE international conference on Robotics and Automation* 2009, IEEE Press: Kobe, Japan. p. 3644-3649.
141. Schibler, U., *The daily rhythms of genes, cells and organs*. EMBO Rep, 2005. **6**(S1): p. S9-S13.
142. Martin, P.R. and P.P.G. Bateson, *Measuring Behaviour: An Introductory Guide* 1986: Cambridge University Press.
143. Glodek, M., et al., *Incorporating uncertainty in a layered HMM architecture for human activity recognition*, in *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding* 2011, ACM: Scottsdale, Arizona, USA. p. 33-34.
144. Chen, C., et al., *Factorial HMM and parallel HMM for gait recognition*. Trans. Sys. Man Cyber Part C, 2009. **39**(1): p. 114-123.
145. \, M., et al., *Applying Space State Models in Human Action Recognition: A Comparative Study*, in *Proceedings of the 5th international conference on Articulated Motion and Deformable Objects* 2008, Springer-Verlag: Port d'Andratx, Mallorca, Spain. p. 53-62.
146. Wren, C.R., D.C. Minnen, and S.G. Rao, *Similarity-based analysis for large networks of ultra-low resolution sensors*. Pattern Recogn., 2006. **39**(10): p. 1918-1931.
147. Friedman, N., K. Murphy, and S. Russell, *Learning the structure of dynamic probabilistic networks*, in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* 1998, Morgan Kaufmann Publishers Inc.: Madison, Wisconsin. p. 139-147.
148. Youtian, D. *Recognizing Interaction Activities using Dynamic Bayesian Network*. 2006.
149. Muncaster, J. and Y. Ma, *Hierarchical Model-Based Activity Recognition With Automatic Low-Level State Discovery*. 2007. Vol. 2. 2007.
150. Liao, L., et al., *Learning and inferring transportation routines*. Artif. Intell., 2007. **171**(5-6): p. 311-331.
151. Han, J., et al., *Frequent pattern mining: current status and future directions*. Data Min. Knowl. Discov., 2007. **15**(1): p. 55-86.
152. Ohsaki, M., et al., *Evaluation of rule interestingness measures in medical knowledge discovery in databases*. Artif. Intell. Med., 2007. **41**(3): p. 177-196.
153. Jaroszewicz, S., *Using interesting sequences to interactively build Hidden Markov Models*. Data Min. Knowl. Discov., 2010. **21**(1): p. 186-220.
154. Eagle, N. and A. Pentland, *Eigenbehaviors: identifying structure in routine*. Behavioral Ecology and Sociobiology, 2009. **63**(7): p. 1057-1066.
155. Virone, G., N. Noury, and J. Demongeot, *A system for automatic measurement of circadian activity deviations in telemedicine*. IEEE Trans Biomed Eng, 2002. **49**(12): p. 1463-9.
156. Barger, T.S., D.E. Brown, and M. Alwan, *Health-status monitoring through analysis of behavioral patterns*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2005. **35**(1): p. 22-27.
157. Berndt, D. and J. Clifford. *Using Dynamic Time Warping to Find Patterns in Time Series*. in *KDD Workshop*. 1994.
158. Ouivirach, K. and M.N. Dailey. *Clustering human behaviors with dynamic time warping and hidden Markov models for a video surveillance system*. in *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*. 2010.

159. Pham, C., et al., *A dynamic time warping approach to real-time activity recognition for food preparation*, in *Proceedings of the First international joint conference on Ambient intelligence*2010, Springer-Verlag: Malaga, Spain. p. 21-30.
160. YuanYuan, L. and L.E. Parker. *Detecting and monitoring time-related abnormal events using a wireless sensor network and mobile robot*. in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. 2008.
161. Mori, T., et al. *Anomaly detection algorithm based on life pattern extraction from accumulated pyroelectric sensor data*. in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. 2008.
162. Markou, M. and S. Singh, *Novelty detection: a review—part 1: statistical approaches*. *Signal Process.*, 2003. **83**(12): p. 2481-2497.
163. Tarassenko, L., et al. *Novelty detection for the identification of masses in mammograms*. in *Artificial Neural Networks, 1995., Fourth International Conference on*. 1995.
164. Viswanath, P., M.N. Murty, and S. Kambala, *An efficient parzen-window based network intrusion detector using a pattern synthesis technique*, in *Proceedings of the First international conference on Pattern Recognition and Machine Intelligence*2005, Springer-Verlag: Kolkata, India. p. 799-804.
165. King, R.C., et al., *Development of a wireless sensor glove for surgical skills assessment*. *IEEE Trans Inf Technol Biomed*, 2009. **13**(5): p. 673-9.
166. Pradhan, G.N. and B. Prabhakaran, *Clustering of human motions based on feature-level fusion of multiple body sensor data*, in *Proceedings of the 1st ACM International Health Informatics Symposium*2010, ACM: Arlington, Virginia, USA. p. 66-75.
167. McIlwraith, D.G., et al. *Probabilistic decision level fusion for real-time correlation of ambient and wearable sensors*. in *Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on*. 2008.
168. Ashit, T., et al., *Predictive Controller for Heterogeneous Sensor Network Operation in Dynamic Environments*, 2008.
169. Zhang, M. and A.A. Sawchuk, *Motion primitive-based human activity recognition using a bag-of-features approach*, in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*2012, ACM: Miami, Florida, USA. p. 631-640.
170. Begg, R.K., M. Palaniswami, and B. Owen, *Support vector machines for automated gait classification*. *IEEE Trans Biomed Eng*, 2005. **52**(5): p. 828-38.
171. He, Z. *Activity recognition from accelerometer signals based on Wavelet-AR model*. in *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*. 2010. IEEE.
172. Martin, E. *Solving training issues in the application of the wavelet transform to precisely analyze human body acceleration signals*. in *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*. 2010.
173. Ha, S., Y. Han, and H. Hahn, *Adaptive gait pattern generation of biped robot based on human's gait pattern analysis*. *International Journal of Mechanical Systems Science and Engineering*, 2007. **1**(2): p. 80–85.
174. Alpaydin, E., *Introduction To Machine Learning*2004: Mit Press.
175. Dreyfus, G., *Neural Networks: Methodology And Applications*2005: Springer.
176. Salzberg, S.L., *C4.5: Programs for Machine Learning* by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 1994. **16**(3): p. 235-240.
177. Lowd, D. and P. Domingos, *Naive Bayes models for probability estimation*, in *Proceedings of the 22nd international conference on Machine learning*2005, ACM: Bonn, Germany. p. 529-536.
178. Hand, D.J. and K. Yu, *Idiot's Bayes—Not So Stupid After All?* *International Statistical Review*, 2001. **69**(3): p. 385-398.

179. Dawson, D., G. Hendershot, and J. Fulton, *Aging in the eighties: The prevalence of comorbidity and its association with disability*, in *Advance Data From Vital and Health Statistics* 1987, National Center for Health Statistics: Hyattsville, MD. p. 87–1250.
180. Britton, E., N. Harris, and A. Turton, *An exploratory randomized controlled trial of assisted practice for improving sit-to-stand in stroke patients in the hospital setting*. Clin Rehabil, 2008. **22**(5): p. 458-468.
181. Dall, P.M. and A. Kerr, *Frequency of the sit to stand task: An observational study of free-living adults*. Applied Ergonomics, 2010. **41**(1): p. 58-61.
182. Cheng, P.T., et al., *The sit-to-stand movement in stroke patients and its correlation with falling*. Arch Phys Med Rehabil, 1998. **79**(9): p. 1043-6.
183. Janssen, W.G.M. *The Sit-to-Stand Movement recovery after stroke and objective assessment*. Van zitten naar staan het herstel na een beroerte en het objectief meten van de beweging 2008; Available from: <http://hdl.handle.net/1765/13555>.
184. Frykberg, G.E., et al., *Temporal Coordination of the Sit-to-Walk Task in Subjects With Stroke and in Controls*. Arch Phys Med Rehabil, 2009. **90**(6): p. 1009-1017.
185. Galli, M., et al., *Sit-to-stand movement analysis in obese subjects*. Int J Obes Relat Metab Disord, 2000. **24**(11): p. 1488-92.
186. Allin, S. and A. Mihailidis, *Low-cost, Automated Assessment of Sit-To-Stand Movement in "Natural" Environments* 4th European Conference of the International Federation for Medical and Biological Engineering, J. Sloten, et al., Editors. 2009, Springer Berlin Heidelberg. p. 76-79.
187. Yang, A.Y., et al. *Distributed segmentation and classification of human actions using a wearable motion sensor network*. in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*. 2008.
188. Tenenbaum, J.B., V.d. Silva, and J.C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*. Science, 2000. **290**(5500): p. 2319-2323.
189. Bernd, K. and M.O. Hinke, *The Lorenz manifold as a collection of geodesic level sets*. Nonlinearity, 2004. **17**(1): p. C1.
190. Chung, F.R.K., *Spectral Graph Theory* 1997: Conference Board of the Mathematical Sciences.
191. Fiedler, M., *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*. Czechoslovak Mathematical Journal, 1975. **25**.
192. Chan, T.F., J. P. Ciarlet, and W.K. Szeto, *On the Optimality of the Median Cut Spectral Bisection Graph Partitioning Method*. SIAM Journal on Scientific Computing, 1997. **18**(3): p. 943-948.
193. Ali, R., et al., *Transitional Activity Recognition with Manifold Embedding*, in *Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks* 2009, IEEE Computer Society. p. 98-102.
194. Belkin, M., *Problems of Learning on Manifolds*, 2003, The University of Chicago.
195. Bezdek, J.C., R. Ehrlich, and W. Full, *FCM: The fuzzy c-means clustering algorithm*. Computers & Geosciences, 1984. **10**(2–3): p. 191-203.
196. Roweis, S.T. and L.K. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*. Science, 2000. **290**(5500): p. 2323-2326.
197. Belkin, M. and P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*. Neural Computation, 2003. **15**(6): p. 1373-1396.
198. Vin De, S. and B.T. Joshua, *Global Versus Local Methods in Nonlinear Dimensionality Reduction*, 2003, MIT Press.
199. Kevin, R.K., et al., *Excitation/emission resolved fluorescence imaging with*, 2010.
200. Specht, D.F., *A general regression neural network*. Neural Networks, IEEE Transactions on, 1991. **2**(6): p. 568-576.
201. Mark, J.L.O. and S. Centre For Cognitive, *Introduction to radial basis function networks*, 1996.

202. Lekadir, K., et al., *Tissue characterization using dimensionality reduction and fluorescence imaging*, in *Proceedings of the 9th international conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part II* 2006, Springer-Verlag: Copenhagen, Denmark. p. 586-593.
203. Mokkink, L.B., et al., *Construct validity of the DynaPort®KneeTest: a comparison with observations of physical therapists*. Osteoarthritis and cartilage / OARS, Osteoarthritis Research Society, 2005. **13**(8): p. 738-743.
204. Mokkink, L.B., et al., *Reproducibility and validity of the DynaPort KneeTest*. Arthritis Care & Research, 2005. **53**(3): p. 357-363.
205. Huynh, T. and B. Schiele. *Analyzing features for activity recognition*. in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. 2005. ACM.
206. Baños, O., H. Pomares, and I. Rojas, *Novel Method for Feature-Set Ranking Applied to Physical Activity Recognition*, in *Trends in Applied Intelligent Systems*, N. García-Pedrajas, et al., Editors. 2010, Springer Berlin Heidelberg. p. 637-642.
207. Mathie, M., *Monitoring and interpreting human movement patterns using a triaxial accelerometer*, 2003, The University of New South Wales.
208. Bonomi, A.G., et al., *Detection of type, duration, and intensity of physical activity using an accelerometer*. Med Sci Sports Exerc, 2009. **41**(9): p. 1770-1777.
209. Janssen, W.G., H.B. Bussmann, and H.J. Stam, *Determinants of the Sit-to-Stand Movement: A Review*. Physical Therapy, 2002. **82**(9): p. 866-879.
210. Demura, S. and T. Yamada, *Height of chair seat and movement characteristics in sit-to-stand by young and elderly adults*. Percept Mot Skills, 2007. **104**(1): p. 21-31.
211. Ali, R., et al. *Pattern mining for routine behaviour discovery in pervasive healthcare environments*. in *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*. 2008.
212. Nagel, T., *What Is It Like to Be a Bat?* The Philosophical Review, 1974. **83**(4): p. 435-450.
213. Agrawal, R. and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, in *Proceedings of the 20th International Conference on Very Large Data Bases* 1994, Morgan Kaufmann Publishers Inc. p. 487-499.
214. Han, J., J. Pei, and Y. Yin, *Mining frequent patterns without candidate generation*. SIGMOD Rec., 2000. **29**(2): p. 1-12.
215. Wang, J., J. Han, and J. Pei, *CLOSET+: searching for the best strategies for mining frequent closed itemsets*, in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* 2003, ACM: Washington, D.C. p. 236-245.
216. Jian, P., et al. *PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth*. in *Data Engineering, 2001. Proceedings. 17th International Conference on*. 2001.
217. O'Hear, S., *Endomondo's Sports Tracker gathers pace – reaches 500,000 users*, in *techcrunch.com* 2010: <http://techcrunch.com/2010/10/21/endomondos-sports-tracker-gathers-pace-reaches-500000-users/>.
218. Apiwat, H. *Accurate Activity Recognition Using a Mobile Phone Regardless of Device Orientation and Location*. 2011.
219. Kwapisz, J.R., G.M. Weiss, and S.A. Moore, *Activity recognition using cell phone accelerometers*. SIGKDD Explor. Newsl., 2011. **12**(2): p. 74-82.
220. Bieber, G., J. Voskamp, and B. Urban. *Activity recognition for everyday life on mobile phones*. 2009.
221. Nanami, R., Y. Kawahawa, and T. Asami. *A Calorie Count Application for a Mobile Phone Based on METS Value*. in *Sensor, Mesh and Ad Hoc Communications and*

- Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on. 2008.
222. Sun, L., et al., *Activity Recognition on an Accelerometer Embedded Mobile Phone with Varying Positions and Orientations* *Ubiquitous Intelligence and Computing*, Z. Yu, et al., Editors. 2010, Springer Berlin / Heidelberg. p. 548-562.
 223. Willett, W.C., et al., *Prevention of Chronic Disease by Means of Diet and Lifestyle Changes.*, in *Disease Control Priorities in Developing Countries* 2006, Oxford University Press: New York.
 224. Brundtland, G.H., *Reducing Risks to Health, Promoting Healthy Life*. JAMA: The Journal of the American Medical Association, 2002. **288**(16): p. 1974.
 225. Vainio, H. and F. Bianchini, *Weight Control and Physical Activity* 2002: IARC Press.
 226. Myers, J., *Exercise and Cardiovascular Health*. Circulation, 2003. **107**(1): p. e2-e5.
 227. Sigal, R.J., et al., *Physical Activity/Exercise and Type 2 Diabetes*. Diabetes Care, 2004. **27**(10): p. 2518-2539.
 228. Shephard, R.J., *Physical Activity and Cancer*. Int J Sports Med, 1990. **11**(06): p. 413,420.
 229. Marcus, B.H., et al., *Physical Activity Intervention Studies*. Circulation, 2006. **114**(24): p. 2739-2752.
 230. Tibshirani, R., G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2001. **63**(2): p. 411-423.
 231. Jain, A.K., M.N. Murty, and P.J. Flynn, *Data clustering: a review*. ACM Comput. Surv., 1999. **31**(3): p. 264-323.
 232. Roberts, S.J., et al., *Bayesian approaches to Gaussian mixture modeling*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. **20**(11): p. 1133-1142.
 233. Moon, T.K., *The expectation-maximization algorithm*. Signal Processing Magazine, IEEE, 1996. **13**(6): p. 47-60.
 234. Ferrari, L. and M. Mamei. *Discovering daily routines from Google Latitude with topic models*. in *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2011 IEEE International Conference on. 2011.
 235. Liao, L., D. Fox, and H. Kautz, *Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields*. Int. J. Rob. Res., 2007. **26**(1): p. 119-134.
 236. Ashbrook, D. and T. Starner, *Using GPS to learn significant locations and predict movement across multiple users*. Personal Ubiquitous Comput., 2003. **7**(5): p. 275-286.
 237. Khetarpaul, S., et al., *Mining GPS data to determine interesting locations*, in *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011* 2011, ACM: Hyderabad, India. p. 1-6.
 238. Arisaka, Y., *Spatiality, temporality, and the problem of foundation in being and time*.
 239. De Silva, V. and J.B. Tenenbaum, *Sparse multidimensional scaling using landmark points*. 2004.
 240. Hubert, L. and P. Arabie, *Comparing partitions*. Journal of Classification, 1985. **2**(1): p. 193-218.
 241. Tan, P.-N., M. Steinbach, and V. Kumar, *Introduction to Data Mining* 2005: Addison Wesley.
 242. Vallieres, A., et al., *Variability and predictability in sleep patterns of chronic insomniacs*. J Sleep Res, 2005. **14**(4): p. 447-53.
 243. Mezick, E.J., et al., *Intra-individual variability in sleep duration and fragmentation: Associations with stress*. Psychoneuroendocrinology, 2009. **34**(9): p. 1346-1354.

244. Zhou, F. and F. la Torre, *Canonical Time Warping for Alignment of Human Behavior*, in *Advances in Neural Information Processing Systems 22*, Y. Bengio, et al., Editors. 2009. p. 2286-2294.
245. Hadidi, T. and N. Noury, *A Predictive Analysis of the Night-Day Activities Level of Older Patient in a Health Smart Home*, in *Proceedings of the 7th International Conference on Smart Homes and Health Telematics: Ambient Assistive Health and Wellness Management in the Heart of the City 2009*, Springer-Verlag: Tours, France. p. 290-293.
246. Li, T., M. Ogihara, and S. Zhu, *Association-based similarity testing and its applications*. *Intell. Data Anal.*, 2003. 7(3): p. 209-232.
247. Strehl, A. and J. Ghosh, *Cluster ensembles --- a knowledge reuse framework for combining multiple partitions*. *J. Mach. Learn. Res.*, 2003. 3: p. 583-617.
248. Kapur, A. and C.R. Bhat, *On Modeling Adults' Weekend Day Time Use by Activity Purpose and Accompaniment Arrangement*. *Journal of the Transportation Research Board*, 2007(2021).
249. Larsen, P., *Chronic Illness: Impact and Intervention* 2011: Jones & Bartlett Learning.
250. Zisberg, A., N. Gur-Yaish, and T. Shochat, *Contribution of routine to sleep quality in community elderly*. *Sleep*, 2010. 33(4): p. 509-14.
251. Prigerson, H.G., et al., *Lifestyle regularity and activity level as protective factors against bereavement-related depression in late-life*. *Depression*, 1995. 3(6): p. 297-302.
252. James, B.D., et al., *Late-life social activity and cognitive decline in old age*. *J Int Neuropsychol Soc*, 2011. 17(6): p. 998-1005.
253. Sherwood, N.E. and R.W. Jeffery, *The behavioral determinants of exercise: implications for physical activity interventions*. *Annu Rev Nutr*, 2000. 20: p. 21-44.
254. Roos, E.M. and S. Toksvig-Larsen, *Knee injury and Osteoarthritis Outcome Score (KOOS) - validation and comparison to the WOMAC in total knee replacement*. *Health Qual Life Outcomes*, 2003. 1: p. 17.
255. Dean, J. and S. Ghemawat, *MapReduce: simplified data processing on large clusters*, in *Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6* 2004, USENIX Association: San Francisco, CA. p. 10-10.
256. White, T., *Hadoop: The Definitive Guide* 2009: O'Reilly Media, Inc. 528.
257. *Sexual Activity Tracked By Fitbit Shows Up In Google Search Results*. 2011; Available from: <http://techcrunch.com/2011/07/03/sexual-activity-tracked-by-fitbit-shows-up-in-google-search-results/>.
258. Bellman, R.E., *Dynamic Programming* 2003: Dover.
259. Hughes, G., *On the mean accuracy of statistical pattern recognizers*. *Information Theory, IEEE Transactions on*, 1968. 14(1): p. 55-63.
260. Jain, A.K. and B. Chandrasekaran, *39 Dimensionality and sample size considerations in pattern recognition practice*, in *Handbook of Statistics*, P.R. Krishnaiah and L.N. Kanal, Editors. 1982, Elsevier. p. 835-855.
261. Vapnik, V.N. and A.Y. Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*. *Theory of Probability & Its Applications*, 1971. 16(2): p. 264-280.
262. Engel, A. and W. Fink, *Statistical mechanics calculation of Vapnik-Chervonenkis bounds for perceptrons*. *Journal of Physics A: Mathematical and General*, 1999. 26(23): p. 6893.
263. Lee, J.M., *Introduction to topological manifolds*. Vol. 202. 2010: Springer.
264. Balasubramanian, M. and E.L. Schwartz, *The isomap algorithm and topological stability*. *Science*, 2002. 295(5552): p. 7-7.
265. Zhang, Z. and H. Zha, *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*. *Journal of Shanghai University (English Edition)*, 2004. 8(4): p. 406-424.

266. Donoho, D.L. and C. Grimes, *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*. Proceedings of the National Academy of Sciences, 2003. **100**(10): p. 5591-5596.
267. Weinberger, K.Q., F. Sha, and L.K. Saul, *Learning a kernel matrix for nonlinear dimensionality reduction*, in *Proceedings of the twenty-first international conference on Machine learning* 2004, ACM: Banff, Alberta, Canada. p. 106.
268. Huo, X., X. Ni, and A.K. Smith, *A survey of manifold-based learning methods*.
269. Venna, J. and S. Kaski, *Local multidimensional scaling*. Neural Networks, 2006. **19**(6): p. 889-899.
270. Ahlfors, L.V., *Conformal Invariants: Topics in Geometric Function Theory* 1973: AMS Chelsea Pub.
271. Weinberger, K.Q. and L.K. Saul. *Unsupervised learning of image manifolds by semidefinite programming*. in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. IEEE.
272. Cukierski, W.J. and D.J. Foran, *Using Betweenness Centrality to Identify Manifold Shortcuts*. Proc IEEE Int Conf Data Min, 2008. **2008**: p. 949-958.
273. Jing, L. and C. Shao, *Selection of the Suitable Parameter Value for ISOMAP*. 2011. Vol. 6. 2011.
274. Feng, L., et al. *A neighborhood selection algorithm for manifold learning*. in *Computer Design and Applications (ICCD), 2010 International Conference on*. 2010. IEEE.
275. Cayton, L., *Algorithms for manifold learning*. 2005.
276. Pless, R. and R. Souvenir, *A Survey of Manifold Learning for Images*. IPSJ Transactions on Computer Vision and Applications, 2009. **1**: p. 83-94.
277. Kim, S., et al., *Beyond Sentiment: The Manifold of Human Emotions*. arXiv preprint arXiv:1202.1568, 2012.
278. Patwari, N. and A.O. Hero III. *Manifold learning algorithms for localization in wireless sensor networks*. in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. 2004. IEEE.
279. Cooley, J.W. and J.W. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series*. Mathematics of Computation, 1965. **19**(90): p. 297-301.
280. Akansu, A.N., W.A. Serdijn, and I.W. Selesnick, *Full length article: Emerging applications of wavelets: A review*. Phys. Commun., 2010. **3**(1): p. 1-18.
281. Burrus, C.S., R.A. Gopinath, and H. Guo, *Introduction to wavelets and wavelet transforms: a primer*. Vol. 23. 1998: Prentice hall Upper Saddle River NJ.