Deakin University
School of Information Technology
Centre for Pattern Recognition and Data Analytics

Making Sense of Pervasive Signals: a Machine Learning Approach

by

Thinh-Binh Nguyen
(Nguyễn Thanh Bình)
M.Sc.

Submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy
of
Deakin University

February 2018
I am the author of the thesis entitled

**Making Sense of Pervasive Signals: a Machine Learning Approach**

submitted for the degree of **Doctor of Philosophy**

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

*I certify that I am the student named below and that the information provided in the form is correct*

**Full Name:** Thanh-Binh Nguyen  
(Please Print)

**Signed:** Signature Redacted by Library

**Date:** 28/5/2018
DEAKIN UNIVERSITY
CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Making Sense of Pervasive Signals: a Machine Learning Approach

submitted for the degree of Doctor of Philosophy

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: Thanh-Binh Nguyen
(Please Print)

Signed: [Signature Redacted by Library]

Date: 28/5/2018
# Contents

Abstract xii

Acknowledgements xv

Relevant Publications xvii

Abbreviations xix

1 Introduction 1
   1.1 Motivations ................................................. 1
   1.2 Aims and Approaches ....................................... 4
   1.3 Contributions and Significance .............................. 7
   1.4 Structure of the Thesis ..................................... 8

2 Related Background 10
   2.1 Introduction to Ubiquitous and Pervasive Computing .... 10
      2.1.1 Pervasive and Ubiquitous Computing .................... 11
      2.1.2 Context Acquisition and Understanding Human Dynamics . 13
         2.1.2.1 Context and Context Acquisition ................... 13
         2.1.2.2 Understanding Human Dynamics ..................... 13
   2.2 Data Modelling and Bayesian Approach ........................ 19
      2.2.1 Mixture Modelling via Topic Models .................... 19
         2.2.1.1 Finite Mixture Model ............................... 19
         2.2.1.2 Bayesian Mixture Model ............................ 21
         2.2.1.3 Dirichlet Processes ................................. 23
         2.2.1.4 Dirichlet Process Mixture .......................... 25
3 Arrhythmia Detection from Wearable ECG Devices

3.1 Motivation

3.2 Electrocardiography and Related Background

3.2.1 Electrocardiography Signals

3.2.2 Arrhythmia

3.2.3 ECG Signatures

3.2.4 Wearable ECG Platforms

3.3 Datasets

3.3.1 MIT-BIH Arrhythmia Dataset

3.3.2 Shimmer ECG Dataset

3.4 Detection Methods and Experiment Results

3.4.1 Arrhythmia in ECG and Clinical Rule-based Detection

3.4.2 Heartbeat Detection and Segmentation

3.4.3 Feature Sets

3.4.4 Comparison of Feature Sets

3.4.5 Normalisation

3.4.6 Comparison of Normalisation-based and Rule-based Methods

3.5 Concluding Remarks

4 Daily Routine Discovery from WiFi Data for Understanding Human Dynamics

4.1 Motivation

4.2 Approach

4.3 Experiments

4.3.1 The Nokia MDC Dataset

4.3.2 Significant Location Discovery from WiFi Data

4.3.3 Daily Routine from Location Trajectory

4.3.4 Understanding Human Dynamics from Locations

4.4 Concluding Remarks
5 Multi-Channel Nonparametric Clustering Model for Pervasive Data 69
5.1 Motivation .................................. 70
5.2 MCNC Model .................................. 72
  5.2.1 Multi-Channel Nonparametric Clustering (MCNC) ............. 72
  5.2.2 Marginalisation Property .............................. 74
  5.2.3 Handling Missing Data ................................. 76
5.3 Experiments .................................. 77
  5.3.1 Experiments with Synthetic Data ......................... 77
  5.3.2 The StudentLife Dataset ............................... 78
  5.3.3 Discovering the Who-When-Where Co-patterns with Complete Data ........................................ 81
  5.3.4 Evaluation of Performance .............................. 82
  5.3.5 Discovering the Who-When-Where Co-patterns with Missing Data ........................................ 85
5.4 Concluding Remarks ......................... 86

6 Learning Hierarchical Representation in Noisy, Heterogeneous and Multi-Channel Data 87
6.1 Motivation .................................. 88
6.2 Additional Related Background ...................... 90
  6.2.1 Pattern Discovery from Heterogeneous Data ............... 90
  6.2.2 Discovery of Interaction and Mobility Patterns from Bluetooth and WiFi Data ........................................ 92
6.3 Proposed Models .................................. 92
  6.3.1 Product-space Hierarchical Dirichlet Processes ............ 93
    6.3.1.1 Stochastic Representation ............................ 93
    6.3.1.2 Stick-breaking Representation ........................ 93
    6.3.1.3 Posterior Inference ................................. 94
  6.3.2 Marginalisation Property .............................. 95
6.4 Experiments .................................. 97
  6.4.1 Experiments with Synthetic Data ......................... 97
  6.4.2 Discovery of Who-When-Where Patterns with Complete Data ........................................ 99
  6.4.3 Multilevel Pattern Analysis ............................. 102
    6.4.3.1 Global Pattern Analysis ............................. 103
    6.4.3.2 Local Pattern Analysis ............................. 104
  6.4.4 Evaluation of Performance ............................. 104
List of Figures

2.1 Graphical representation of the finite mixture model. . . . . . . . . . 20
2.2 Graphical representation of the Bayesian mixture model. . . . . . . . 22
2.3 Graphical representation of the DPM model. . . . . . . . . . . . . . 26
2.4 Graphical representation of the HDP model. . . . . . . . . . . . . . 28

3.1 Diagram of the heart’s electrical system. . . . . . . . . . . . . . . . . 37
3.2 ECG tracing with cardiac electrical activity. . . . . . . . . . . . . . . 38
3.3 An excerpt of ECG signal showing major clinical features in a normal heartbeat. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
3.4 VitalJacket for ECG recordings. . . . . . . . . . . . . . . . . . . . . . 40
3.5 A Shimmer ECG platform. . . . . . . . . . . . . . . . . . . . . . . . . 41
3.6 The electrode positions of the Shimmer ECG device. . . . . . . . . . 42
3.7 Excerpt from recording 100 of the MIT-BIH Arrhythmia dataset. . . 43

4.1 Example of adjacent WiFi access points and user’s location. . . . . . 56
4.2 Factor graph for message passing in the AP algorithm. . . . . . . . . 57
4.3 Number of scans corresponding to the ground truth places of six users which have the largest number of scans. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ..
7.13 Location ground truth drawn on a map using GPS data of the StudentLife dataset. .................................................. 123
7.14 The grid maps of ground truth locations of the three selected users. 124

A.1 The network of who-when-where co-patterns from the StudentLife dataset. .................................................. 130
A.2 Relation of a proximity topic and its corresponding location and time topics. .................................................. 131
A.3 Relation of a location topic and its corresponding proximity and time topics. .................................................. 132
List of Tables

3.1 Experiment results using three sub-datasets and different feature extraction methods. ........................................ 47
3.2 Experiment results with the MIT-BIH Arrhythmia dataset and our dataset recorded from Shimmer devices. .................. 49
5.1 Example of processed WiFi scans. ............................... 80
5.2 Example of processed Bluetooth scans. .......................... 80
5.3 Example of matched Bluetooth and WiFi scans. ................. 81
5.4 Example of extracted dataset. ................................. 81
5.5 The patterns of proximity (who), location (where) and time (when) .......................................................... 83
5.6 Performance of location clustering. ................................. 85
6.1 The patterns of proximity (who), location (where) and time (when) discovered by PS-HDP. ............................... 100
6.2 Performance of different clustering algorithms on the StudentLife dataset. .................................................... 106
6.3 Performance of PS-HDP with respect to different amount of missing data. .................................................... 106
7.1 Quantitative evaluation of performance of MC-iHMM and iHMM models on synthetic data. .............................. 119
List of Algorithms

1 Gibbs sampler routine. ........................................... 23
2 Collapsed Gibbs sampler routine. ............................. 23
3 Dataset creation for observation units of varying length. .... 46
4 Affinity Propagation (AP) algorithm .......................... 58
5 Pseudo-code framework for MCNC model. .................. 76
Abstract

Computing devices nowadays have become not only more powerful but also smaller and cheaper. As a consequence, they are now more mobile and affordable than ever before. It is common for a person to own more than one device, including a computer, tablet, smartphone, or smart-watch. The proliferation and ubiquity of personal computing devices enables the actualisation of Mark Weiser’s vision of ubiquitous or pervasive computing. Central to this proliferation is the massive accumulation of data, or ‘big’ data, and the quest to draw insights and intelligence from this. One of the biggest challenges is that pervasive data are typically noisy, heterogeneous, and incomplete due to the nature of the data collection process. To this end, it is an important research direction to develop advanced machine learning and data mining methods to learn the patterns from such data.

Leveraging on recent advances in the machine learning field, specifically in Bayesian nonparametrics, this thesis aims to make sense of pervasive signals by introducing a set of machine learning methods to address the aforementioned challenges. Our contributions through the thesis are presented as follows.

In chapter 3, we develop a machine learning-based approach to detect arrhythmia from electrocardiography (ECG) signals, which are inherently noisy. We demonstrate our methods on ECG data recorded from low-cost wearable ECG devices. There are a number of reasons why ambulatory ECG signals suffer from noise such as the inconsistency of contact between the sensor pads and the skin, and the low signal quality of sensors on affordable wearable devices in comparison to the high signal quality of clinical devices. Noisy signals make the detection of fiducial points in ECG signals more difficult, thus they adversely affect the detection of abnormal heartbeats. To tackle this problem, we propose a new approach to detect arrhythmia from ECG
signals which is able to accommodate both individual differences and suboptimal signal quality. At the centre of our approach is a temporal normalisation of ECG features which leads to a set of robust features even with poor signal segmentation. Moreover, because our normalisation method is done with a temporal window, its training process can be performed online and can adapt to situational changes.

Chapter 4 addresses the problem of significant location discovery from WiFi data captured by smartphones. Our approach does not use any geographical information or fingerprint data. It instead discovers significant locations by clustering access points in close proximity. Particularly, we use the Affinity Propagation (AP) algorithm which can automatically discover the number of clusters from data, thus the model can adapt well to changes in pervasive data. Moreover, by analysing the sequence of significant visited places, we can also learn the daily routines of individuals, which can help in the understanding of human dynamics.

From chapter 5 to chapter 7, we tackle the challenges of dealing with multi-channel and heterogeneous data as well as missing data. In chapter 5, we propose a novel Bayesian nonparametric (BNP) framework, termed the multi-channel nonparametric clustering (MCNC) model. This model is able to extract rich, high-order latent co-patterns (i.e., patterns that co-exist through different channels) from mixed-type data. The key idea of our model is to extend the machinery of current BNP models through the use of a richer product-space base measure. This kind of base measure enables the MCNC model to simultaneously explore data from multiple sources and extract hidden co-patterns, even if there are missing elements in the data. Additionally, as a BNP model, the MCNC model can automatically discover the space of latent patterns from the data, thus it does not need to specify the number of patterns in advance. We demonstrate the properties of our MCNC model on a synthetic dataset as well as on the StudentLife dataset – a real-world pervasive dataset collected from smartphones. We aim to discover who-when-where patterns, which are useful for human dynamic understanding and context-aware applications, from the Bluetooth and WiFi channels in the StudentLife dataset.

In chapter 6, we introduce the product-space hierarchical Dirichlet processes (PS-HDP) model to address a special case of clustering problem where multi-channel heterogeneous data can be grouped hierarchically. The major strength of the PS-HDP model is that it can combine the strengths of the MCNC model to simultaneously extract latent patterns from multi-channel heterogeneous data and the strengths of
the HDP model to group data in a hierarchical fashion.

In chapter 7, we present the multi-channel infinite hidden Markov model (MC-iHMM) to discover dynamic patterns from sequential multi-channel heterogeneous data. It combines the strengths of the MCNC model and the infinite hidden Markov model (iHMM) to learn patterns from sequential data. In short, the main features of the PS-HDP model and the MC-iHMM model still lie in the product-space approach to deal with multi-channel heterogeneous data. However, they have different model architectures to adapt to two different clustering problems: the hierarchical data clustering and the sequential data clustering. We demonstrate the two models on both synthetic and real-world pervasive datasets. We achieve comparable quantitative results to popular baseline methods. We also provide an interactive tool to visualise who-when-where patterns as a network so that it can facilitate the exploration of our experiment results.
Acknowledgements

“I have a lot to say but my words are few …

With you my life is amazing.”

Unknown

Time flies so fast. It has been more than 4 years from the first day I came to the PRaDA centre to start a hard but remarkable Ph.D. journey full of challenges and joy. Of course, not everything went smoothly along the journey. I had difficult moments where I struggled to conduct experiments and write papers. I also suffered from stress, home sickness, sleeplessness and financial hardship. Luckily, I had great supervisors, family and friends around me who encouraged and helped me to overcome the hard times and to persevere with my thesis. Needless to say, this thesis would never have been completed without their support. On this page, I would like to express my sincere thankfulnesses to these great people.

First and foremost, I would like to express my respect and gratitude to my principal supervisor, Prof. Dinh Phung. I learned a lot from you in both academic and non-academic matters. My words will never be enough to show my gratitude for guiding me through my Ph.D. journey, for shaping the ideas of my research papers, for helping me to structure this thesis, for your inspiration, patience and emotional support during my hard times.

I would also like to say thank you to my co-supervisors Prof. Svetha Venkatesh, Dr Wei Luo and Prof. Terry Caelli for their invaluable guidance, support and inspiration, especially during the first year of my Ph.D.
My deep gratitude also goes to Dr Thuong Nguyen and Dr Vu Nguyen, my technical advisors and collaborators. I will never forget your support and advice when I had difficulties conducting experiments, on sharpening my research papers and on revising this thesis. Also, I would send my sincere thank you to Ms. Michele Mooney for her diligent proofreading of this thesis.

Last but not least, I would like to send my heartfelt thanks to my beloved parents, Thao and Jolie, for their endless love and understanding. To Mum and Dad, I am forever indebted to you for your sacrifice and for everything you have done for me. To Thao and Jolie, I always remember your smiles, the funny things you have done for me and the time we have spent together.
Relevant Publications

Part of this thesis has been published or documented elsewhere. The details of these publications are as follows:

Refereed Conference Papers:


4. Thanh-Binh Nguyen, Vu Nguyen, Svetha Venkatesh, Dinh Phung. Learning Multi-faceted Activities from Heterogeneous Data with the Product Space Hierarchical Dirichlet Processes. In proceedings of the 3rd Workshop on

Awards

- I got the Finalist best IBM track 1 student paper award for the paper “MCNC: Multi-Channel Nonparametric Clustering from Heterogeneous Data” at the ICPR 2016 conference.
- I was the winner of the IEEE Deakin University Student Branch postgraduate poster competition for the paper “Unsupervised Inference of Significant Locations from WiFi Data for Understanding Human Dynamics” in 2014.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Affinity propagation</td>
</tr>
<tr>
<td>BNP</td>
<td>Bayesian nonparametrics</td>
</tr>
<tr>
<td>CRP</td>
<td>Chinese restaurant process</td>
</tr>
<tr>
<td>DP</td>
<td>Dirichlet process</td>
</tr>
<tr>
<td>DPM</td>
<td>Dirichlet process mixture</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation–Maximisation algorithm</td>
</tr>
<tr>
<td>HDP</td>
<td>hierarchical Dirichlet processes</td>
</tr>
<tr>
<td>HMM</td>
<td>hidden Markov model</td>
</tr>
<tr>
<td>iHMM</td>
<td>infinite hidden Markov model</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>independent and identically distributed</td>
</tr>
<tr>
<td>JS</td>
<td>Jensen–Shannon divergence</td>
</tr>
<tr>
<td>KL</td>
<td>Kullback–Leibler divergence</td>
</tr>
<tr>
<td>kNN</td>
<td>k-nearest neighbours</td>
</tr>
<tr>
<td>LDA</td>
<td>latent Dirichlet allocation</td>
</tr>
<tr>
<td>MCMC</td>
<td>Markov chain Monte Carlo</td>
</tr>
</tbody>
</table>

xix
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCNC</td>
<td>multi-channel nonparametric clustering</td>
</tr>
<tr>
<td>MDC</td>
<td>Mobile Data Challenge dataset</td>
</tr>
<tr>
<td>NMI</td>
<td>normalised mutual information</td>
</tr>
<tr>
<td>RI</td>
<td>rand-index</td>
</tr>
</tbody>
</table>
Introduction

“Ubiquitous computing names the third wave in computing, just now beginning.”

Mark Weiser, 1991

1.1 Motivations

Let us turn back time to the 1950s and 1960s, when computers were huge machines that occupied an entire large room, weighed several dozen tons, and were very expensive. Now in retrospect, we appreciate that these bulky machines played a very important role in history. They started the so-called first wave of computing, named the mainframe computing era, which lasted until around the 1980s. A centralised computing model is the main characteristic of this era, where each mainframe was shared by many people and served as a central business tool for large organisations.

Thanks to advances in semi-conductor technology, computers became smaller, light-weight, and cheaper. This led to the emergence of personal computers in the 80s which moved people away from mainframes since they could now afford their own computers for their personal use. It was a radical change in the computer-human relationship – from one-to-many in the mainframe computing era to one-to-one in the personal computing era. However, in this personal computing era, computers were still at the centre of human attention. For example, people had to type on keyboards to input data or give commands to computers. Thus, it required a degree
1.1. Motivations

of human effort to complete computing tasks.

In 1991, Mark Weiser told the world of his vision for the next era of computing in his seminal paper “The Computer for the 21st Century” (Weiser, 1991). He envisioned that in accordance with Moore’s law, computing devices in the future would be tiny, lightweight and very cheap. Thus, one individual could own several computing devices at one time. In this scenario, if the usage model of computing systems follows the same model as that of mainframe and personal computers, where a substantial effort is required on the part of users to accomplish any computing tasks, then users would be constantly distracted by these numerous devices (Adelstein et al., 2005). Therefore, computing should change from a computer-centered design, with the aim “to make a computer so exciting, so wonderful, so interesting, that we never want to be without it” (as of desktop and personal computers), to a human-centered design to make a computer “so imbedded, so fitting, so natural, that we use it without even thinking about it.” To achieve this aim, computers would be an invisible part of our everyday lives, and computing would change from desktop, personal computing to a more distributed, mobile and embedded form (Reis and Maximiano, 2016). Mark Weiser wrote “the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” Computational capability (generally in the form of microprocessors) will be embedded into everyday objects “to make them effectively communicate and perform useful tasks in a way that minimises the end users’ need to interact with computers as computers.” He termed this computing model ubiquitous computing, which lately is also known as pervasive computing. Unlike personal computing, ubiquitous/pervasive computing can occur on any device, at any time, in any place and in any data format across any network, and can handle tasks from one computer to another as, for example, a user moves from his car to his office. In short, the main idea behind ubiquitous computing is “instead of asking us to enter into the computer’s world, ubiquitous computing would bring computers into our world.”

Nowadays, pervasive computing surrounds us, with or without our notice, and

---

1Since the mid-1990s, ubiquitous computing also has been known as pervasive computing. Despite claims that these two terms are not equivalent, they are usually used interchangeably in the field of computer science. Readers are referred to (Reis and Maximiano, 2016, chapter 1, page 3) for a detailed discussion on the two terms.

2http://internetofthingsagenda.techtarget.com/definition/pervasive-computing-ubiquitous-computing
benefits us in many ways. We list some examples of pervasive computing in daily life as follows.

- Digital cameras usually support two modes of operation: manual or automatic. In manual mode, users have to set parameters such as ISO (light sensitivity of the image sensor), exposure, aperture and shutter speed. To be able to use such cameras effectively, one has to learn a lot about the meaning of these parameters and how to combine them to obtain a good picture in a specific condition. In automatic mode, digital cameras use their image processor and sensors to analyse the view and compute a good set of parameters for the camera, hence helping the users to avoid the difficult task of setting the parameters.

- Recent smartphones are equipped with many sensors, including ambient light sensors. Using information from the ambient light sensors, a smartphone can know how dark (or how bright) the environment is, then automatically adapt the screen brightness so that users can read texts or see photos on screen better.

- Reading information from built-in accelerometers, which reflects its own movement, a smart watch can keep track of the activities which the users have performed (e.g. the number of steps taken), then report to the users on how much energy they have burnt in a day, or how long they have been asleep.

- All recently manufactured cars are equipped with the Anti-lock Braking System (ABS) as standard safety equipment. Usually, stopping a car quickly on a slippery road can be very challenging. A heavy brake could result in the wheels locking up. This leads to two consequences: first, the wheels are skidding, which makes it longer for the car to stop, and second, as the wheels lock up, the driver is unable to steer the car. The ABS uses wheel speed sensors to detect if one or more wheels are trying to lock up during braking. If detected, a processor will control a series of hydraulic valves to reduce the braking on that wheel, thus preventing skidding and allows the driver to maintain steering control.

- A recent and well-known example of pervasive computing is FaceID technology. It is available on iPhone products and is used as a security mechanism to unlock the iPhone, to purchase apps from the Apple store, or to use the Apple Pay digital wallet. FaceID is a new facial recognition system which obtains

\[^3\text{https://www.tc.gc.ca/eng/motorvehiclesafety/tp-tp13082-abs1_e-214.htm}\]
input data from a True Depth camera and then processes this data using an artificial intelligent system to recognise the face. The True Depth camera is in fact a system of many digital sensors such as a regular front camera, a flood illuminator, an infrared camera, and a dot projector. It makes the iPhone unlock process simpler, more natural and effortless. It works seamlessly and users are not aware of the security check that had done to unlock their iPhone.

Certainly, pervasive computing also comes with challenges in various research areas, such as human-computer interaction, hardware designs, information management and artificial intelligence have been identified in (Abowd and Mynatt, 2000; Satyanarayanan, 2001; Conti et al., 2012). Especially, in comparison to other kinds of data, pervasive data are well-known by their mass-volume and uncertainty. Let us clarify by taking a wearable electrocardiogram (ECG) device as an example. This device is used to continuously monitor electrical activities of the heart. To achieve an adequate ECG signal, the device has to record from 128 to 512 data points per second, thus it can generate a huge amount of data everyday. Moreover, as the device is used to record data in a daily life environment (instead of a controlled environment such as in a hospital), its data are typically affected by human movements which result in the data containing more uncertainties. For these reasons, making sense of these enormous and uncertain pervasive data is one of the biggest challenges of pervasive computing. Tackling this challenge is not only necessary but also a must to enable a pervasive intelligent system to have the capabilities of decision making or task planning. Therefore, relevant machine learning and data mining techniques must be developed to extract complex hidden patterns from pervasive data. This is the main aim of our work throughout this thesis.

1.2 Aims and Approaches

The overarching aim of this thesis is to develop methods and techniques to infer complex patterns from pervasive data collected by heterogeneous sensors. Of the challenges related to this aim, the four most pressing challenges are:

- Pervasive devices may generate outputs with noises and variations. A pervasive system typically employs a lot of sensors to capture environmental and users’ information. Thus, to reduce the cost of the system, low-cost sensors, which produce larger noises than state-of-the-art sensors, are usually used.
More importantly, pervasive data are collected in the wild in an uncontrolled environment and as a result, pervasive data are typically noisy and disrupted.

- Different sensor devices have different data types and formats. For example, an accelerometer measures the vibration of the motion of an object and outputs 3-channel continuous values of acceleration along 3 axes X, Y and Z; while a door sensor generates binary values to indicate the opening or closing state of the door. Thus, pervasive data are generally multi-channel and heterogeneous. These issues are problematic for traditional machine learning methods to learn the underlying patterns in pervasive data.

- The environmental parameters and users’ behaviours in some cases are very complex and might change over time. Thus, people usually do not know in advance the number of contexts and hidden patterns in pervasive data. This issue makes traditional parametric machine learning methods impractical.

- Pervasive signals typically come in a stream and/or grow over time with “no clear beginning and ending” (Abowd and Mynatt, 2000).

The above challenges lead to the following research questions:

- How to factor out the noises and variations in low-cost pervasive monitoring devices and how to capture the high-level and rich patterns from physical signals without them being too sensitive to the noise inherent in low-cost pervasive devices?

- How to consolidate signals from heterogeneous devices with different data formats and statistical properties in a consistent framework to learn rich patterns that otherwise will not be able to be extracted from individual data channels?

- How to handle model complexity where the underlying structure of data might be changed?

- How to address some special and practical cases of pervasive complex data where data are time-dependent or hidden patterns embedded in data can be grouped hierarchically?

- How to make sense of pervasive data in some important applications such as healthcare or understanding human dynamics?
Departing from a machine learning and data science foundation, this thesis aims to develop methods and approaches to answer these questions. We address the first question by focusing on the low-level features in the data, and rely on high-level machine learning models to tackle the remaining questions. Unlike traditional supervised learning researches where a huge amount of label data is available and required for learning, data collected from wearable devices generally lack of ‘ground truth’ or having weak supervised information. Our approach hence focus on unsupervised learning research where our algorithms are designed to be exploratory and perform clustering and discovery of patterns.

Another important requirement for our current work is to deal with uncertainty in data, measurements as well as models. This translates to machine learning methods which are robust, flexible to deal with data types and are able to incorporate priors. Our approach is then to ground the work in Bayesian unsupervised learning. In addition, to enable automatic pattern discovery, we apply Bayesian nonparametric approaches to these developed techniques. More specifically, our research aims to develop:

- A normalisation method to adjust for the temporal drifting and augment subject-specific and device-specific.
- An extension of current Bayesian nonparametric models with a richer product-space base measure to model data on multiple channels and mixed-types.
- A HDP-like model with a product-space base measure to capture the hierarchical underlying structures of pervasive complex signals.
- A HMM-like model with a product-space base measure to capture temporal and sequential characteristic of pervasive complex signals.

Building on the processing models and the sensing platform, our research can also facilitate several applications including long-term health monitoring of patients in their everyday environment with automatic alerts when abnormal physiological signals are detected, or discovering individuals’ routines and interaction patterns to assist context-aware applications.
1.3 Contributions and Significance

The main significance of our thesis is that it bridges the gap between pervasive computing and machine learning by leveraging recent advances in Bayesian non-parametrics for extracting patterns from complex pervasive data. While our work still remains at algorithmic and proof-of-concept level, it has benefit in pervasive computing applications where signals are noisy, disrupted, heterogeneous and complex. Hence, at the intersection between applied machine learning and pervasive computing, it has, at least, the following two benefits:

- It contributes to new techniques in Bayesian nonparametric modelling to learn rich latent co-patterns from multi-channel and heterogeneous pervasive data, even if there are missing elements in the data. These co-patterns, which occur across all data channels, will not be able to be extracted from individual ones. This allows different types of sensors in a pervasive computing system to collaborate seamlessly.

- It has strong applicability to real-world pervasive applications. Our methods have a solid foundation in Bayesian nonparametric theories which allow the models to adaptively grow with new incoming data; thus they are particularly suitable for pervasive applications where data are collected in real-world settings and typically arrive as a stream. Moreover, our methods are developed on real-world datasets collected using wearable devices or mobile phones during daily life activities, thus they are consequently well-suited to real-life applications.

Specific contributions and significance of this work are:

- Chapter 3 addresses the problem of noisy pervasive data where we developed and demonstrated a temporal normalisation method for ECG signals collected from low-cost wearable devices. Our method can accommodate both individual differences and suboptimal signal quality and leads to a set of robust features to address the problem of arrhythmia detection using nonlinear classification methods. Therefore, our work amplifies the utility and value of low-cost wearable health monitoring devices.

- Chapter 4 tackles the problem of understanding human dynamics using pervasive signals. Specifically, we use unsupervised methods to discover significant locations and the daily routines of individuals from WiFi data captured by their
smartphones. Our experiment results show a significant correlation between the location distributions with the age and occupations of individuals.

- Chapter 5 presents a novel Bayesian nonparametric model, termed MCNC, for discovering latent patterns from multi-channel and heterogeneous data. This model is particularly useful for pervasive computing systems with multiple types of sensors, each with its own data format and statistical properties. With this model, we also contribute to Bayesian data modelling using product-space base measure distributions.

- Chapter 6 and chapter 7 present two variants of the MCNC model to address two specific cases of hierarchical data clustering and sequential data clustering in which data comprise multiple channels and mixed types.

1.4 Structure of the Thesis

The remainder of this thesis is organised as follows.

Chapter 2 provides the preliminary background and literature review related to the research conducted in this thesis. First, it briefly introduces the fundamentals of the pervasive computing area and the challenges arising from pervasive data. Then, it provides a systematic review of existing machine learning approaches for pattern discovery from pervasive data. Particularly, the section puts the focus on unsupervised learning techniques (e.g., clustering, latent feature analysis or topic modelling). It also provides a preliminary background on Bayesian nonparametric modelling, including the Dirichlet process, infinite mixture modelling by the Dirichlet process mixture model and the hierarchical Dirichlet processes model. This background paves the way for our proposed frameworks in the next chapters.

In Chapter 3, we tackle one of the challenges of pervasive data analysis – data with noises. Our focus is on data collected from wearable devices, which are usually noisy due to the nature of the data collection process. More specifically, we aim to detect abnormal heartbeats in the ambulatory electrocardiogram (ECG) data collected by wearable Shimmer ECG devices. We propose a new temporal normalisation method which leads to a set of robust ECG features even with noisy data. In combination with nonlinear machine learning classification models, our approach achieves higher performance than the rule-based approach, which is used by clinical physicians.
Chapter 4 presents a machine learning approach for an essential component of context-awareness application – the discovery of significant locations and daily routines of individuals from WiFi signal collected by smartphones. Assuming that WiFi hotspots are immobile, we group the hotspots captured by the smartphones which are in close proximity and consider each group a significant location. We then analyse the sequence of visited places of each individual to discover his/her daily routines. Finally, we examine the entropy of location distributions of individuals, which shows significant correlation with their age and occupations, to understand human behavior.

In chapter 5, we present a novel approach to the discovery of co-patterns from multi-channel and heterogeneous data. Our proposed model, termed the multi-channel nonparametric clustering (MCNC) model, leverages on the recent advances of Bayesian nonparametrics (BNP) theory. At the centre of the MCNC model is a richer product-space base measure, which allows the model to simultaneously explore data and extract hidden co-patterns which occur across data channels, even if data have missing elements. Moreover, as a BNP model, it can also automatically infer the number of latent patterns from data, thus it does not require this number to be specified in advance. We demonstrate the model on the StudentLife dataset to discover who-when-where patterns, which are essential for understanding human dynamics or for context-awareness applications.

Chapter 6 addresses a special case of pervasive data where latent patterns can be grouped hierarchically. Our model, termed product-space hierarchical Dirichlet processes (PS-HDP), combines the strengths of the MCNC model and the HDP model to simultaneously extract latent patterns embedded in multi-channel heterogeneous data in a hierarchical fashion.

In chapter 7, we present a rich Bayesian framework for dynamic temporal pattern discovery from pervasive data. The model, termed the multi-channel infinite hidden Markov model (MC-\iHMM), is a variant of the MCNC model which is able to extract co-patterns from sequential data. In short, its strengths still lie in the product-space base measure approach, but its model architecture is inherited from the \iHMM model.

Finally, chapter 8 concludes the thesis with a summary and discussions on the further research directions of the thesis.
In this chapter, we provide a discussion on the background to the research conducted in this thesis and the related work. It contains two main sections. In the first section, we provide the relevant background on pervasive computing and a literature review of the recent research on selected pervasive applications, including pervasive computing for healthcare applications and pervasive computing for understanding human dynamics. The second section presents an overview of applied machine learning with an emphasis on the foundation of Bayesian nonparametric data modelling and recent advances in this field.

2.1 Introduction to Ubiquitous and Pervasive Computing

In this section, we briefly introduce the domain of ubiquitous and pervasive computing. Excellent and comprehensive surveys and important research literature in this field can be found in (Weiser, 1991, 1993; Weiser and Brown, 1997; Satyanarayanan, 2001; Hightower and Borriello, 2001; Krumm, 2016).
2.1.1 Pervasive and Ubiquitous Computing

As discussed in chapter 1, Mark Weiser envisioned the pervasive computing area in which computing devices should be easy to use and people should not need any specific skills to use these devices other than regular skills used in their daily life. According to Weiser, pervasive computing is “the calm technology, that recedes into the background of our lives” which is available anytime, anywhere, and can occur using any device, in any location and any format. In a more technological way, Sakamura and Koshizuka (2005) described ubiquitous computing as a new trend of information and communication technologies, in which a huge number of tiny computers are embedded into an invisible part of the fabric of everyday life. These computers are equipped with sensors and/or actuators that interact with our living environment, and with communication functions to exchange data.

Typically, a pervasive computing system has the three main characteristics (Nguyen, 2015a):

- **Embeddedness**: computing hardware and software should be embedded into daily life objects and can appear anytime and anywhere.

- **Context-awareness**: the system should have context-awareness to use the surrounding information.

- **Adaptation**: the system should react and behave adaptively to changes in its environment.

In addition to these three core characteristics, the research community has identified some other characteristics of a pervasive computing system (Graham, 2007):

- **Transparency**: the computing environment should be transparent to the users.

- **User-centric**: pervasive computing devices should be designed to serve their users in the easiest way possible.

- **Automatic**: the system should react automatically to meet the users’ needs. This is done with the help of context discovery from the information collected from the users and the environment.

To achieve these goals, pervasive computing has to leverage the advances in multiple research fields including:
• **Hardware and sensor requirements:** the computing units and sensors need to be small enough so that they can be integrated into daily life objects. Furthermore, they also need to be inexpensive as a large number of these devices are needed to build a pervasive system.

• **Networking and connectivity:** an essential key for context discovery and the automatic adaptation ability of a pervasive system is the information exchange between devices in the system. Thus, these devices need to be connected. Fortunately, wireless connectivity is highly advanced and extremely popular nowadays. The bandwidth of these wireless connectivities is wide and the speed is fast enough to allow pervasive devices to instantly interconnect to each other and to the Internet.

• **Data communication and management:** a pervasive system typically has a large number of sensors which generate a massive amount of data. Moreover, they are typically heterogeneous. Thus, data management in pervasive computing faces a number of challenges (Perich et al., 2004): spatio-temporal variation of data; the lack of a global catalogue and schema; no guarantee of reconnection and collaboration.

• **Context reasoning and management:** the raw data captured by sensors are typically too coarse for the system to use as instructions. Instead, high level information needs to be obtained by extracting latent patterns from the sensing data. This is where machine learning methods can help to make sense of pervasive signals.

• **Applications and services:** pervasive applications have to work on a wide range of devices, including mobile and wearable devices. These devices have limited resources, thus they require new programming techniques. Moreover, applications and services have to be user-centric, requiring new design and development paradigms.

• **Human-computer interaction:** pervasive systems need to support natural human behaviors (e.g., voice, gesture, movement) as interaction methods between users and devices to minimise the specific skills that the users are required to learn to use the systems.
2.1.2 Context Acquisition and Understanding Human Dynamics

As discussed in the previous chapter, the main focus of this thesis is to develop machine learning methods to make sense of pervasive signals to discover latent patterns which are useful for context discovery. In this section, we present the background on contexts and one of its important applications in context discovery – understanding human dynamics.

2.1.2.1 Context and Context Acquisition

Although identified as one of the core characteristics of pervasive computing, the term context was not clearly defined in the seminal paper of Weiser (1991). Since then, there have been various definitions of the terms context and context-awareness. We present the most accepted definitions of these terms introduced by (Dey, 2001) and (Abowd et al., 1999).

**Context** is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

A system is **context-aware** if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task.

We refer readers to (Nguyen, 2015a, section 2.1) for more information about context and context data acquisition.

2.1.2.2 Understanding Human Dynamics

Human dynamics comprises the actions and interactions of personal, interpersonal, and social/contextual factors and their effects on behavioral outcomes. Research in this area has significantly increased from the work of Barabási (2005), who developed a model known as Barabasi’s model which can help in the understanding of different aspects of human behaviors.

Human dynamics can be described as physical activity patterns or as interaction patterns between people in a physical social network. To discover physical activity patterns, accelerometer signals are usually used. For example, Duong et al. (2005) used a switching hidden semi-Markov model to recognise different classes of activities and to also detect abnormal behaviors. Huynh et al. (2008) used topic models to extract physical activities from the accelerometer signals and used them to discover people’s activity patterns, while Duong et al. (2009) used a Coxian model for efficient human activity recognition. To understand human dynamics in the physical social network, the interaction patterns need to be extracted first. The most popular type of sensor for this task is a Bluetooth device that can record the surrounding devices. Groups of people who are usually in close proximity can be discovered from this type of data. For example, Eagle and Pentland (2009) used principal component analysis (PCA) approach to discover typical proximity patterns. Do and Gatica-Perez (2011, 2013) proposed an LDA-based probabilistic model to extract groups of users from Bluetooth data. They then used the additional information to find the social context for these groups (e.g. meeting, lunch time). Nguyen et al. (2013) and Phung et al. (2014) used HDP to learn the social interaction contexts from Bluetooth signals. They then used the extracted contexts to understand human dynamics by clustering the users according to their interaction behaviors. Nguyen et al. (2014a) proposed an incremental approach to discover the social interaction contexts using the fixed-lag particle filter on the Indian buffet process (IBP).

Using location to discover human dynamics is widely explored in the literature. Location information is mainly used to gain an understanding of human mobility in daily life. For example, González et al. (2008) built a time-resolved trajectory of users based on location sequences. From these trajectories, they learned a distribution of displacement of all users and modelled the mobility patterns of individuals by this probability density function. Do and Gatica-Perez (2014) aggregated all the available spatial data (GPS, GSM cell towers, WiFi access points, Bluetooth) to characterise locations and discover the mobility patterns of individuals via their visiting patterns to different place categories. These types of data were also used in (Montoliu and Gatica-Perez, 2010) to discover the places of interest of smartphone users. Phung et al. (2009a,b) proposed a novel approach (by leveraging clustering techniques) to recover user context that includes user motion state and significant places the user visits from WiFi access point ID and signal strength. Then, higher context levels (e.g. user rhythms, sequences of places) were inferred using LDA.
2.1. Introduction to Ubiquitous and Pervasive Computing

Location Inference from Pervasive Signals  Direct spatial data are the most popular source for inferring significant locations, thanks to the increasing availability of integrated sensors in wearable devices. Most smartphones now include a GPS, WiFi, Bluetooth and accelerometers. Given a clear sky view of the device, the GPS sensor can provide the exact geographic position. The other types of sensors (WiFi, Bluetooth) are used as an additional source when the GPS signal is unavailable.

The approaches for learning spatial location can be divided into two categories: geometry-based and fingerprint-based approaches (Dousse et al., 2012; Kim et al., 2009; Montoliu and Gatica-Perez, 2010). Geometric-based techniques produce coordinates, circles or polygons to describe places, while fingerprint-based techniques detect stable radio environments that indicate a stay but provide no geographical location for the place.

In the geometry-based approach, the most common framework is firstly pre-processing and creating location points from spatial data and secondly inferring stay points from clusters of location points. One example of a location point is a triple of GPS Longitude, GPS Latitude and the time where the GPS coordinates are recorded. Stay points, on the other hand, represent a geographic region where the user tends to stay for a significant duration. Generally, stay points are usually locations that can be associated with meaningful labels (e.g. “home”, “office”) and therefore they are much more interesting than the coordinate numbers of the longitude and latitude of location points. This approach can be found in several studies (Ashbrook and Starner, 2003; Kim et al., 2009; Ye et al., 2009; Do and Gatica-Perez, 2014).

Most geometry-based studies use GPS data to infer locations for location history (for example, in Ashbrook and Starner, 2002; Ye et al., 2009). However, as previously mentioned, a GPS has drawbacks in relation to coverage duration, start-up time as well as indoor signal quality. To overcome these drawbacks, some work integrated more location sources available on smartphones such as GSM cell towers, WiFi or Bluetooth signals to infer locations. This approach can be found in Dousse et al. (2012); Kang et al. (2004) and Montoliu and Gatica-Perez (2010).

In the fingerprint-based approach, a place is defined by its fingerprint, typically as a vector of currently detectable GSM cell towers or WiFi access points. The place is recognised when its fingerprint is detected. This approach has an important advantage in that it does not require the geolocated information of the GSM towers or the WiFi access points. It is useful for applications that only need to
know the semantic meanings of the places and not the exact location of the places. Two examples using this approach are BeaconPrint (Hightower et al., 2005) and PlaceSense (Kim et al., 2009). In Hightower et al. (2005), the authors created fingerprints from a set of beacons captured from GSM and WiFi radio, then merged similar fingerprints to find a stable set of fingerprints, which is equivalent to a place. This approach works well in cities or similar places where GSM and WiFi radio usually have good coverage. Recently, in Kim et al. (2009), the authors proposed an algorithm to accurately discover the entrance or the departure of a user to a place based on the timestamped log of radio beacons. This algorithm can also be adjusted to different types of radio frequency (RF) beacons, thus better detecting stable sets of fingerprints than the BeaconPrint algorithm.

WiFi-base Location Discovery WiFi-enabled portable devices are changing our daily lives. Today we have more mobile phones than people living in the world\(^2\). More than half of the US population own smartphones, with the percentage projected to reach 67.8% by 2017\(^3\). Of the smartphone users, 79% have their phones with them 22 hours a day. Portable devices have certainly become part of our lives.

Due to the ubiquitous presence of smartphones, they provide a good indicator of a user’s location, and lead to an active research area of location and context-aware applications (Banerjee et al., 2002; Schilit et al., 1994). Beyond location-aware applications, location information has been extensively harvested to understand human dynamics (Adams et al., 2006; Dong et al., 2011; González et al., 2008; Ye et al., 2009) and to improve well-being (Belik et al., 2009; Moturu et al., 2011).

To now, GPS is the most popular means to inform location. This data include the longitude and latitude coordinates of the device at the time of scanning and can be easily clustered using unsupervised learning techniques. However, while it is suitable for a certain class of real-time applications such as navigation, GPS usage can drain the battery quickly. Moreover, the GPS sensor needs a long start-up period before it can capture a position accurately. It also needs a clear sky view to perceive the signal from satellites; thus it is unsuitable for indoor use. The US EPA (United State


Environment Protection Agency) estimated that Americans spend on average 90% of their time indoors\(^4\). An experiment conducted by Kim et al. (2009) showed that on average, GPS coverage is available only 5–30% of the time during a typical day of a user. Therefore, while providing relatively accurate position information, a GPS is not suitable for tracking location to understand human dynamics, where long-term data collection requires minimum power consumption and indoor connectivity.

In contrast to GPS sensors, WiFi devices consume much less power and connect to WiFi access points in both indoor and outdoor settings. Today, most WiFi access points have a static location. As a part of the IEEE 802.11 wireless LAN standard, a public WiFi access point frequently broadcasts its identity information (including its MAC address which is unique to each device) so that any smartphone can easily catch them. By scanning for these WiFi signals, we can know what nearby WiFi access points are. The visibility of a WiFi access point can be established by simple scans, which is much faster than establishing GPS connections. The number of public and freely available WiFi access points is increasing rapidly and totaled 5.8 million in 2015\(^5\). This makes WiFi access points suitable for location discovery.

In addition to GPS and WiFi signals, Bluetooth sensors are also used in some work to discover the proximity between users. In (Nguyen et al., 2013) and (Phung et al., 2014), the authors used Bluetooth devices carried by the users to detect their proximity and potentially their co-location. However, due to the mobility of the devices, this method cannot always give the information about the users’ location.

WiFi access points have been widely used for location detection. Typical WiFi-based location recovery relies on supervised techniques that require labeled data to train a classification model (Bell et al., 2010; Kelly et al., 2009; Yang et al., 2008). But labeled data are often difficult to obtain in real-world applications, and the training step needs to be rerun when data on new locations arises.

**Significant Location Discovery as a Clustering Process**  As described in section 2.1.2.2, an important step in significant location discovery is grouping similar location points into stay points, which is often done through clustering techniques.


K-means is a simple but popular clustering algorithm. K-means and its variants have also been applied to the location discovery task. In Ashbrook and Starner (2002), one of the first geometry-based place learning approaches, stay points created from GPS data are clustered in places using the k-means algorithm. The signatures of the discovered locations are then inputted into Markov models to learn movement patterns of the user. However, one important drawback of the k-means algorithm (and its variants) is that it requires the number of locations to be specified in advance, an assumption that is often not true in many applications.

Another preferred clustering approach is density-based clustering algorithms. Two of the most important density-based algorithms are DBSCAN (Ester et al., 1996) and its variant OPTICS (Ankerst et al., 1999). In contrast to the k-means algorithm, DBSCAN and OPTICS do not require the number of clusters to be specified beforehand. It can also find arbitrarily shaped clusters and remove “noisy” data points. For example, Ye et al. (2009) used OPTICS to detect a stay point sequence and then build a location history based on stay point clusters. As reported in the work of Montoliu and Gatica-Perez (2010), density-based clustering algorithms achieved better results than k-means. Generally, these algorithms work well for spatial 2D data with a proper distance metric among data points. However, when dealing with high-dimensional data, they are considered inefficient due to their reliance on the MD-tree structure to maintain the distances among data points.

Some other clustering techniques have also been developed to overcome the specific problems of spatial data. For example, to avoid dependence on properties of GPS satellite signals, Kang et al. (2004) designed a new time-based clustering algorithm to take advantage of the continuity of WiFi positioning and find significant places from timestamped coordinates derived from any location system. In another work, Zheng et al. (2010) developed a grid-based clustering algorithm to make constraints on the cluster size.

Bayesian non-parametric clustering methods automatically adjust the number of clusters to fit the data. They also work on multi-dimensional data. However, early attempts to use the Dirichlet process mixture (DPM) model to cluster GPS data was applied on 2D data for which DBSCAN has been known to work well (Nurmi and Bhattacharya, 2008). In addition, computation in DPM is expensive due to its MCMC sampling, which is prohibitive being in the scale of millions of data points. Recently, a new Bayesian non-parametric clustering method named
2.2 Data Modelling and Bayesian Approach

DP-means (Kulis and Jordan, 2012), which is a k-means-like clustering algorithm constructed from the Bayesian non-parametric viewpoint, has been developed to overcome the computational bottleneck in DPM.

In summary, most of the existing approaches for significant location discovery require either the spatial location or fingerprint data. The spatial location might conflict with user’s privacy. It is also not easy to obtain and might require a complicated algorithm if the GPS signal is unavailable. Fingerprint positioning methods require a large amount of data, which is costly to collect, to learn a-priori information. To address these problems, we propose a novel approach to discover significant locations using WiFi data only without any spatial location information or fingerprint data.

2.2 Data Modelling and Bayesian Approach

We start this section by introducing a simple but powerful framework for data modelling, the parametric mixture model. Then, we describe the Dirichlet process mixture (DPM) model, a Bayesian approach for mixture models with an infinite capacity of the number of parameters to overcome the shortcomings of the parametric mixture model. Next, we present the hierarchical Dirichlet processes (HDP), a multi-level Bayesian nonparametric model which is able to model groups of data where each data point within a group is a draw from a mixture model. After this, we detail the hidden Markov model (HMM) which is well-known for modelling sequential data, such as data captured from pervasive systems, and the infinite hidden Markov model (iHMM), the Bayesian nonparametric version of HMM. Finally, we end the section by introducing approaches for multi-level clustering and multi-channel data clustering.

2.2.1 Mixture Modelling via Topic Models

2.2.1.1 Finite Mixture Model

Simple data could be represented by a single mode distribution, such as a multivariate Gaussian distribution. However, it is usually insufficient to use these single mode distributions to represent more complex data of which the density function has multiple modes. For this reason, the mixture model is proposed to represent such kinds of data. Intuitively, mixture model is a probabilistic model that represents an
2.2. Data Modelling and Bayesian Approach

Figure 2.1: Graphical representation of the finite mixture model.

overall population with the presence of $K$ subpopulations by using a mixture of $K$ component distributions where each component distribution is a representation of the corresponding subpopulation.

Formally, a mixture model can be any convex combination such as $\sum_{k=1}^{K} \pi_k f_k(x \mid \phi_k)$ with $\pi_k$ being the mixing weights such that $\pi_k > 0$, $\sum_k \pi_k = 1$, $\phi_k$ being a parameter for each mixture component $k$, and $f(\cdot \mid \phi_k)$ being a probability density function given the component’s parameter $\phi_k$. Then, data points are represented under a finite mixture model as:

$$p(x) = \sum_{k=1}^{K} \pi_k f_k(x \mid \phi_k) \tag{2.1}$$

with the $\pi_k$ being the mixing weights, $\pi_k > 0$, $\sum_k \pi_k = 1$.

Equation 2.1 gives us a way to generate new data points in two steps: first, choose a component distribution $f_k$ with the probabilities given by the mixing weights $\pi_1, \pi_2, \ldots, \pi_K$, and then, generate one observation according to that distribution.

$$z_i \sim \text{Mult}(\pi_1, \pi_2, \ldots, \pi_K)$$

$$x_i \mid z_i \sim f_{z_i}$$

where $N$ is the total number of observations, $\{x_i\}_{i=1}^{N}$ is the data observations and $\{z_i\}_{i=1}^{N}$ is the latent variables, assigning an observation $x_i$ to the component $k$ such that $k = z_i$. Figure 2.1 shows the graphical representation of the finite mixture model.

In practice, we usually see all $f_k$ are from the same family, but with different
parameters, e.g. all $f_k$ are Gaussians with different means and variances. Hence, the formula 2.1 could be rewritten as:

$$p(x) = \sum_{k=1}^{K} \pi_k f(x \mid \phi_k)$$

where $\phi_k$ is the parameter of the component $k$ and $f(\cdot \mid \phi_k)$ is the probability distribution parametrised on $\phi_k$.

Given a set of data $\{x_i\}_{i=1}^{N}$, parameters $\pi = (\pi_1, \ldots, \pi_K)$ and $\{\phi_k\}_{k=1}^{K}$ can be estimated using the Expectation-Maximisation (EM) algorithm (Dempster et al., 1977). This algorithm repeatedly iterates over two steps: E-step and M-step, where the E-step calculates the expected value of the likelihood function of all data points over the current estimated parameters, and the M-step finds the set of parameters that maximises the expectation function in the E-step.

### 2.2.1.2 Bayesian Mixture Model

One of the drawback of the mixture model is that the learned models can easily over-fit to the data. This can be overcome using the Bayesian approach for the mixture model.

**Prior Distributions and Bayesian Approach** Under Bayesian formalism, the parameters of the mixture model have been drawn from prior distributions. These prior distributions represent the prior knowledge on the distributions of the parameters of the model. Since the parameters are drawn from prior distributions, they avoid fitting to a local optimum.

Figure 2.2 shows the graphical representation of the Bayesian mixture model, where the mixing weight vector $\pi$ is assumed to be drawn from a Dirichlet distribution with parameter $\alpha$: $\pi \sim \text{Dir}(\alpha)$, and the component parameter $\phi_k$ is generated from a prior distribution $\phi_k \sim H(\lambda); \forall k \in \{1, 2, \ldots, K\}$. The posterior distributions of parameters $\pi$ and $\phi_k(s)$ are typically inferred using the Markov Chain Monte Carlo (MCMC) sampling method.

**MCMC for Posterior Inference** Monte Carlo methods (Metropolis et al., 1953) provide complementary solutions for estimating posterior distribution in Bayesian
2.2. Data Modelling and Bayesian Approach

Figure 2.2: Graphical representation of the Bayesian mixture model.

inference. In contrast to optimisation approaches (e.g. EM algorithm), they are guaranteed to give a precise estimation, provided that the methods run a sufficient amount of computation. MCMC (Andrieu et al., 2003) is a special class of Monte Carlo methods where the sampling elements $X_1, X_2, \ldots, X_n$ satisfy the Markov conditions (see section 2.2.2.2).

There are some widely used MCMC methods, such as importance sampling, the Metropolis-Hasting algorithm, Gibbs sampling and its variants such as blocked Gibbs sampling and collapsed Gibbs sampling. We briefly present collapsed Gibbs sampling in the following section as it is used intensively in our work.

Collapsed Gibbs Sampling

The general idea of Gibbs sampling (Turchin, 1971) is to find the stationary distribution of the target distribution, such as the posterior distribution in the Bayesian mixture model. It is useful in case we do not know the joint distribution or in case it is hard to directly sample from the joint distribution, but the conditional distribution of each variable is known and is relatively easy to sample from. Gibbs sampling works by generating a data point from the distribution of each parameter (or variable), conditional on the current values of the other parameters. Algorithm 1 shows the Gibbs sampler routine for a set of three parameters $A, B, C$.

In case one or more variables can be integrated out from the conditional distribution of a given variable, we can use collapsed Gibbs sampler, a variant of the Gibbs sampler, to reduce the computation. For example, a collapsed Gibbs sampler generates a sample
Initialize: $A^0, B^0, C^0$

for $t = 1, \ldots, T$ do
\[
\begin{align*}
A^{t+1} &\sim p(A \mid B^t, C^t) \\
B^{t+1} &\sim p(B \mid A^{t+1}, C^t) \\
C^{t+1} &\sim p(C \mid A^{t+1}, B^{t+1})
\end{align*}
\]
end

Return: $A, B, C$

Algorithm 1: Gibbs sampler routine.

for variable $A$ using the marginal distribution $p(A \mid C) = \int_B p(A \mid B, C)p(B)dB$, with variable $B$ integrated out in this case. Algorithm 2 shows the collapsed Gibbs sampler routine for a set of three variables $A, B, C$ with $B$ integrated out.

Initialize: $A^0, C^0$

for $t = 1, \ldots, T$ do
\[
\begin{align*}
A^{t+1} &\sim \int_B p(A \mid B, C^t)p(B)dB \\
C^{t+1} &\sim \int_B p(C \mid A^{t+1}, B)p(B)dB
\end{align*}
\]
end

Return: $A, C$

Algorithm 2: Collapsed Gibbs sampler routine.

2.2.1.3 Dirichlet Processes

Probability theory provides a solid and consistent methodology to systematically represent and handle uncertainty in data modelling. In particular, mixture modelling represents a powerful latent variable approach to learn patterns from complex data. However, in the case of pervasive computing, data grow in complexity as they are often collected in the wild. It makes existing parametric mixture models, in which the number of mixture components and the corresponding parameters are fixed (i.e. the number of components $K$ has to be specified beforehand), become less practical as these models will reach a limit point where they cannot be grown to fit the data. Fortunately, this shortcoming of parametric mixture models could be overcome thanks to a recent advance in Bayesian nonparametrics which has extended the conventional finite mixture models to have infinite capacity through the Dirichlet processes (DP).
To obtain a solid understanding of DP, we first present the Dirichlet distribution, which is a probability distribution over a simplex space $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ where $\pi_k \in (0, 1)$ and $\sum_{k=1}^{K} \pi_k = 1$. It is often denoted $\text{Dir}(\alpha)$ with parameter $\alpha$ being a vector of positive reals, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K) > 0$, and has the probability density function:

$$f(\pi, \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha_k-1}$$

where $B(\alpha)$ is the normalising constant.

As a draw $\pi$ is a vector with $\pi_k \in (0, 1)$ and $\sum_{k=1}^{K} \pi_k = 1$, it can also be thought of as a probability mass function of a multinomial distribution of $K$ events. Thus, the Dirichlet distribution is intuitively a probability distribution over probability mass functions, with the number of events implied by the length of parameter $\alpha$.

A special case of Dirichlet distribution where $\alpha_1 = \alpha_2 = \cdots = \alpha_K = \alpha$ is called the symmetric Dirichlet distribution, of which the likelihood is:

$$f(\pi, \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha-1}$$

where the normalising constant $B(\alpha)$ can be computed using the gamma function as:

$$B(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(\alpha K)}$$

with $\Gamma$ denoted as the gamma function.

The Dirichlet distribution is limited in that a draw from the Dirichlet distribution is a probability mass function of the same length as of parameter vector $\alpha$. In other words, these probability mass functions can only model probability distributions over a finite set of events. The DP is proposed to handle an infinite set of events, and hence enables us to model probability distributions over infinite sample spaces.

Formally, the DP is defined by Ferguson (1973) as follows:

**Definition 2.1.** A Dirichlet process is a distribution of a random probability measure $H$ over a measurable space $(\Omega, \mathcal{B})$, such that for any partition $(A_1, A_2, \ldots, A_r)$ of $\Omega$ (i.e. $\Omega = \bigcup_{j=1}^{r} A_j$, where $\bigcup$ denotes the disjoint union and $A_j \in \mathcal{B}$), we have:

$$(G(A_1), \ldots, G(A_r)) \sim \text{Dir}(\alpha H(A_1), \ldots, \alpha H(A_r))$$
2.2. Data Modelling and Bayesian Approach

A sample $G$ drawn from a DP is a probability distribution specified by a base measure $H$ and a concentration parameter $\gamma > 0$, denoted DP($\gamma, H$).

A random probability measure $G$ over the measurable space $\Theta$, drawn from DP($\gamma, H$) can be explicitly constructed via the stick-breaking process described in Sethuraman (1994) as: $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ where $\phi_k$ are i.i.d drawn from $H$ and the weights $\pi_k(s)$ are constructed as $\pi_k = v_k \prod_{l<k}(1-v_l)$ with $v_l$ being i.i.d drawn from a Beta distribution Beta(1, $\gamma$). We can think of building $\pi_k(s)$ as breaking a stick of length 1 via the proportion $v_k(s)$, hence the name stick-breaking.

Due to its discreteness, the DP is often not applied directly to modelling the data (e.g., it is unable to model continuous data), instead it can be effectively used as a nonparametric prior on the mixture components $\theta$, which in turn, serves as the parameters within another likelihood function $F$ to generate data – a model which is known as the Dirichlet process mixture (DPM) model.

2.2.1.4 Dirichlet Process Mixture

The Dirichlet process mixture (DPM) model (Antoniak, 1974) has been embraced with great success and enthusiasm recently. The crucial advantage is its ability to naturally address the problem of model selection - a major obstacle encountered in several parametric mixture models, such as the Gaussian Mixture Models where the number of mixtures cannot be specified a priori in a principal way.

Under DPM formalism, an observation $x_n$ is generated from a two-step process: $x_n \sim F(x_n | \theta_n)$ where $\theta_n \sim G$. Using the stick-breaking representation, DPM yields an infinite mixture model representation

$$p(x | \gamma, H) = \sum_{k=1}^{\infty} \beta_k \times f(x | \phi_k)$$

where $f$ denotes the density function for $F$.

The stick-breaking representation for DPM is described as follows. Given the concentration parameter $\gamma$, the mixing proportion is drawn from $\pi_\infty \sim \text{GEM} (\gamma)$ and the collection of topics is sampled from $\phi_k \overset{iid}{\sim} H (\lambda)$. Thus, the global atom $G$ can be represented as $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$. Each local indicator (or topic assignment) is then generated as $z_i \overset{iid}{\sim} \pi$. Finally, we draw the data observation $x_i \sim F(\phi_{z_i})$. Figure 2.3
Another useful interpretation for DPM is the Chinese restaurant process (CRP) (Pitman, 1995). The concept of CRP is described as follows. Let us imagine a restaurant with an infinite number of tables and \( N \) customers \( x_1, x_2, \ldots, x_N \). Each customer chooses a table at which to sit by the following process. The first customer sits at the first table. The \( i \)-th customer sits at the \( k \)-th occupied table with the probability of \( \frac{n_k + \gamma - 1}{N + \gamma - 1} \) where \( n_k \) is the number of people sitting at table \( k \) and \( \gamma \) is a concentration parameter, or this customer sits at a new (and empty) table with the probability \( \frac{\gamma}{N + \gamma - 1} \). In topic modelling, each table represents a topic (or cluster) and each customer represent a document (or data point).

**Posterior Inference for DPM** Since the exact inference for DPM is intractable, its posterior inference is usually done using MCMC methods such as collapsed Gibbs sampling (see section 2.2.1.2). Under the Chinese restaurant process, the collapsed Gibbs inference reduces to the following conditional distribution (Neal, 2000):

\[
p(z_i = k \mid z_{-i}, x_{1:N}, \gamma) \propto \begin{cases} 
(n_k^{-i}) f_k^{-i}(x_i) & \text{if } k \text{ is previously used} \\
\gamma f_k^{-i}(x_i) & \text{if } k \text{ takes a new value}
\end{cases}
\]
where \( n_k^{-i} \) denotes the number of times cluster \( k \) has been used excluding data point \( x_i \) and \( f_{k}^{-i}(x_i) = \int_{\phi_k} p(x_i | \phi_k) p(\phi_k | \{ x_j : z_j = k, j \neq i \}) \, d\phi_k \) is the predictive likelihood of observing \( x_i \) under the \( k \)-th cluster. If \( H \) and \( F \) are conjugate distributions, then \( f_{k}^{-i}(x_i) \) can be analytically evaluated.

**Applications of DPM** Since DPM does not require a number of clusters to be specified a priori, it is suitable for applications where the number of true clusters is not known in advance. Wood et al. (2006) employed DPM for spike sorting problem. Sudderth et al. (2008) used DPM to perform visual scene analysis, in which the number of objects, parts and features in a given image are identified. Vlachos et al. (2008) used DPM for the lexical-semantic verb clustering task in natural language processing. Another use of DPM is in density estimation where modelling the density of a given set of observations (Escobar and West, 1995; Rasmussen, 1999) is the focus of interest.

### 2.2.1.5 Hierarchical Dirichlet Processes

**Representation of hierarchical Dirichlet processes (HDP)** The HDP is well-known in the discovery of multilevel significant patterns in data that has multiple groups. This model can be described as follows. Let \( J \) be the number of groups and \( \{ x_{j1}, \ldots, x_{jN_j} \} \) be \( N_j \) observations associated with group \( j \). These observations are assumed to be exchangeable within the group. Under the HDP framework, each group \( j \) is endowed with a random group-specific mixture distribution \( G_j \) which is statistically connected with other mixture distributions via another Dirichlet process (DP) sharing the same base probability measure \( G_0 \):

\[
G_j | \alpha, G_0 \overset{\text{iid}}{\sim} \text{DP} (\alpha, G_0), j = 1, \ldots, J
\]

This generative process further indicates that \( G_j (s) \) are exchangeable at the group level and conditionally independent given the base measure \( G_0 \), which is also a random probability measure distributed according to another DP:

\[
G_0 | \gamma, H \sim \text{DP} (\gamma, H)
\]  

(2.2)

It is clear from the definition of the HDP that \( G_j \)'s, \( G_0 \) and \( H \) share the same support.
2.2. Data Modelling and Bayesian Approach

Then the local atoms in group $j$ is drawn as $\theta_{ji} \overset{iid}{\sim} G_j$ and the observation is generated following $x_{ji} \sim F(\theta_{ji})$.

The stick-breaking representation of HDP for posterior inference can be summarised as follows. We draw a global mixing weight $\beta \sim \text{GEM}(\gamma)$, then generate the topics $\phi_k \overset{iid}{\sim} H(\lambda)$. The global atom $G_0$ in Equation 2.2 can be characterised as $G_0 = \sum_{k=1}^{\infty} \delta_{\phi_k} \times \beta_m$. We next sample the mixing proportion for each document $j$ such that $\pi_j \overset{iid}{\sim} \text{DP}(\alpha \beta)$. The local atom in each document is represented as $G_j = \sum_{k=1}^{\infty} \pi_{j,k} \delta_{\phi_k}$. Finally, we draw the latent assignment $z_{ji} \overset{iid}{\sim} \text{Mult}(\pi_j)$ and observation $x_{ji} \overset{iid}{\sim} F(\phi_{z_{ji}})$ accordingly. Figure 2.4 shows the graphical representation of the HDP model.

The HDP model has been embraced with great success and enthusiasm recently. However, it is designed to work with a single data source. In the HDP model, each data point $x_i$ is drawn from a single likelihood $F(x_n \mid \theta_n)$ where $\theta_n$ was sampled from the base measure $H$. To deal with multiple data sources, we need a richer generative process that can explain observed data simultaneously. To this end, we
2.2. Data Modelling and Bayesian Approach

Applications of HDP  HDP was originally applied to nonparametric text modelling, but it has also been applied in the field of natural language processing to detect how many grammar symbols exist in a particular set of sentences (Liang et al., 2007; Finkel et al., 2007). Another extension of HDP is Dynamic HDP (Ren et al., 2008) which is used for modelling time series documents. Evolutionary HDP (EvoHDP) (Zhang et al., 2010) is used for multiple correlated time-varying corpora by adding time dependencies to the adjacent epochs.

HDP was also used to construct the prior block over the number of states in the hidden Markov model (HMM) (Rabiner and Juang, 1986), resulting in the hierarchical Dirichlet processes hidden Markov model (HDP-HMM) (Teh et al., 2006), which is a Bayesian extension to HMM.

2.2.2 Models for Sequential Data

2.2.2.1 State-Space Models

State space models allow us to model an observable time series data \( \{y_t\}_{t=1}^T \) as being explained by a vector of unobserved state variables \( \{z_t\}_{t=1}^T \) which are driven by a stochastic process (Kunst, 2007).

A basic linear state space model takes the following form:

\[
y_t = Bz_t + v_t \quad (2.3)
\]
\[
z_t = Az_{t-1} + w_t \quad (2.4)
\]

where \( B \) and \( A \) are called the measurement (or emission) and the transition matrix respectively, \( v_t \) is the measurement error, \( v_t \sim N(0, \Sigma_v) \), and \( w_t \) is called the innovation, \( w_t \sim N(0, \Sigma_w) \). Equation 2.3 describes the relation between the observed time serie data \( y_t \) and the unobserved state \( z_t \) with the measurement error \( v_t \). Equation 2.4 describes the evolution of the state variable \( z_t \) driven by a stochastic process of innovations \( w_t \). Typically, the matrices \( A, B \) together with the variances \( \Sigma_v, \Sigma_w \) are not known and have to be estimated. A popular method to estimate these matrices is the Kalman filter (Kalman, 1960) which allows the construction of the
2.2. Data Modelling and Bayesian Approach

likelihood function associated with a state space model.

In this thesis, we are interested in a special case of state space model, called the hidden Markov model (HMM). In HMM, \( z_t \) takes values from a finite discrete set, and \( \Sigma_v, \Sigma_w \) are zero matrices. In other words, HMM is a state space model where states are finite discrete, and without measurement errors nor innovations. We present the HMM in the next section.

2.2.2.2 Hidden Markov Model

The hidden Markov model (HMM) can be considered as a generalisation of the mixture model where the hidden variables, which control the mixture component to be selected for each observation, are related through a Markov process rather than being independent of each other. In a Markov process, data in timestamp \( t \) are assumed to depend on data in timestamp \( t-1 \) only. This generalisation enables HMM to effectively model time series or sequential data.

Model Representation and the Three Canonical Problems  A HMM of a set of \( K \) latent states is characterised by a parameter \( \Theta = (\pi, A, B) \), where \( \pi \) is the initial probability, \( A \) is the state transition probability matrix, and \( B \) is the output emission probability matrix (a.k.a observation probability matrix). The initial probability \( \pi = [\pi_1 \pi_2 ... \pi_K] \) indicates the likelihood of starting the sequence with state \( k \) is defined as: \( \pi_k = p(s_1 = k) \), \( \forall k = 1...K \), and \( \sum_{k=1}^{K} \pi_k = 1 \). The state transition matrix \( A = \{a_{ij}\} \) where \( a_{ij} = p(s_{t+1} = j \mid s_t = i) \), \( \forall i, j = 1...K \), indicates the likelihood of the latent state at time \( t+1 \) given the state at time \( t \). We note that \( \sum_{j=1}^{K} a_{ij} = 1 \). The observation probability matrix is denoted as \( B = \{b_k(v)\} \) where \( b_k(v) = p(x_t = v \mid s_t = k) \), \( \forall k = 1...K \) and \( v = 1...V \) if we assume the observed value \( x_t \) is discrete, taking value from \( v = 1...V \), and indicates the likelihood of the observed variable given a state.

Given an output sequence \( X = \{x_t\}_{t=1}^{T} \) and a latent state sequence \( S = \{s_t\}_{t=1}^{T} \), and the model parameter \( \Theta = (\pi, A, B) \), the HMM joint probability distribution is:

\[
p(X, S \mid \Theta) = p(s_1 \mid \pi) \prod_{i=2}^{T} p(s_i \mid s_{i-1}, A) \prod_{i=2}^{T} p(x_i \mid s_i, B).
\]
In the above HMM model, some assumptions have been implemented to make the model tractable and computationally efficient. First, it is assumed that the conditional probability distribution of the hidden variable $s_t$ at time $t$, given the values of the hidden variable $s$ at all times, depends only on the value of the hidden variable $s_{t-1}$, i.e. $p(s_t \mid s_{1:t-1}) = p(s_t \mid s_{t-1})$. In other words, the values of hidden variables at time $t - 2$ and before have no influence on $s_t$ given $s_{t-1}$. This is the limited horizon assumption, or Markov assumption. Second, we assume that the probability of transition from a state $s_t$ to another state $s_{t+1}$ does not change over time. This is called the stationary process assumption. Third, the emission probability matrix is assumed to be unchanged over time – the output independence assumption. Furthermore, we note that the state space of the hidden variables in HMM is discrete while the observations themselves can either be discrete or continuous (e.g., observations from a Gaussian distribution). This is sometimes called the discrete state space assumption.

There are three basic problems for HMMs described in Rabiner (1989).

- **Problem 1**: How do we efficiently compute the probability of an output sequence $X = \{x_t\}_{t=1}^T$ given the model parameter $\Theta = (\pi, A, B)$, i.e. compute $p(X \mid \Theta)$?

- **Problem 2**: How do we choose a latent state sequence $S = \{s_t\}_{t=1}^T$ which is the most likely underlying explanation of a given output sequence $X = \{x_t\}_{t=1}^T$ and the model parameter $\Theta = (\pi, A, B)$, i.e. find $\hat{S} = \arg\max_S (S \mid X, \Theta)$?

- **Problem 3**: How do we estimate the model parameter $\Theta = (\pi, A, B)$ to maximise the likelihood $p(X \mid \Theta)$, i.e. find $\hat{\Theta} = \arg\max_{\Theta} (X \mid \Theta)$?

The first problem is solved by the Forward and Backward algorithm, while the second problem is solved by the Viterbi algorithm and Posterior decoding, and the third problem is solved by the Baum-Welch algorithm (Rabiner and Juang, 1986). Note that currently, there is no algorithm which can efficiently provide the exact solution for problem 3, but instead derives a local maximum likelihood by using the Expectation-Maximization algorithm (Dempster et al., 1977), of which the Baum-Welch algorithm is a special case. We refer the readers to (Rabiner, 1989) for more details on HMM and its related algorithms.

**Applications of HMM** HMM (Rabiner and Juang, 1986) has been widely used over the last few decades for modelling time series or sequential data. The most
2.3. Concluding Remarks

well-known applications of HMM are in speech recognition systems (Rabiner, 1989). It has been also successfully applied in protein modelling (Krogh et al., 1994), computational molecular biology (Bishop and Thompson, 1986), activity recognition and abnormality detection (Duong et al., 2005), learning and detecting activities in computer vision (Nguyen et al., 2005), video analysis and segmentation (Phung et al., 2005), etc.

2.2.2.3 Bayesian Extension to HMM

To provide the connection to the work presented in this thesis, we shall defer a discussion on the literature on recent advances in a Bayesian treatment to HMM with nonparametric priors to chapter 7.

2.3 Concluding Remarks

In this chapter, we firstly presented an overview of ubiquitous and pervasive computing, including the definitions of pervasive computing, their main characteristics and the elements of a pervasive system. Then, we briefly discussed the machine learning approaches for data modelling. Particularly, we focused on Bayesian machine learning approaches, which are the foundation of our work throughout this thesis. We presented the background of the Dirichlet process, the infinite topic modelling by the Dirichlet process mixture model and the hierarchical Dirichlet processes model. We also provided the background of the hidden Markov model and its infinite version for handling sequential data. These models are used as the building blocks for our work in chapters 5–7.
Arrhythmia Detection from Wearable ECG Devices

As previously discussed, the high-level goal of this thesis is to make sense of pervasive signals by discovering latent patterns from such kind of data. In chapter 2, we discussed how pervasive signals are useful for a wide range of applications, including human dynamics understanding (González et al., 2008; Barabási, 2005), context awareness (Schilit et al., 1994), and healthcare (Varshney, 2007). In this chapter, we aim to investigate machine learning methods for pervasive healthcare applications. Specifically, we focus on arrhythmia detection from data collected by low-cost wearable electrocardiography (ECG) devices. To this end, we propose a new approach for anomaly detection which can deal with patient-to-patient differences and the suboptimal quality of pervasive ECG signals. Experiment results show that our approach achieves higher performance in comparison to a clinical rule-based approach.

3.1 Motivation

An ECG is simply a representation of the electrical activity of the heart muscle as it changes with time (Katz, 2011). It is used to assess cardiac problems and individual health status since it can reflect the vital sign signals of the heart, one of the most important organs of human body. Usually, when an individual has a health check at a hospital, their ECG is recorded. However, these intermittent measurements
have some limitations. For example, some abnormal cardiac activities may not be reproduced at the time of measurement. In some cases, patients may experience psychological anxiety when near doctors. In these situations, continuous ECG monitoring using wearable ECG devices is preferable to a regular hospital-setting ECG as it enables a full overview of cardiac activities in daily life.

Unfortunately, long-term ECG monitoring generates a large amount of data, making the visual inspection of each heartbeat by clinicians infeasible. Ideally, if the patient has felt abnormal rhythms and can later recall the exact timing of the events, clinicians can focus on the ECG segment at that time period. However, such patient self-reporting is not always practical or reliable. Therefore, the computer-based screening of long-term ECGs is not only efficient but also a necessity.

In many computer-based ECG alert systems, a set of rules used by clinicians is translated into a decision tree. These rules have a set of fixed thresholds that are the same for all patients. In practice, such rules suffer from two shortcomings. First, these rules do not take into account patient-to-patient differences and environment variations of ECG measurements (e.g., resting vs. exercising). Second, these rules assume reliable ECG segmentation, which is often difficult to guarantee with lower signal quality in a pervasive setting.

To address this gap, we propose a new approach to detect arrhythmia that accommodates both individual differences and suboptimal signal quality. At the center of this approach is a temporal normalisation of ECG features. The normalisation is feasible because of the long-term nature of a pervasive ECG in comparison with a short-term ECG in a hospital setting. The normalisation leads to a set of robust features even with poor signal segmentation. The robustness of detection is further enhanced through the use of nonlinear machine-learning classification models. Because normalisation is done with a temporal window, the training of the model can be done online and can adapt to situational changes.

The new approach also contributes to pattern recognition applications related to ECGs in two ways. First, a set of normalised ECG features for pervasive ECGs is proposed and its pattern recognition properties are evaluated. Next, the performance of machine learning classifiers is evaluated with ECGs obtained using low-cost pervasive monitors.
3.2 Electrocardiography and Related Background

3.2.1 Electrocardiography Signals

Electrocardiography (ECG) is a medical test that detects cardiac abnormalities by measuring the electrical activity generated by the heart as it contracts. It is considered an important tool to assess the cardiac-related health status of an individual.

Usually, when an individual has a health check at a hospital, their ECG is recorded using clinical standard devices. However, it is not always possible to obtain standard measurements which are reliable. For example, some patients experience psychological anxiety when they are near doctors in a hospital environment which makes their heart rhythms different to their usual ones, and also makes the ECG less reliable. As another example, taking an ECG of athletes during a training session is impossible in a hospital environment, so wearable device is a better choice.

Currently, the most popular wearable ECG devices are Holters, which are small, light-weight, and clinically approved for continuous ECG monitoring. Unfortunately, their price is relatively high in comparison with other wearable ECG monitors. Moreover, Holter devices are not equipped with other kinds of sensors; therefore users of a Holter have to keep a diary of their daily activities to help clinicians interpret the ECG data, e.g., to discriminate between sinus tachycardia in response to exercise from supraventricular tachyarrhythmia. In the real world, it is hard for patients to remember and take notes on all their activities during their daily life.

Automatic activity recognition using accelerometers can provide an independent and ubiquitous method to record the daily activity of patients. It can be used to replace the traditional documentation diary and help physicians interpret ECG data. Therefore, some wearable ECG devices, such as Shimmer ECG devices, have integrated accelerometers to simultaneously record both ECG and acceleration signals to help infer the user’s activities. A Shimmer ECG is smaller, lighter and cheaper than a standard Holter and is equipped with accelerometers as well as a wireless connection to signals in real time. However, the Shimmer ECG has a lower signal quality than the Holter which may be problematic for physicians during an ECG examination. Fortunately, advances in signal processing and machine learning may help overcome this limitation. In the first step, signal processing techniques can be applied to reduce noise and significantly improve signal quality. Then, machine
learning techniques can be applied to accelerometer data to learn patterns and infer which activity the users are doing to provide useful information for physicians during the ECG interpretation.

Low cost pervasive ECG monitors are changing how sinus arrhythmia is diagnosed in patients with mild symptoms. However, the large amount of data generated from long-term monitoring brings new data science and analytical challenges. Although traditional rule-based detection algorithms still work on relatively short clinical-quality ECGs, they are not optimal for pervasive signals collected from wearable devices as they do not adapt to individual differences and assume the accurate identification of ECG fiducial points.

3.2.2 Arrhythmia

An arrhythmia is an abnormal heart rhythm. It occurs when the electrical signals to the heart that coordinate heartbeats are not working properly which causes the heartbeat to be too fast (tachycardia), too slow (bradycardia), too early (premature contraction), or irregular (fibrillation or flutter) (Katz, 2011). Although many arrhythmias are harmless, some particular arrhythmias which result from a weak or damaged heart can cause serious health issues and even potentially fatal symptoms. Some common types of arrhythmia are:

- **Tachycardia**: a fast heart rhythm with a rate over 100 beats per minute.
- **Bradycardia**: a slow heart rhythm with a rate below 60 beats per minute.
- **Supraventricular arrhythmias**: arrhythmias that begin in the atria.
- **Ventricular arrhythmias**: arrhythmias that begin in the ventricles.
- **Bradyarrhythmias**: slow heart rhythms that may be caused by disease in the heart’s conduction system, such as the sinoatrial (SA) node, atrioventricular (AV) node or His-Purkinje network.

The last three types of arrhythmia above can be categorised in several sub-types. We refer readers to (Walraven, 2016) for more details about different types of arrhythmia.
3.2. Electrocardiography and Related Background

3.2.3 ECG Signatures

![Diagram of the heart's electrical system. Source: http://www.uhs.nhs.uk/Media/Controlleddocuments/Patientinformation/Heartandlungs/Bradyarrhythmias-patientinformation.pdf.](http://www.uhs.nhs.uk/Media/Controlleddocuments/Patientinformation/Heartandlungs/Bradyarrhythmias-patientinformation.pdf)

Like other muscles, the cardiac muscle contracts in response to the electrical depolarisation of the muscle cells. The normal cardiac cycle begins with the spontaneous depolarisation of the SA node, an area of specialised tissue situated in the high right atrium (RA) (see Figure 3.1). A wave of electrical depolarisation then spreads through the RA and across the inter-atrial septum into the left atrium (LA). The atria are separated from the ventricles by an electrically inert fibrous ring, so in a normal heart, the only route of transmission of electrical depolarisation from atria to ventricles is through the AV node. The AV node delays the electrical signal for a short time, and then the wave of depolarisation spreads down the interventricular septum, via the bundle of His and the right and left bundle branches, into the right ventricle (RV) and left ventricle (LV) (Price, 2010). Hence with normal conduction the two ventricles contract simultaneously, which is important in maximizing cardiac efficiency.
Figure 3.2: ECG tracing with cardiac electrical activity. Source: http://cnx.org/content/m46664/latest/2023_ECG_Tracing_with_Heart_ContractionN.jpg

Figure 3.2 illustrates the different stages of cardiac electrical activities and their corresponding waveforms. The first electrical signal is the depolarisation from the right atrium, closely followed by the left atrium. It is represented by the P wave. There is then a short, physiological delay as the AV node slows the electrical depolarisation before it proceeds to the ventricles. This delay is responsible for the PR interval, a short period where no electrical signal is seen on the ECG, represented by a straight isoelectric line. Next, the depolarisation of the ventricles usually results in the largest part of the ECG signal (because of the greater muscle mass in the ventricles) known as the QRS complex. After this, there is also an electrical signal reflecting repolarisation of the myocardium. This is shown as the ST segment and the T wave. The T wave is usually considered the end of a depolarisation cycle, although a small U wave may immediately follow the T wave. The source of the U wave is unknown. Figure 3.3 shows an ECG of a normal heartbeat with its waveforms and major clinical features.
3.2. Electrocardiography and Related Background

3.2.4 Wearable ECG Platforms

Of the wearable physiological sensors, the most commonly used are ECG sensors. This type of sensor could be a small and lightweight device (e.g. Shimmer ECG device<sup>1</sup>) or in the form of clothes (e.g. VitalJacket<sup>2</sup>). The role of the ECG sensors is to measure the electrical activity of the heart, and via these cardiac physiological signals, to diagnose heart-related problems for chronic disease management (Bellos et al., 2011), arrhythmia detection and heartbeat classification (Ye et al., 2011; Fensli et al., 2005) and stress indexing (Gaggioli et al., 2013). Ye et al. (2011) analysed long-term ECG signals collected from a VitalJacket (see Figure 3.4), a wearable and real-time ECG monitor to help improve the heartbeat recognition performance. Gaggioli et al. (2013) measured ECG signals from patients to calculate HRV, and derived a stress index from the ECG. In this thesis, we collect real pervasive ECG data through the

<sup>1</sup>http://www.shimmersensing.com/shop/wireless-ecg-sensor

<sup>2</sup>http://www.vitaljacket.com/
3.3. Datasets

Shimmer ECG platform (see Figure 3.5), a small size, light weight and extensible platform for physiological signal capturing in non-invasive biomedical research (Burns et al., 2010). The Shimmer ECG can be used for long-term ambulatory monitoring up to 24 hours. This device supports recordings of ECG lead II (RA-LL) and lead III (LA-LL). A recommended position for these 4 connectors is shown in Figure 3.6.

The device can be programmed to work in an online or offline mode. In the online mode, the recorded data are sent to a PC via Bluetooth. The data can be visualised in real-time or saved on this PC in a comma-separated values file. In the offline mode, data are saved on a miniSD memory card and can be retrieved in raw or in calibrated values.

Compared to traditional Holter monitors, a Shimmer ECG has the advantage of being low cost and light in weight, supporting real-time monitoring through a Bluetooth connection.

3.3 Datasets

We used two sources of ECG data in this study. The first source was the public MIT-BIH Arrhythmia dataset\(^3\) from Physiobank. This dataset is considered a benchmark

\(^3\)http://www.physionet.org/physiobank/database/mitdb/
Figure 3.5: A Shimmer ECG platform, developed by Intel’s Digital Health Group, compared with an Australian 50 cent. Its size is \(53\text{mm} \times 32\text{mm} \times 23\text{mm}\) and its weight is 32g.

for arrhythmia detectors (Moody and Mark, 2001). The second source was ECG data recorded in a pervasive manner in our lab using the Shimmer ECG devices. While the Physiobank data are not strictly pervasive ECG, they provide comprehensive sets of arrhythmia heartbeat examples. We briefly describe these two datasets in the following sub-sections.

3.3.1 MIT-BIH Arrhythmia Dataset

The MIT-BIH Arrhythmia dataset (Goldberger et al., 2000) consists of 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. 23 recordings (100 series) were randomly chosen from 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston’s Beth Israel Hospital, and the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. The sample rate of all recordings is 360Hz, therefore each recording has about 650000 samples. Samples
Figure 3.6: The electrode positions of the Shimmer ECG device. In comparison to 12 leads clinical ECG, the Shimmer ECG has only three signal leads (LA, RA, LL) and one ground lead (RL). (source: Shimmer ECG User Guide)

are digitised in a 12 bit integer over 10mV range. Examples of ECG segments from this MIT-BIH dataset are plotted in Figure 3.7.

The MIT-BIH dataset is well-annotated by cardiologists. Each recording in the dataset was examined independently by two cardiologists and all abnormal heart beats are labelled. Any discrepancy was reviewed and resolved by consensus. Beat labels generally appear at the R-wave peak. There are approximately 109,000 beat labels in the dataset.

3.3.2 Shimmer ECG Dataset

We used two Shimmer ECG modules to collect data from two members from our lab. The subjects were all healthy males under the age of 40. The devices were configured to work in the offline mode. The sampling rate was set at 256Hz. The electrodes were placed at the recommended positions as shown in Figure 3.6. Each recording was first retrieved as a CSV file before being converted to the WFDB format, a standard format for ECG databases in Physiobank.

We collected 11 recordings from the first subject. The data collection took place in

4See: http://www.physionet.org/physiobank/database/html/mitdbdir/intro.htm for more details
various environments, from working at the office to resting at home. The activity level also covered a wide range, including sitting and working at a desk, having lunch at the office, walking around the house, or lying on a bed, to name a few. The length of each recording varied from about 30 minutes to over 8 hours.

For the second subject, we collected data in the work environment at the office for five consecutive working days. Each recording is therefore about 8 hours long.

3.4 Detection Methods and Experiment Results

3.4.1 Arrhythmia in ECG and Clinical Rule-based Detection

To identify abnormal heartbeats in ECG recordings, doctors and physicians usually follow the rule-based approach where a set of clinical rules on abnormal heartbeats is used. Typically, the following five rules are used (Rawshani, 2017):

- Rule 1 (*QRS interval*): the time of a QRS interval is longer than 0.12s.
• Rule 2 (*PR interval*): the time of a PR interval is not in the range from 0.12s to 0.22s.

• Rule 3 (*T-wave amplitude*): the amplitude of T-wave (after baseline wander removal) is negative.

• Rule 4 (*Ventricular Activation Time - VAT*): the VAT is the duration from the beginning of the QRS interval to the R wave. If it is longer than 0.03s in lead $V_1$ and 0.05s in lead $V_5$ or $V_6$, the heartbeat is abnormal.

• Rule 5 (*corrected QT interval*): the corrected QT interval ($QTc$) is calculated by formula 3.1 (Bazett’s formula). The normal limits of $QTc$ vary according to the previous RR interval, but normally it is from 0.3s to 0.45s. If the $QTc$ does not have a value in this range, the heartbeat is abnormal.

$$QTc = \frac{QT\text{ interval}}{\sqrt{\text{previous RR interval}}}$$  \hspace{1cm} (3.1)

Rule 4 assumes the availability of V-leads, which are often missing in pervasive ECG monitoring. Moreover, because its duration thresholds are small, this rule depends heavily on the segmentation accuracy, which is difficult to achieve in pervasive context. Rule 5 is used to detect prolonged QT, a condition has its own interest.

### 3.4.2 Heartbeat Detection and Segmentation

Two types of segmentations were required before the features can be extracted from a continuous ECG recording. The first step was to divide the entire ECG recording into heartbeats. This was done through identifying the R peaks in the ECG recording. We defined a heartbeat to be sitting between two consecutive R peaks as R peak identification is very reliable, even on noisy ECG signals. We used the program *wqrs* (Zong et al., 2003) in the Physiobank WFDB toolkit to find all the R peaks. To verify the program *wqrs*s’s accuracy, we took advantage of the fact that annotations in the Physiobank database were placed on the R-wave peaks. Verification with the Physiobank annotation and visual inspection confirmed the reliability of *wqrs*.

After the ECG recording had been divided into heartbeats, the second step was to identify the clinically meaningful points within each heartbeat. For this step, we used the ECG segmentation program *ecgpuwave*. This tool was validated in (Laguna et al., 1994). It identified P-waves, QRS complexes, T-waves and also their associated
onset and offset points. Note that for a pervasive ECG, the identification of the fiducial points is much more challenging and not all the identified fiducial points are reliable. Therefore, we need to select the feature sets that minimise reliance on these within-beat fiducial points.

3.4.3 Feature Sets

The main purpose of this step is to select the features that are both indicative of arrhythmia and robust to noise in pervasive monitoring. To begin, we identified the R peak to be the most reliable feature in our pervasive ECG signal, hence we focused only on the ECG features between two neighboring R peaks. In particular, RR intervals were used. The RR interval is closely related to heart rate, but is easier to measure than heart rate. At the same time, most types of arrhythmia, in particular tachycardia and bradycardia, lead to changes in RR intervals.

Within an RR interval, we considered 5 sets of candidate features:

1. **Histogram based.** For each heartbeat segment in the dataset, we built a histogram of 21 equal bins from the amplitude of samples in this segment. We then shifted the histogram so that the center was the mean of the sample amplitudes. The idea behind this histogram building is to help us examine the distribution of sample amplitudes in each segment.

2. **Percentile based.** For each heartbeat segment in the dataset, we calculated the percentiles of the sample amplitude in this segment. The percentiles varied from 0% to 100% with the 5% steps in between. This was another method to help us examine the distribution of sample amplitudes in each segment.

3. **Percentile of derivative.** This feature set was created in a similar way to the percentile-based one. However, for this method, we did not use the sample amplitudes but its derivative to capture the changes in the heart beat. In more detail, we applied a mean filter on the raw data, then we calculated the derivative by the equation: \( y(t) = x(t+1) - x(t) \)

4. **Percentile of derivative with Savitzky-Golay filter.** This is the same as the percentile of derivative method, except that we used the Savitzky-Golay filter in place of the mean filter to smooth the data before calculating the derivative.
5. **RR-intervals and ST duration.** The previous four feature sets did not assume the identification of fiducial points between two R peaks. This feature set added in a feature that relies on the correct identification of S and T. However, in general, S and T are still easier to identify and the feature set has the advantage of simplicity.

### 3.4.4 Comparison of Feature Sets

The purpose of this experiment step was to find robust feature sets for pervasive ECGs from the five feature sets presented in Section 3.4.3. To compare the feature sets, we used the labelled ECG recording 100 from the MIT-BIH Arrhythmia dataset.

When an arrhythmia happens, it may affect both the heartbeat preceding the R peak and the one following the peak. In extreme cases, an anomalous heartbeat may have longer lasting effects and change the shape of the other near heartbeats. Thus, in this experiment, we also compared the number of heartbeats to be included in a unit of observation. That is, should an arrhythmia occurrence include only one heartbeat, or two, or even four heartbeats? To answer the question, we created three divisions of continuous ECGs using Algorithm 3. We called these datasets $A_1, A_2, A_4$ respectively. Dataset $A_1$ contains segments of one heartbeat length. Dataset $A_2$ contains segments of two heartbeat length, and in the same way for the dataset $A_4$.

**Initialise:** $n$ be the total heartbeats segmented from data

**for** $i = 1, 2, 4$ **do**

| Divide data in $i$-heart-beat segment $s_i^j$ with $j = 1, 2, 3, \ldots, \lfloor n/i \rfloor$
| Build dataset $A_i = \{s_i^j\}$

**end**

**Algorithm 3:** Dataset creation for observation units of varying length.

The different combinations of settings were compared in relation to classification accuracy using the WEKA’s decision tree package (Hall et al., 2009). F1 score ($\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$) was used to measure classification accuracy and the mean of the 10-fold cross validation was reported. Table 3.1 shows the classification accuracy for different feature selection methods and different observation unit lengths. The results show that the feature set of the RR-interval with ST duration achieves the optimal classification accuracy. Also it is valid to set each observation unit to be one heartbeat.
Table 3.1: Experiment results using the MIT-BIH Arrhythmia dataset with three sub-datasets created using Algorithm 3, with different feature extraction methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Histogram</th>
<th>Percentile</th>
<th>Percentile of derivative (PoD)</th>
<th>PoD with Savitzky-Golay filter</th>
<th>RR intervals and ST duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>0.8</td>
<td>0.3617</td>
<td>0.4833</td>
<td>0.8923</td>
<td>0.9275</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0.0857</td>
<td>0.1091</td>
<td>0.127</td>
<td>0.4941</td>
<td>0.8776</td>
</tr>
<tr>
<td>$A_4$</td>
<td>0.2295</td>
<td>NaN</td>
<td>0.3019</td>
<td>0.5</td>
<td>0.8571</td>
</tr>
</tbody>
</table>

3.4.5 Normalisation

In pervasive monitoring, one challenge is the variations among different subjects and data collection environments. For example, heartbeats of older people tend to be longer. Also, the heart beats slower at resting state compared to the time when the person is walking or running. Such overt or hidden variations make automated arrhythmia detection in pervasive monitoring particularly difficult. Clearly arrhythmia detection rules, which are based on a set of fixed thresholds, are not designed to handle such variations.

In simple settings with overt variations, one could define a small number of modes and model each mode explicitly. However in practice, the variations are often too complex to be modeled exhaustively. For real individualised and adaptive ECG features, a more natural approach is to smooth the ECG signal before feeding it to an identification algorithm. Smoothing is common in many pattern recognition applications. It also matches the requirements of pervasive ECG monitoring.

Our smoothing method consists of two elements. First, we normalised each ECG feature for a heartbeat against the distribution of the feature for the individual at the period of time. Second, robust statistics were used in the normalisation process.

More specifically, let $A(i)$ be the original ECG feature at the $i$-th heartbeat and let $W$ be a smoothing window size. The normalised feature $B(i)$ is calculated as:

$$B(i) = \frac{A(i) - \text{MED}_W(i)}{\text{MAD}_W(i)}$$
where $\text{MED}_W(i)$ and $\text{MAD}_W(i)$ are the median and median absolute deviation (MAD) of \{A(k) : i \pm 1/2 \leq k \}. In our experiments, the smoothing window size $W$ was set to cover 30 minutes. This is because most non-chronic arrhythmias, for example ventricular fibrillation, only last for several minutes.

### 3.4.6 Comparison of Normalisation-based and Rule-based Methods

As mentioned earlier, normalisation-based and rule-based arrhythmia detection methods (section 3.4.5 and 3.4.1) were compared on two following datasets.

**MIT-BIH Arrhythmia dataset** The dataset contains 48 recordings, but unfortunately not every recording has a sufficient number of abnormal heartbeats. In our experiment, we excluded those recordings with 50 or less abnormal heartbeats, which results in 34 ECG recordings as listed in Table 3.2.

**Shimmer ECG dataset with the inclusion of abnormal heartbeats** Because we collected data from members of our lab who were in good health, we did not have abnormal data in our dataset. To create a dataset which have both normal and abnormal heartbeats, we had to insert abnormal heartbeats from another source. We extracted abnormal heartbeats from the MIT-BIH Arrhythmia dataset and then inserted them into a random 2-hour-long ECG segment collected using the Shimmer devices. Therefore, for each ECG recording in the MIT-BIH dataset, there was a collection of inserted heartbeats hence the dataset consisted of over 8000 normal heartbeats plus all the abnormal heartbeats from the MIT-BIH ECG recording.

**Classification results** For each collection of normal and abnormal heartbeats, we applied 10-fold cross validation to measure the classification accuracy. For each training-testing data partition, a decision tree model was constructed using the training data and its accuracy was measured using the testing set. Both the decision tree and the cross validation were implemented in Weka. Table 3.2 shows the comparison results using both the MIT-BIH data and the inserted Shimmer data. For almost all the ECG recordings, the decision tree using normalised features achieved higher classification accuracy.
### Table 3.2: Experiment results with the MIT-BIH Arrhythmia dataset and our dataset recorded from Shimmer devices.

<table>
<thead>
<tr>
<th>Record</th>
<th>No. of heart beats</th>
<th>No. of abnormal heart beats</th>
<th>F1 score</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MIT-BIH dataset</td>
<td>Shimmer dataset</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rule-based method</td>
<td>Decision tree based method</td>
<td>Rule-based method</td>
</tr>
<tr>
<td>102</td>
<td>2192</td>
<td>60</td>
<td>0.43</td>
<td>0.47</td>
<td>0.07</td>
</tr>
<tr>
<td>104</td>
<td>2311</td>
<td>689</td>
<td>0.58</td>
<td>0.7</td>
<td>0.33</td>
</tr>
<tr>
<td>106</td>
<td>2098</td>
<td>518</td>
<td>0.53</td>
<td>0.99</td>
<td>0.39</td>
</tr>
<tr>
<td>107</td>
<td>2140</td>
<td>56</td>
<td>0.44</td>
<td>0.88</td>
<td>0.06</td>
</tr>
<tr>
<td>109</td>
<td>2535</td>
<td>2528</td>
<td>0.71</td>
<td>0.71</td>
<td>0.77</td>
</tr>
<tr>
<td>111</td>
<td>2133</td>
<td>2115</td>
<td>0.89</td>
<td>0.89</td>
<td>0.74</td>
</tr>
<tr>
<td>114</td>
<td>1890</td>
<td>58</td>
<td>0.09</td>
<td>0.70</td>
<td>0.06</td>
</tr>
<tr>
<td>116</td>
<td>2421</td>
<td>109</td>
<td>0.7</td>
<td>0.99</td>
<td>0.13</td>
</tr>
<tr>
<td>118</td>
<td>2301</td>
<td>2273</td>
<td>0.69</td>
<td>0.69</td>
<td>0.75</td>
</tr>
<tr>
<td>119</td>
<td>2094</td>
<td>447</td>
<td>0.99</td>
<td>0.99</td>
<td>0.37</td>
</tr>
<tr>
<td>124</td>
<td>1634</td>
<td>1617</td>
<td>0.84</td>
<td>0.85</td>
<td>0.67</td>
</tr>
<tr>
<td>200</td>
<td>2792</td>
<td>850</td>
<td>0.69</td>
<td>0.85</td>
<td>0.51</td>
</tr>
<tr>
<td>201</td>
<td>2039</td>
<td>241</td>
<td>0.56</td>
<td>0.7</td>
<td>0.21</td>
</tr>
<tr>
<td>202</td>
<td>2146</td>
<td>54</td>
<td>0.11</td>
<td>0.76</td>
<td>0.03</td>
</tr>
<tr>
<td>203</td>
<td>3108</td>
<td>412</td>
<td>0.7</td>
<td>0.92</td>
<td>0.31</td>
</tr>
<tr>
<td>205</td>
<td>2672</td>
<td>80</td>
<td>0.31</td>
<td>0.82</td>
<td>0.08</td>
</tr>
</tbody>
</table>
This empirical evaluation shows that our proposed approach has higher accuracy than the widely accepted diagnostic rules derived from clinical-quality ECG signals. An explanation for this result is that as low-cost pervasive devices produce signals of lower quality, it is not wise to blindly apply traditional methods established for clinical-quality devices. Our work also shows that the proper application of even simple pattern recognition techniques works better than traditional clinical rules.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>207</td>
<td>2385</td>
<td>2166</td>
<td>0.76</td>
<td>0.956</td>
<td>0.56</td>
<td>0.96</td>
</tr>
<tr>
<td>208</td>
<td>3040</td>
<td>1353</td>
<td>0.87</td>
<td>0.91</td>
<td>0.62</td>
<td>0.95</td>
</tr>
<tr>
<td>209</td>
<td>3052</td>
<td>384</td>
<td>0.23</td>
<td>0.89</td>
<td>0.08</td>
<td>0.91</td>
</tr>
<tr>
<td>210</td>
<td>2685</td>
<td>166</td>
<td>0.46</td>
<td>0.75</td>
<td>0.16</td>
<td>0.87</td>
</tr>
<tr>
<td>213</td>
<td>3294</td>
<td>605</td>
<td>0.59</td>
<td>0.581</td>
<td>0.317</td>
<td>0.936</td>
</tr>
<tr>
<td>214</td>
<td>2297</td>
<td>2255</td>
<td>0.49</td>
<td>0.826</td>
<td>0.360</td>
<td>0.963</td>
</tr>
<tr>
<td>215</td>
<td>3400</td>
<td>165</td>
<td>0.45</td>
<td>0.87</td>
<td>0.171</td>
<td>0.89</td>
</tr>
<tr>
<td>217</td>
<td>2280</td>
<td>415</td>
<td>0.8</td>
<td>0.93</td>
<td>0.29</td>
<td>0.94</td>
</tr>
<tr>
<td>219</td>
<td>2312</td>
<td>72</td>
<td>0.24</td>
<td>0.88</td>
<td>0.04</td>
<td>0.77</td>
</tr>
<tr>
<td>220</td>
<td>2068</td>
<td>94</td>
<td>0.02</td>
<td>0.94</td>
<td>0.02</td>
<td>0.9</td>
</tr>
<tr>
<td>221</td>
<td>2462</td>
<td>394</td>
<td>0.97</td>
<td>0.99</td>
<td>0.34</td>
<td>0.97</td>
</tr>
<tr>
<td>222</td>
<td>2634</td>
<td>414</td>
<td>0.3</td>
<td>0.76</td>
<td>0.14</td>
<td>0.76</td>
</tr>
<tr>
<td>223</td>
<td>2643</td>
<td>549</td>
<td>0.53</td>
<td>0.84</td>
<td>0.37</td>
<td>0.85</td>
</tr>
<tr>
<td>228</td>
<td>2141</td>
<td>365</td>
<td>0.65</td>
<td>0.92</td>
<td>0.3</td>
<td>0.94</td>
</tr>
<tr>
<td>231</td>
<td>2011</td>
<td>1256</td>
<td>0.6</td>
<td>0.71</td>
<td>0.46</td>
<td>0.98</td>
</tr>
<tr>
<td>232</td>
<td>1816</td>
<td>1774</td>
<td>0.75</td>
<td>0.77</td>
<td>0.67</td>
<td>0.98</td>
</tr>
<tr>
<td>233</td>
<td>3152</td>
<td>844</td>
<td>0.86</td>
<td>0.97</td>
<td>0.52</td>
<td>0.94</td>
</tr>
<tr>
<td>234</td>
<td>2764</td>
<td>52</td>
<td>0.38</td>
<td>0.04</td>
<td>0.06</td>
<td>0.82</td>
</tr>
</tbody>
</table>
Certainly, our method by no means is the optimal solution, but we hope it may prompt pattern recognition researchers to participate in the exciting opportunities which exist in pervasive health monitoring. The ECG feature normalisation we propose has the potential to evolve into a new set of metrics for health information exchange. Further research on feature normalisation is needed in the context of pervasive data collection.

3.5 Concluding Remarks

We proposed a normalisation based approach for arrhythmia detection that effectively addresses two major challenges in pervasive long-term ECG monitoring. Empirical evaluation showed that the proposed approach has higher accuracy than the widely accepted diagnostic rules for clinical-quality ECG signals.

Low-cost pervasive devices are gaining wider adoption, serving a large population of relatively healthy individuals interested in the early detection of potential health risks. As these devices produce lower-quality signals, we should not blindly apply traditional methods for clinical-quality devices. Our work shows that the proper application of even simple data science techniques has advantages over traditional clinical rules. Although our method is by no means the optimal solution, it may motivate more data scientists to research the challenges in pervasive health monitoring.

The ECG feature normalisation we proposed has the potential to evolve into a new set of metrics for health information exchange. Further research on feature normalisation is needed in the context of pervasive data collection. Many pervasive ECG devices collect other health signals, for example, the Shimmer ECG device also has in-built accelerometers to collect accelerometer data. Many of the participants of our study also wore other devices such as Shimmer GSR (Galvanic Skin Response), Fitbit (fitbit.com) or sociometric devices (http://hd.media.mit.edu/badges). Hence, one further research direction is how to combine different signals to enhance arrhythmia detection. For example, accelerometer data may help identify a period when a subject is in rapid movement and consequently reduce the number of false positives.
In chapter 3, we presented a machine learning approach for arrhythmia detection on pervasive physiological signals that are captured in daily life using wearable SHIMMER ECG devices. We now move on to another application of machine learning for location and motion pattern discovery from pervasive signals.

To date, a GPS has been the most popular means to provide location information. However, GPS sensors drain the battery quickly and have poor accuracy indoors. Thus, it is unsuitable for some classes of applications which require long-term data collection and large indoor coverage, such as understanding human dynamics. In such cases, WiFi data are more preferable as WiFi sensors consume less power and have a high indoor connectivity.

In this chapter, we present a method to the problem of understanding human dynamics via location and motion using WiFi data. Firstly, we cluster the WiFi access points in close proximity into groups where each group is a location. Then, from the discovered location trajectories, we discover individual mobility patterns of mobile phone users. Finally, we analyse the entropy of the location distributions which shows significant correlation with the age and occupations of users. This entropy information reflects the human dynamics of participants.
4.1 Motivation

Location information plays a key role in understanding human dynamics (Adams et al., 2006; González et al., 2008; Phung et al., 2009a; Ye et al., 2009; Dong et al., 2011). It is also essential to mobile applications including context-aware applications (Schilit et al., 1994; Banerjee et al., 2002), mobile marketing and advertising (Rao and Minakakis, 2003), or healthcare (Belik et al., 2009; Moturu et al., 2011). Up to now, a GPS has been the most popular means to inform of locations. GPS data include the longitude and latitude of the device at the scanning time and can be easily clustered using unsupervised learning techniques. However, GPS sensors drain the battery quickly. Moreover, they need a long start-up period before they can capture an accurate position. They also need a clear sky view to perceive the signal from satellites; thus making them unsuitable for indoor applications. An experiment conducted by Kim et al. (2009) showed that, on average, GPS coverage is available only 5–30% of the time during a user’s typical day. Therefore, while providing relatively accurate information on position, a GPS is not suitable for tracking location to understand human dynamics, where long-term data collection requires low power consumption and large indoor coverage.

In contrast to GPS sensors, WiFi devices consume much less power. They also have a high indoor connectivity as a result of the rapidly increasing number of public and freely available WiFi access points – from 1.3 million globally to 5.8 million in 2015\(^1\). As a part of the IEEE 802.11 wireless LAN standard, a public WiFi access point broadcasts its identity information (including its MAC address which is unique for each device) frequently so that any smartphone can easily detect them. The visibility of a WiFi access point can be established by simple scans, which is much faster than establishing a GPS connection. This makes WiFi more suitable for understanding human dynamics via location and motion.

Typical WiFi-based location discovery relies on supervised learning techniques that require labelled data (e.g., geo-localised information of access points) to train a classifier (Yang et al., 2008; Kelly et al., 2009; Bell et al., 2010). However, such labelled data are often difficult to obtain in real-world applications, and the training process needs to be repeated when data from new locations arise. This limits

the scope of location applications based on WiFi access points. Using GPS to calibrate locations associated with WiFi signatures is another approach to provide labels, popularly used in commercial products such as Google Maps. However, this approach is centralised, targets a large population, requires large infrastructure and provides rather coarse spatial resolutions. Hence, it is not suitable for understanding individualised human dynamics over a long period of time.

In this chapter, we use only WiFi access point information captured by smartphones to discover the individuals’ significant locations and their daily routines. Unlike previous studies, we use unsupervised techniques to discover the significant locations. By ‘significant location’, we mean a location where a user comes to this location in an adequate number of times, or this user stays at the location in an adequate duration. We construct the adjacent matrix of access points in which each element is the number of times the corresponding pair of access points appear together in the same scan. We use the Affinity Propagation (AP) algorithm (Frey and Dueck, 2007) to cluster the access points on this adjacent matrix. The advantage of the AP algorithm is that it can discover the number of clusters automatically. We consider each cluster of access points as a location. For each scan, we find the cluster that covers the largest number of access points in such scan, then we assign this cluster as the location of the scan. Furthermore, by analysing the sequence of significant visited places, we can also discover the daily routines of this user.

We demonstrate our approach on the Nokia Mobile Data Challenge (MDC) dataset\(^2\) (Laurila et al., 2012) – a real-world dataset collected using the smartphones of 191 persons over 18 months. We cluster the data from each user separately and evaluate the clustering results against the visited locations of user, which are inferred from the exact coordinates obtained from other sources such as GPS. The performance of the algorithm is evaluated using standard metrics including \textit{F-measure}, \textit{Rand-index}, \textit{normalised mutual information} (NMI) and \textit{purity}. The experiment results show high performance for most of the users. The coverage time is also high compared to the approach using multimodal data on the same dataset. We further use the extracted locations to cluster the days according to the distributions of the locations. From the day clusters, we find some typical daily patterns and also the outlying days when the user’s location pattern is different. We also demonstrate the human dynamics of users by computing the entropy over the distribution of locations and show a

\(^2\)http://www.idiap.ch/project/mdc/
significant correlation of entropy with age and working status.

Our main contributions in this chapter are as follows. First, we only use data from WiFi access points to discover the locations and daily routines of a user. Second, we use the AP algorithm to accomplish our goal without the need to explicitly specify beforehand the number of clusters as the AP algorithm can infer such information automatically. Third, we demonstrate our approach on the large MDC dataset, which contains more than 30 million real-world WiFi scans.

4.2 Approach

As discussed in Section 4.1, typical existing approaches to significant location discovery use either exact location information provided by GPS sensors or fingerprint data to train the classifiers. Our approach, in contrast, uses WiFi data only without any information on the location of the devices or the access points. Assuming that the WiFi access points’ locations are fixed, access points that appear together in the same scan are close to each other. Considering each access point as a location indicator, we cluster them into groups and consider each group to be a location. An example of the access points and the user’s location is illustrated in Figure 4.1. In this example, the four access points from 1 to 4 can be clustered into one group. Any position in the range of at least one access point of this group can be seen as belonging to the same location. Statistically, the possibility of two access points belonging to the same cluster is proportional to the number of times they co-occur in the same scans. Thus, we construct a square matrix $S$ where each dimension is the number of access points and each element $s_{ij}$ is the number of times the access points $x_i$ and $x_j$ occur together. We consider this co-occurrence matrix as the similarity matrix between the access points.

Given the similarity matrix $S$ described above, we now cluster the access points into groups. Because we do not know the number of clusters in advance and this is difficult to identify without any training data, we choose the AP algorithm (Frey and Dueck, 2007) as it can deal with non-metric space and produce representative examples (a.k.a, exemplars) without knowing the cluster numbers in advance.

The AP algorithm discovers the latent sub-groups by exchanging the local messages. For each access point $x_i$, an exemplar node $c_i$ is created and treated as a hidden
Figure 4.1: Example of adjacent WiFi access points and user’s location. The red circles indicate the coverage range of the corresponding access point. At position 1, the device can detect the AP2, AP3 and AP4. At position 2, the device can detect the AP1 and AP2. In contrast, AP5 is the only access point that can be seen at position 3 and 4.

variable, subject to be inferred. A factor graph is constructed and function potentials are designed to encode the similarity measure between the access points as well as to enforce specific constraints to make all $c_i$ a valid configuration, as shown in Figure 4.2 (see Frey and Dueck (2007) for further details). This representation is now treated strictly as an undirected probabilistic graphical model to be inferred (Koller and Friedman, 2009). Performing max-sum message-passing to minimise the energy function in this factor graph results in the solution for the AP algorithm. The energy function equals the sum of all potentials. In the sum, there are two main types of messages being passed: ‘responsibility’ and ‘availability’. The ‘responsibility’ $r(i, k)$ sent from a point $i$ to its candidate exemplar $k$ represents how well $i$ “trust” $k$ as its exemplar. It is updated by the following equation, which allows all the candidate
exemplars to compete for ownership of data point $i$.

$$r(i, k) \leftarrow s(i, k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i, k') + s(i, k')\} \quad (4.1)$$

The ‘availability’ $a(i, k)$ sent from a candidate exemplar $k$ to a point $i$ represents the goodness it would have for $i$ to choose $k$ as its exemplar. It is updated by the following equations:

$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i' \text{ s.t. } i' \notin \{i, k\}} \max \{0, r(i', k)\} \right\} \quad (4.2)$$

$$a(k, k) \leftarrow \sum_{i' \neq k} \max(0, r(i', k)) \quad (4.3)$$

At any point during affinity propagation, we can combine availabilities and responsibilities to obtain the exemplars.

$$c_i = \arg\max_k (a(i, k) + r(i, k)) \quad (4.4)$$

where $c_i = k$ indicates that the point $k$ is the exemplar for the point $i$.

\[\text{Figure 4.2:} \quad \text{Factor graph for message passing in the AP algorithm (courtesy of Frey and Dueck, 2007).}\]

The pseudocode of the AP algorithms is shown in Algorithm 4.

In our application, the measure for clustering the access points is the number of
4.3. Experiments

Input: similarity matrix $S_{N \times N}$, where an element $s_{ij}$ represents the similarity of data points $i$ and $j$

Output: exemplar indicators $\{c_1, \ldots, c_N\}$

Initialize: set ‘availability’ matrix $A$ to zeros

while termination criteria are not met do
  Update the ‘responsibility’ matrix $R$, using rule 4.1, given the ‘availability’ matrix $A$.
  Update the ‘availability’ matrix $A$, using rules 4.2 and 4.3, given the ‘responsibility’ matrix $R$.
  Combine availabilities and responsibilities, using rule 4.4, to identify exemplars.
end

Algorithm 4: Affinity Propagation (AP) algorithm (Frey and Dueck (2007))

Times they coexist in the same scans (representing how likely they belong to the same location cluster). Groups are gradually formed based on these messages and the most dominant access point will emerge as the exemplar in each cluster.

From the clusters obtained by the AP algorithm, we assign the location label of each scan by finding the cluster that includes the largest number of access points to appear in that scan. This completes the location discovery through WiFi access points. Once we obtain the location information of mobile phone users, we can visualise the individuals’ daily routine through changes in locations throughout the day. The entropy reflected in the location/motion trajectories can be used to understand human dynamics. More details of this analysis can be found in section 4.3.

4.3 Experiments

We demonstrate our approach on the WiFi data obtained from the MDC dataset. We discover the significant locations of each user separately as the ground truth information is inconsistent between the users. We then show the correlation between the inferred locations with the user’s visited locations included in the dataset.
4.3. Experiments

4.3.1 The Nokia MDC Dataset

The Nokia Mobile Data Challenge (MDC) dataset was one of the outcomes of the Lausanne Data Collection Campaign (LDCC) which ran from 2009 to 2011 by the Nokia Research Centre at Lausanne, the IDIAP Research Institute and the École Polytechnique Fédérale de Lausanne (EPFL). This campaign used sensors embedded in mobile phones to collect continuous data pertaining to the behavior of individuals and social networks. More specifically, the data were collected using Nokia N95 smartphones of 191 users over a period of 578 days (from September 1st, 2009 to April 1st, 2011). A rich and wide range of data were collected including spatial data (e.g., GPS longitudes and latitudes, cellular IDs, and geo-localised WiFi access points), social interaction data (e.g., call and SMS logs, Bluetooth connectivity), as well as phone usage collected from the log files (Kiukkonen et al., 2010).

In this chapter, we used only WiFi data to extract the significant locations of users, thus we introduce this data in the following. Details on other types of data included in the dataset can be found in (Laurila et al., 2012). Every 2 minutes, a pre-installed application in the phone scans for surrounding WiFi access points. Each scan might detect one or more access points. The MAC addresses of detected access points were hashed and then recorded together with the timestamps. There is no information on the real MAC addresses nor the exact locations of the access points in the dataset. In total, there are over 31 million WiFi scans including more than 556,000 unique access points in the dataset.

The dataset also includes information about locations where users stay for the duration of at least 10 or 20 minutes (there are two different tables including visits that are longer than 10 or 20 minutes). We used this information to extract the ground truth location of each WiFi scan. We note that in (Montoliu and Gatica-Perez, 2010) the significant locations were extracted using clustering methods on the exact positions (longitude and latitude) of users which were mostly obtained from GPS data. However, due to privacy, the MDC dataset does not include the exact coordinates obtained from GPS. Our method, in contrast, does not require any information about the exact positions including the positions of the access points. For the convenience of evaluation, we selected at most 20 places for each user having the top number of WiFi scans and ignored the scans that do not have ground truth information. We also eliminated the access points that are visible less than 10 times by each user.
A natural but important property of location data is its unbalance. For each user, there is typically a place that was visited dominantly and thus, the number of scans belonging to that place is much larger than other places. This property is illustrated in Figure 4.3 with six examples of users who have the largest number of WiFi scans. This property is trivial as home and/or work are one or two places where everyone spends most of their time.

Figure 4.3: Number of scans corresponding to the ground truth places of six users which have the largest number of scans.

4.3.2 Significant Location Discovery from WiFi Data

For each user, we cluster the access points in his/her WiFi data into groups and each group of access points is considered a significant location. Note that we already eliminated the access points that appear less than 10 times as they do not contribute to the significant location discovery. The similarity of each pair of access points
is the number of times they coexist in the scan. We fed this similarity matrix to
the AP algorithm (Frey and Dueck, 2007), which discovered the number of clusters
automatically. The location of each scan was assigned to the group of access points
that best matches the set of access points appearing in that scan. We evaluated the
performance of our approach using standard metrics including $F$-measure, Rand-index,
normalised mutual information (NMI) and purity. We visualise the performance in
Figure 4.4, in which the metrics are sorted in descending order. It is clear that we
achieve good performance for most of the users. For example, about 110 users have
an $F$-measure greater than 0.9, about 120 users have a Rand-index greater than 0.9,
and about 140 users have purity greater than 0.95. The performance metrics shown
in Figure 4.4 prove that our method provides consistent results to the method using
multiple data sources including GPS in (Montoliu and Gatica-Perez, 2010).

![Graphs of performance metrics](a) F-measure (b) Rand-index (c) Normalised mutual information (d) Purity

**Figure 4.4:** Performance metrics of all users. The metrics are sorted according to
descending order.

We have a deeper look at three examples corresponding to the three users that
have an F-measure equal or close to the max, min and average F-measure. As
4.3. Experiments

Figure 4.5: The normalised confusion matrix between location ground truth and discovered locations. Each row is a ground truth location and each column is a discovered location. Each element is the percentage of data points belonging to the corresponding ground truth location which are assigned to the corresponding discovered location.

aforementioned, the data are unbalanced, thus the above clustering performance metrics might not fully show the importance of the discovered locations. In other words, we do care not only the locations that are visited most regularly, but also the places that appear less frequently. We thus look at the confusion matrix between the ground truth locations and the discovered locations. As the data are unbalanced, we normalise the confusion row of each ground truth location into 1 and each element represents the percentage of the data points belonging to such ground truth locations assigned to the discovered locations. We show the normalised confusion matrices of three users in Figure 4.5. The first example is user 6033 shown in Figure 4.5a. This user includes 60,024 valid WiFi scans and has a min F-measure (0.5573). We can see from this user that some ground truth places are fragmented into different clusters. This is the reason why the F-measure is so low. Most of the places, however, match well into some separate clusters. The second example is user 6075 which has a max F-measure (0.9914) as shown in Figure 4.5b. For this user, only 4 out of 20 ground truth places are split into smaller clusters. The remaining 16 places match perfectly each with a cluster. The last example is user 6187 who has an F-measure closest to the average F-measure (0.9297) as shown in Figure 4.5c. This user has 14 ground truth places and most of the places are exactly discovered except the first three places which are fragmented. Overall, the typical trend is that some places are fragmented into different clusters and most places are correctly discovered. The mixing of ground truth places in a cluster is rarely seen.
Another issue of using WiFi data for significant location discovery is coverage time. Adapting the idea from Montoliu and Gatica-Perez (2010), we generated the visit time spans that include the same extracted location. We computed the percentage of the coverage time using WiFi only over the coverage time using multimodal data as in the work of Montoliu and Gatica-Perez (2010) as shown in Figure 4.6. We sorted the users in descending order of coverage percentage. The average percentage is 60.88% and the standard deviation is 16.58%. This statistic shows a high coverage time using WiFi only without any information on the exact location of users.

![Figure 4.6: Percentage of time coverage using WiFi only versus multimodal data in (Montoliu and Gatica-Perez, 2010).](image)

### 4.3.3 Daily Routine from Location Trajectory

Using the location trajectory extracted in the above section, we can discover the typical daily patterns of users as well as the outlying days. For this, we consider the distribution of each day of each user as a multinomial distribution over the discovered locations. The distance between the days is computed by the Jensen-Shannon divergence as:

$$JS(p, q) = \frac{1}{2} (KL(p, m) + KL(q, m))$$

where $m = \frac{1}{2} (p + q)$ and $KL(p, m) = \sum_i p_i \log \frac{p_i}{m_i}$ is the Kullback-Leibler divergence between the $p$ and $m$ distributions. The similarity between two days $p$ and $q$ is computed by:

$$\text{Similarity}(p, q) = -\exp (JS(p, q))$$
4.3. Experiments

Figure 4.7: Three routines of user 6075 who is a full-time working participant. The $x$ axis is 24 hours during the day and the $y$ axis is the location labels extracted from WiFi data.

Figure 4.8: Three routines of user 5945 who is a part-time working participant. The $x$ axis is 24 hours during the day and the $y$ axis is the location labels extracted from WiFi data.

Figure 4.9: Three routines of user 5955 who is a housewife. The $x$ axis is 24 hours during the day and the $y$ axis is the location labels extracted from WiFi data.
The similarity matrix of the days of a user is then fed to the AP algorithm to obtain the clusters of days. The clusters that contain many days can be seen as a daily routine pattern while the days that stay alone in a separate cluster can be seen as an outlying day. We show some examples of the routines and outlying days of some users. For the clusters of days that include many days, we add the locations of these days into a vector of 24 elements, each being an hour in the day. We show three interesting routines of user 6075 in Figure 4.7. The first routine in Figure 4.7a shows that he mostly stays at home, while the second routine in Figure 4.7b shows that he stays at home at night time and in the early morning and he goes to the office during working hours. Note that he works full time and these are the two typical routines for full-time working participants. The third routine in Figure 4.7c shows that he goes somewhere else at night.

The second example is the daily routines of user 5945 who is a part-time working participant, as shown in Figure 4.8. The first routine in Figure 4.8a is similar to that of user 6075 as both stay home all day. On working days, Figure 4.8b and Figure 4.8c show that the routine of user 5945 is arbitrary compared to that of user 6095. He also goes to several places during working hours. This is in line with the fact that he works part time, so he does not stay in an office during regular working hours.

Another example of the daily routine of user 5955 is visualised in Figure 4.9. In contrast to the full-time working participants who have regular behaviour patterns in the office during working hours, this participant who is a housewife does not have regular patterns. Instead, the places she visits are quite random and irregular in time.

4.3.4 Understanding Human Dynamics from Locations

The distribution over the locations of each user reflects his/her dynamics or mobility. We used information entropy to measure this distribution. In brief, entropy is a measure of information in a probability distribution, with higher entropy corresponding to less information (i.e., more uncertainty). Entropy is zero when one outcome is certain to occur and entropy is maximized when the probability density function corresponds to the one with maximum uncertainty – the uniform distribution. In case of a distribution over locations, high entropy means the user tends to stay at more locations and spend the same amount of time at each location, while low
entropy means the users tends to stay at some certain locations and spend much time at these locations. Thus, we computed the entropy of all users using their location distributions which is visualised in Figure 4.10. From the demographics table included in the database, we can obtain the age group and working status of the users. We drew some statistical results from this information. Firstly, we computed the average entropy of users according to their age groups as shown in Figure 4.11. There are five age groups shown in this figure with the peak being the age group between 22 and 27 year olds. The average entropy decreases when age increases. This is reasonable as people between 22 and 27 years of age are the most active group and they tend to move a lot in their daily life.

Secondly, we computed the average entropy of users based on their working status. There are eight different working status labels, including two statuses that do not include any user. We visualise the average entropy by the working status labels in Figure 4.12. As shown in this figure, there is not much difference between the participants who work and study, but the housewives have a higher average entropy. This might be because they have more leisure time and do not have regular routines.

![Entropy of all users computed from location distribution](image)

**Figure 4.10:** Entropy of all users computed from distribution over locations. The users are sorted in the descending order of entropy.

### 4.4 Concluding Remarks

In this chapter, we presented an approach for the discovery of significant location using WiFi access points. The access points were clustered into significant locations.
4.4. Concluding Remarks

![The average entropy by age group](image1)

**Figure 4.11:** The average entropy by age group. The age groups are divided as follows: group 2: between 16 and 21 year old, group 3: between 22 and 27 year old, group 4: between 28 and 33 year old, group 5: between 33 and 38 year old, group 6: between 39 and 44 year old.

![The average entropy by working status](image2)

**Figure 4.12:** The average entropy by working status. There are 6 groups: group 1: working full time, group 2: working part time, group 3: not currently working, group 4: studying full time, group 5: housewife, group 6: other.
The main difference of our approach to the existing ones is that it does not require exact position information. We demonstrated its advantages over the the other approaches using multimodal data including GPS from the same dataset. The method discovered not only frequently visited locations but also the locations of infrequent visits. From the locations extracted using the WiFi data, we gained insights into human dynamics including the discovery of typical location patterns as well as the outlying days when the user had unusual location trajectories. The coverage time of our method is high compared to the method using multimodal data.

The main drawback of our method is the fragmentation of locations. The reason for this might be that we have not used time information for clustering. Thus, in the future work, we would like to add time information into the clustering process. We would also like to examine the graph clustering approaches for detecting the groups of WiFi access points.
Multi-Channel Nonparametric Clustering Model for Pervasive Data

In chapter 4, we presented a machine learning approach to discover significant locations and daily routines of individuals from WiFi data captured by their smartphones. We further analysed their human dynamics by computing the entropy over the distribution of locations and identified a significant correlation of entropy with age and working status. Unfortunately, while working well on single channel data such as WiFi data, this approach cannot be applied on heterogeneous data.

Pervasive data are usually presented in various types and from multiple sources. One typical example is data collected from mobile devices, especially smartphones. These devices are equipped with a wide range of sensors, where data collected from these sensors can be presented in different formats (i.e., continuous or category) and stored in multiple sources. Unfortunately, most existing machine learning methods are typically not designed to work with multi-channel heterogeneous data, making these data a problem for the machine learning and data mining research community.

In this chapter, we tackle the challenges of pervasive complex data. We leverage recent advances in Bayesian nonparametric (BNP) machine learning to propose a novel BNP model which is able to extract richer, high-order latent patterns from heterogeneous multi-channel data. Our model can also deal with missing data (i.e.
5.1 Motivation

Today, mobile devices come with a wide range of sensors that measure some physical environment’s parameters and convert them into signals, including motion sensors (accelerometer, gyroscope, pedometer), environmental sensors (magnetometer, barometer, thermometer, camera, microphone), physiological and biological sensors (heart rate, fingerprint), and connectivity sensors (Bluetooth, WiFi, GPS). Data collected from the sensors of mobile devices are used in a wide range of applications such as understanding individual human mobility (González et al., 2008), co-evolution modelling of human behaviors and social relationships (Dong et al., 2011), human interaction discovery (Do and Gatica-Perez, 2013), daily routine discovery (Nguyen et al., 2014b) and stress detection (Lu et al., 2012), to name a few. Unfortunately, as mobile data are recorded from various sensors, it is often presented in different formats (i.e. continuous and categorical) and stored in multiple data sources. Furthermore, being collected in the wild, data are often disrupted, irregular, and disparate, resulting in missing data which is difficult to deal with. Both issues are problematic to most machine learning methods which are typically designed to work with only one data type and/or not designed to work with missing elements. Moreover, useful patterns in heterogeneous data, such as daily life activities from mobile data, usually do not appear in the form of raw data. They need to be learned or inferred by exploiting the rich dependency between multiple channels of data. Extracting these hidden patterns, often known as latent variable modelling, has been challenging to the data mining and machine learning fields for a long time. Three key obstacles have been identified: (1) dealing with heterogeneous data sources results in a combinatorial explosion; (2) one does not often know how many patterns there are in advance due to the never-ending growth of data; and (3) making an inference in the presence of missing data is known to be extremely difficult.

Various methods have been proposed to tackle these challenges during the last decade. For example, a commonly used strategy is to apply those single-channel machine learning methods on each data channel independently, and then find a way to
5.1. Motivation

combine the results of all channels to form a final result (Cao et al., 2011). However, by treating each data channel separately, we ignore the correlation between data channels, which might contain valuable information to help discover the underlying structure of latent patterns and the interesting co-patterns among data channels. Generally speaking, the existing machine learning methods either solve the three above obstacles independently or they fail to address the problem of inferring latent activities and contexts from heterogeneous data sources.

During the last decade, Bayesian topic modelling has achieved impressive success. In particular, new models such as latent Dirichlet allocation (LDA) (Blei et al., 2003), hierarchical Dirichlet processes (HDP) (Teh et al., 2006), and the Indian buffet processes (IBP) (Griffiths and Ghahramani, 2006) have been developed and are commonly used in topic modelling applications, especially in learning and extracting topics from a document corpus. The new models have raised the field of latent variable modelling to a new high. Unfortunately, despite its tremendous performance in other applications, these models have little success in pervasive computing due to their inability to deal with pervasive data which usually come from multiple sources. Thus, handling heterogeneous or multi-channel data is the key to being able to use these Bayesian topic models in the pervasive computing world.

In this chapter, we propose a BNP model which is able to extract richer and high-order latent patterns from heterogeneous and multi-channel data, termed the multi-channel nonparametric clustering (MCNC) model. The key idea is to extend the machinery of the current BNP models through the use of a richer base measure distribution – being a product-space. The major strength of this framework lies in the product-space approach which gives our MCNC model the ability to simultaneously examine data from multiple sources and extract hidden patterns, even if the data have missing elements (which are treated as a by-product). In addition, the MCNC has all the advantages of BNP modelling with regard to automatically discovering the space of latent patterns from the data, thus it is not necessary to specify a number of patterns in advance.

We demonstrate the key properties and advantages of the MCNC model on a synthetic data as well as a real-world dataset collected from smartphones – the StudentLife dataset (Wang et al., 2014). Our aim is to discover the latent structures of identity–location–time patterns, which are one of the most fundamental context acquisition settings for a wide range of context-aware applications. To this end, we
extract the Bluetooth and WiFi-related data along with their timestamps from the StudentLife dataset. We then apply our MCNC model on the extracted data to discover who–where–when patterns. We further analyse the discovered patterns to demonstrate the merit of our model. We also quantitatively evaluate and report the performance of the model using standard metrics including F1-score, normalised mutual information (NMI), rand index (RI), and purity. These metrics are compared with those from popular baseline clustering methods including k-means (MacQueen, 1967), Gaussian mixture model (GMM) (McLachlan and Peel, 2004), DP-means (Kulis and Jordan, 2012), and Dirichlet process mixture (DPM) (Antoniak, 1974). Lastly, we demonstrate its ability to handle missing data by applying the MCNC model on the same dataset but with some random missing values. The experiment results show a consistent increase in the standard metrics when more data are observed.

Our main contributions in this chapter include: (1) a Bayesian nonparametric framework for pattern discovery that can simultaneously exploit all channels of heterogeneous data; (2) a framework that can handle missing values; and (3) a demonstration of the proposed model on a synthetic dataset as well as on the real-world StudentLife dataset.

5.2 MCNC Model

In this section, we present our proposed MCNC model and provide a theoretical proof of the marginalisation consistency of our model.

5.2.1 Multi-Channel Nonparametric Clustering (MCNC)

For simplicity, suppose each of our observations is now the 2-tuple \((x^{(1)}_i, x^{(2)}_i)\) where \(x^{(1)}_i\) and \(x^{(2)}_i\) can be completely different in the data types (e.g., one is continuous and the other is discrete). To generate each pair \((x^{(1)}_i, x^{(2)}_i)\) we introduce a product-space measure \(H_1 \times H_2\), each being responsible to generate parameters to explain \(x^{(1)}_i\) and \(x^{(2)}_i\) respectively through the likelihood functions \(F_1\) and \(F_2\). Hence, our stick-breaking representation now becomes: 
\[
G = \sum_{k=1}^{\infty} \pi_k \delta_{\psi_k}
\]
where each atom \(\psi_k\) is now a 2-tuple \(\psi_k = (\phi_k, \lambda_k)\) where \(\phi_k\) (s) are i.i.d drawn from \(H_1\) and \(\lambda_k\) (s) from \(H_2\). The construction of stick-breaking weights \(\pi_k\) (s) remains the same. For each data
5.2. MCNC Model

$\gamma \rightarrow G$

\[ \theta_i^{(1)} \rightarrow x_i^{(1)} \]

\[ \theta_i^{(2)} \rightarrow x_i^{(2)} \]

\[ H_1 \times H_2 \]

Figure 5.1: The stochastic view (left) and stick-breaking view (right) of the multi-channel nonparametric clustering model.

index $i$, we draw a sample from $G$, which is now a pair $\left( \theta_i^{(1)}, \theta_i^{(2)} \right)$, and generate data as $x_i^{(1)} \sim F_1(\cdot | \theta_i^{(1)})$ and $x_i^{(2)} \sim F_2(\cdot | \theta_i^{(2)})$, respectively. Figure 5.1 shows the graphical model of the MCNC model from the generative view (a.k.a., stochastic view) as well as the stick-breaking view.

In our model, the construction of the stick-breaking weights remains the same as in the standard DPM. Hence, the posterior inference can be efficiently computed using the collapsed Gibbs sampling method (Liu, 1994) as follows:

\[
p(z_i = k | \mathbf{x}, z_{i-1}, \gamma, H) \propto \begin{cases} 
\frac{n_k}{\gamma + n - 1} \times \int f_k^{-x_i^{(1)}}(x_i^{(1)}) \int f_k^{-x_i^{(2)}}(x_i^{(2)}) 
\quad \text{for } 1 \leq k \leq K \\
\frac{\gamma}{\gamma + n - 1} \times \int f_{k_new}^{-x_i^{(1)}}(x_i^{(1)}) \int f_{k_new}^{-x_i^{(2)}}(x_i^{(2)}) 
\quad \text{for } k = K + 1
\end{cases}
\]

(5.1)

where $f_k^{-x_i^{(1)}}(x_i^{(1)}) = \int_{\phi_k} p(x_i^{(1)} | x_{i-1}^{(1)}, \phi_k) d\phi_k$ is the predictive likelihood for $x_i^{(1)}$ conditional on the remaining variables denoted by $x_{i-1}^{(1)}$. Likewise, we can define the predictive likelihood for $f_k^{-x_i^{(2)}}(x_i^{(2)})$ and $f_{k_new}^{-x_i^{(1)}}(x_i^{(1)})$, $f_{k_new}^{-x_i^{(2)}}(x_i^{(2)})$. Via induction analogy, it is straightforward to generalise our process described above for more than two data sources.
5.2. MCNC Model

Sampling concentration parameter $\gamma$

A hyperparameter sampling makes the model more robust in identifying an unknown number of clusters. By robust, we mean the results are resilient against changes to the hyperparameters. The posterior distribution of the hyperparameter can be computed as a function of the prior hyperparameters and the observed data. Sampling $\gamma$ can be referred to (Escobar and West, 1995; Nguyen, 2015b). We place a Gamma prior over $\gamma$, assuming $\gamma \sim \text{Gamma}(\gamma_1, \gamma_2)$. Let $N$ be the number of data points, we define the auxiliary variable $t$ as $p(t | \gamma, K) \propto \text{Beta}(\gamma_1 + 1, N)$. Then, the posterior distribution for sampling $\gamma$ is:

$$p(\gamma | t, K) \sim \pi t \text{Gamma}(\gamma_1 + K, \gamma_2 - \log t) + (1 - \pi t) \text{Gamma}(\gamma_1 + K - 1, \gamma_2 - \log t)$$

(5.2)

where $\pi_k$ are computed as $\pi_t = \frac{\gamma_1 + K - 1}{N(\gamma_2 - \log t)}$ given the auxiliary variable $t$.

5.2.2 Marginalisation Property

It is worth noting that the major technical issue with the product-space approach to DPM is marginalisation consistency. That is, our specified model has to be consistent when one (or more) data channels are marginalised out. For example, in the above description, we must verify that marginalizing out the second data channel $x_i^{(2)}$ will recover a single DPM over the first data channel $x_i^{(1)}$. While this marginalisation property is less problematic to derive for parametric and finite models, the technicality can be complicated for BNP models. This is because the prior distribution for the mixture components is drawn from a stochastic process (DP), hence marginalisation consistency requires us to integrate all the way up to the base measure. In this section, we briefly present the main results of the proof of this marginalisation property. More details on the proof can be found in Nguyen et al. (2014c) and its supplementary materials.

Lemma 5.1. Let $S_1, \ldots, S_n$ be $n$ measurable sets in $\Sigma$. We form a measurable partition of $\Theta$, a collection of disjoint measurable sets, that generates $S_1, \ldots, S_n$ as follows. If $S$ is a set, let $S^1 = S$ and $S^{-1} = \Theta \setminus S$. Then $S^* = \left\{ \bigcap_{i=1}^n S_i^{c_i} | c_i \in \{-1, 1\} \right\}$ is a partition of $\Theta$ into a finite collection of disjoint measurable sets with the property that any $S_i$ can be written as a union of some sets in $S^*$. Let the elements of $S^*$ be
A_1, \ldots, A_{n^*} \) (note that \( n^* \leq 2^n \). Then the expectation

\[
\mathbb{E}_G [G(S_1), \ldots, G(S_n)] = \int_G \prod_{i=1}^{n} G(S_i) DP(dG \mid \alpha H)
\]

depends only on \( \alpha \) and \( H(A_i) \). In other words, the above expectation can be written as a function \( E_n(\alpha, H(A_1), \ldots, H(A_{n^*})) \).

It is easy to see that since \( S_i \) can always be expressed as the sum of some disjoints \( A_i \), \( G(S_i) \) can respectively be written as the sum of some \( G(A_i) \). Furthermore, by definition of the Dirichlet process, the vector \( G(A_1), \ldots, G(A_{n^*}) \) are distributed according to a finite Dirichlet distribution \( \alpha H(A_1), \ldots, \alpha H(A_{n^*}) \), therefore the expectation \( \mathbb{E}_G [G(S_i)] \) depends only on \( H(A_i)(s) \).

**Definition 5.1.** Consider a group of \( n \) data points, a Dirichlet process mixture prior is a probability measure over \( \Theta^n \ni (\theta_1, \ldots, \theta_n) \) with the usual product sigma algebra \( \Sigma^n \) such that for every collection of measurable sets \( \{(S_1, \ldots, S_n) : S_i \in \Sigma, i = 1, \ldots, n\} \):

\[
\text{DPM} (\theta_1 \in S_1, \ldots, \theta_n \in S_n | \alpha, H) = \int_G \prod_{i=1}^{n} G(S_i) DP(dG | \alpha H)
\]

We first state a result regarding the marginalisation of draws from a DP mixture with a joint base measure. Consider two measurable spaces \( (\Theta_1, \Sigma_1) \) and \( (\Theta_2, \Sigma_2) \) and let \( (\Theta, \Sigma) \) be their product space where \( \Theta = \Theta_1 \times \Theta_2 \) and \( \Sigma = \Sigma_1 \times \Sigma_2 \). Let \( H^* \) be a measure over the product space \( \Theta = \Theta_1 \times \Theta_2 \) and let \( H_1 \) be the marginal of \( H^* \) over \( \Theta_1 \) in the sense that for any measurable set \( A \in \Sigma_1 \), \( H_1(A) = H^*(A \times \Theta_2) \). Then, the process of drawing \( (\theta_i^{(1)}, \theta_i^{(2)}) \) from a DP mixture with base measure \( \alpha H \) and marginalising out \( \theta_i^{(2)} \) is the same as the process of drawing \( \theta_i^{(1)} \) from a DP mixture with base measure \( H_1 \). Formally, we have the following propositions and definitions.

**Proposition 5.1.** Denote by \( \theta_i \) the pair \( (\theta_i^{(1)}, \theta_i^{(2)}) \), there holds:

\[
\text{DPM} (\theta_1^{(1)} \in S_1, \ldots, \theta_n^{(1)} \in S_n | \alpha H_1) = \text{DPM} (\theta_1 \in S_1 \times \Theta_2, \ldots, \theta_n \in S_n \times \Theta_2 | \alpha H^*)
\]

for every collection of measurable sets \( \{(S_1, \ldots, S_n) : S_i \in \Sigma_i, i = 1, \ldots, n\} \).

**Proof.** Since \( \{(S_1, \ldots, S_n) : S_i \in \Sigma_i, i = 1, \ldots, n\} \) are rectangles, expanding the \( \text{RHS} \)
using Definition 5.1 gives:

\[ \text{RHS} = \int G(S_1 \times \Theta_2) \ldots G(S_n \times \Theta_2) \, d\text{DP}(dG \mid \alpha H^*) \]

Let \( T_i = S_i \times \Theta_2 \), the above expression is the expectation of \( \prod_i G(T_i) \) when \( G \sim \text{DP}(dG \mid \alpha H^*) \). Form a collection of the disjoint measurable sets \( T^* = (B_1 \ldots B_{n^*}) \) that generates \( T_i \), then note that \( B_i = A_i \times \Theta_2 \), and \( S^* = (A_1 \ldots A_{n^*}) \) generates \( S_i \). By definition of \( H_1 \), \( H_1(A_i) = H^*(A_i \times \Theta_2) = H^*(B_i) \). Using Lemma 5.1 above, \( \text{RHS} = E_n(\alpha, H^*(B_1), \ldots, H^*(B_{n^*})) \), while \( \text{LHS} = E_n(\alpha, H_1(A_1), \ldots, H_1(A_{n^*})) \) and they are indeed the same.

We note that \( H^* \) can be any arbitrary measure on \( \Theta \) and, in general, we do not require \( H^* \) to be factorised as a product measure.

Algorithm 5 presents the pseudo-code framework for our MCNC model. The termination criteria could be a given number of iteration or a threshold for log-predictive improvement.

**Input**: data points \( \{x_1, \ldots, x_N\} \), prior number of cluster \( K \)

**Output**: mixture indicators \( \{z_1, \ldots, z_N\} \)

**Initialize**: randomly assign \( \{z_1, \ldots, z_N\} \) to clusters \( \{1, \ldots, K\} \)

**while** termination criteria are not met **do**

\[ \text{for } i \text{ from } 1 \text{ to } N \text{ do} \]

\[ \text{Sampling } z_i \text{ using equation 5.1} \]

\[ \text{end} \]

\[ \text{Sampling hyperparameter } \gamma \text{ using equation 5.2} \]

**end**

**Algorithm 5**: Pseudo-code framework for MCNC model.

### 5.2.3 Handling Missing Data

The sampling equations and procedures developed in section 5.2.1 show that missing values can be seamlessly handled. For example, if \( x_1^{(1)} \) is missing, then the likelihood can be set to \( f_{\tilde{x}_1^{(1)}}(x_1^{(1)}) = 1 \) since there is no observation, or if other variables are missing, they can be systematically ignored in the predictive likelihood computation. Therefore, our model can handle missing data points naturally. By missing data points, we mean that one or some of the data sources are not seen, but we can
still observe at least one data source. This is in contrast with DPM where it is not trivial to handle these data points. A simple hack for DPM in this case is to ignore these missing data points completely. However, this fails to leverage on the partial observed information.

5.3 Experiments

To demonstrate our framework, we first applied it on synthetic data, then on the StudentLife dataset. Our experiments were run on a PC equipped with an Intel Xeon E5-2460 CPU (8 cores, 2.6Ghz) and 16GB of RAM. To compute the posterior inference, we ran collapsed Gibbs sampling over 500 iterations. On average, each iteration costs around 19 seconds.

5.3.1 Experiments with Synthetic Data

To verify the properties of the MCNC model and its advantages in modelling multiple data sources simultaneously, we experimented with synthetic data for which the ground truth is known. The synthetic data were generated as follows. We first generated $K = 6$ atoms $\psi_k = (\phi_k, \lambda_k), k = 1 \ldots K$, where $\phi_k$ are multinomial distributions and $\lambda_k$ are univariate Gaussian distributions. Each $\phi_k$ had been designed as follows. We created six $7 \times 5$ dot matrices of character C-A-N-C-U-N (as shown in Figure 5.2) and then flatten these dot matrices to 35 dimension multinomial vectors. We noted that the 6 atoms $\phi_k$s (top row) were replicated from 4 unique multinomial distributions, and the 6 atoms $\lambda_k$s (bottom row) were replicated from 3 unique univariate Gaussian distributions. After this, we generated 2000 2-tuple data points $\left(x_i^{(1)}, x_i^{(2)}\right)$. First, we drew uniformly a cluster $c_i \in [1 \ldots K]$ for each data point. Then, we used the multinomial distribution $\phi_{c_i}$ of the corresponding atom $\psi_{c_i}$ to generate $x_i^{(1)}$ and used the univariate Gaussian distribution $\lambda_{c_i}$ to generate $x_i^{(2)}$. This synthetic data represent a common situation in real-world applications where data channels can be in continuous as well as in discrete domains. We used this synthetic data to verify the capability of MCNC in identifying the appropriate co-patterns existing in the data with heterogeneous sources.

We applied the MCNC model on the generated data. Our model correctly recovers the 6 true co-patterns. In contrast, DPM can only utilise single channel information,
thus it failed to learn the co-patterns existing from the data. Figure 5.3 shows the clusters discovered by the DPM model running on the multinomial channel of the above synthetic data. Because the multinomial observations from true clusters 1 and 4 (Figure 5.2) had been generated from the same multinomial distribution, they merged into one cluster. It was also true for multinomial observations from true clusters 3 and 6. Therefore, DPM resulted in only 4 instead of the true 6 clusters. This can be explained by observing that a single data channel does not help to obtain all the properties of the whole data which are far more complicated.

5.3.2 The StudentLife Dataset

The StudentLife study\(^1\) Wang et al. (2014) is a real-world pervasive dataset collected from the smartphones of 48 students at Dartmouth College over a 10-week spring term in 2013. It originally aimed to assess students’ mental health (e.g., depression, loneliness, stress), academic performance (grades across all their classes, term GPA and cumulative GPA) and behavioral trends (e.g., how behaviors – such as stress,}

\(^1\)http://studentlife.cs.dartmouth.edu/
sleep, visits to the gym – change in response to college workload – i.e., assignments, midterms, finals – as the term progresses). To achieve this aim, an application (called the StudentLife app) was installed on students’ phones to automatically collect data without any user interaction. A wide range of human behaviours was measured including bed time, sleep duration, conversations, physical activities, users’ locations, surrounding people, outdoor and indoor mobility, stress level, positive affect, eating habits, app and phone usage, and more. More details can be found at the URL http://studentlife.cs.dartmouth.edu/dataset.html.

In this chapter, as we aim to demonstrate our framework on discovering who–when–where patterns, we limit the use of the StudentLife dataset to the Bluetooth and WiFi data. The Bluetooth data are used to identify nearby people (who), while the WiFi data are used to infer the significant locations of users (where). Both WiFi and Bluetooth data have been associated with timestamps that we use to infer the change of patterns over time (when). In addition, the dataset also includes the ground truth information about locations (indicated by building names) where Bluetooth/WiFi scans take place. We extracted data in Bluetooth, WiFi and timestamp channels as described below.

**WiFi data** As mentioned, an app was installed on students’ smartphones to automatically collect environmental data using integrated sensors on the phones. This app collected WiFi data by frequently scanning (every 2 minutes) for surrounding WiFi access points. If detected, the app recorded the timestamp, BSSID (access points’ MAC address), frequency and signal strength values of the surrounding access points. The BSSIDs and signal strengths were then used to determine if a student

**Figure 5.3:** The DPM estimates $K = 4$ topics on the multinomial channel of synthetic data. It cannot discover the 6 true co-patterns as it does not utilise data from the second channel.
was in a specific building or near some buildings. If yes, the building names were
recorded in the dataset along with the timestamp of the scans. Note that the access
point deployment information, which was used to map WiFi access points to a specific
location (e.g., a building name) was not shared in the dataset. In total, there are
more than 19 million WiFi scans in the dataset, with 118,505 unique WiFi IDs.

The WiFi data were then processed as follows. We eliminated access points that had
been scanned less than 10 times as they are not statistically meaningful and do not
affect the location discovery. Empty scans after this elimination were also removed.

<table>
<thead>
<tr>
<th>Time</th>
<th>WiFi ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>14/2/2011 08:02:01</td>
<td>w₁, w₂</td>
</tr>
<tr>
<td>14/2/2011 08:04:01</td>
<td>w₂</td>
</tr>
<tr>
<td>14/2/2011 08:06:02</td>
<td>w₁, w₂, w₄</td>
</tr>
</tbody>
</table>

**Table 5.1**: Example of processed WiFi scans.

**Bluetooth data**  Every 10 minutes, the StudentLife app scans for nearby Bluetooth
devices and records their IDs. There were roughly 1.3 million Bluetooth scans with
5,486 unique Bluetooth IDs. We also eliminated the Bluetooth devices that had been
scanned less than 10 times, and empty scans after this elimination were removed.

<table>
<thead>
<tr>
<th>Time</th>
<th>Bluetooth ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>14/2/2011 08:02:01</td>
<td>b₁</td>
</tr>
<tr>
<td>14/2/2011 08:06:02</td>
<td>b₁, b₃</td>
</tr>
</tbody>
</table>

**Table 5.2**: Example of processed Bluetooth scans.

**Matching Bluetooth and WiFi scans**  The Bluetooth and WiFi scans sometimes
had different timestamps due to the design of the data collecting application. However,
in our experiments, a Bluetooth and a WiFi scan were considered to be from the
same scan if they had exactly the same timestamp.
5.3. Experiments

<table>
<thead>
<tr>
<th>Time</th>
<th>Bluetooth ID</th>
<th>WiFi ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>14/2/2011 08:02:01</td>
<td>$b_1$</td>
<td>$w_1, w_2$</td>
</tr>
<tr>
<td>14/2/2011 08:06:02</td>
<td>$b_1, b_3$</td>
<td>$w_1, w_2, w_4$</td>
</tr>
</tbody>
</table>

Table 5.3: Example of matched Bluetooth and WiFi scans.

**Timestamp data**  Each timestamp contains information on time, i.e., year, month, day, hour, minute and second. However, as we aimed to discover interesting context patterns during a day, we ignored the year, month and day information. Furthermore, we converted hours, minutes and seconds to a decimal format (e.g., 10 hours and 30 minutes to 10.5 hours).

Table 5.4 shows an example of the extracted dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Bluetooth ID</th>
<th>WiFi ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.0336</td>
<td>$b_1$</td>
<td>$w_1, w_2$</td>
</tr>
<tr>
<td>2</td>
<td>8.1006</td>
<td>$b_1, b_2, b_3$</td>
<td>$w_1, w_2, w_4$</td>
</tr>
</tbody>
</table>

Table 5.4: Example of extracted dataset.

**Indoor scans**  For the convenience of evaluation step, we used only Bluetooth and WiFi scans which took place inside a building. The building names are then used as location ground-truth in the quantitative evaluation of our model’s performance.

5.3.3 Discovering the *Who-When-Where* Co-patterns with Complete Data

We fed the extracted dataset in Section 5.3.2 to our MCNC framework. The experiment ended up with 14 patterns (clusters). Each pattern has three elements: a Gaussian distribution (parametrised by a mean and a variance) and two multinomial distributions (parametrised by weight vectors). It presents the time at which the activity occurs, the group of users who participate in the activity (proximity pattern) and the group of WiFi hotspot at the place where the activity occurs (location pattern). Table 5.5 visualises 6 significant patterns of proximity (*who*) - location...
(where) - time (when) discovered by our MCNC model. Both the Bluetooth ID and WiFi ID are displayed using tag clouds, in which the size of each number reflects the contribution of the ID to the pattern. As a result, some IDs associated with a very small probability would not be seen on the plot.

By making a comparison between the patterns in Table 5.5, we can obtain interesting information on the discovered activities. For example, we can see that patterns 6 and 14 represent a proximity group (dominated by the user who owns the device corresponding to Bluetooth ID #5) at the same location (Sudikoff building). However, their corresponding timestamps are different in the early morning (around 4:00) and at lunch time (around 13:00), respectively. Another interesting example can be seen in patterns 1 and 7. These two patterns represent events that happened at slightly different proximity and location groups, but with the same significant user #69. In our opinion, these patterns would be hard to discover if one treats each data channel (i.e. Bluetooth and WiFi) independently. This demonstrates the advantage of our model over traditional approaches.

To facilitate the exploration of the discovered patterns, we created an interactive demonstration\(^2\) that displays patterns in the form of a network (as shown in Figure 5.4). When a user clicks on a node (e.g., a location topic node), it emphasises the top 5 significant WiFi IDs that contribute the most to this location topic, and hides the other nodes. It also connects the location topic node to corresponding proximity topic and time nodes. This applet makes it easier to see which WiFi IDs belong to a given location topic, or which Bluetooth IDs belong to a specific proximity pattern.

### 5.3.4 Evaluation of Performance

After being clustered by the MCNC model, each data point was assigned to a cluster through the indicator \(z_i = k\). Moreover, from the ground truth, the location of each scan can also be known. We relied on this information to evaluate the clustering performance of our approach. To conduct a quantitative comparison, we used four clustering baselines, namely k-means (MacQueen, 1967), Gaussian mixture model (GMM) (McLachlan and Peel, 2004), DP-means (Kulis and Jordan, 2012), and standard DPM (Antoniak, 1974). We note that the k-means and GMM

\(^2\)Readers are referred to appendix A for more details about the interactive demonstration.
Table 5.5: The patterns of proximity (*who*), location (*where*) and time (*when*)
Figure 5.4: All 14 discovered patterns are displayed as a network. Each location topic is linked to 5 WiFi IDs that contribute the most to that location topic, and each proximity topic is connected to 5 Bluetooth IDs that frequently participate in that proximity topic. Location and proximity topics are linked to a corresponding time node showing the time of the activity.
5.3. Experiments

clustering methods require a number of clusters $K$, and this $K$ can significantly affect the performance score. Our approach, DPM and DP-means, in contrast, can automatically learn the number of patterns from the data. Therefore, to be fair, we run k-means and GMM with different values of $K$ (from 2 to 30) and report the average values. The performance scores over the dataset are presented in Table 5.6. The MCNC model outperforms its competitors on F1-score and RI while achieving good scores on NMI and purity.

<table>
<thead>
<tr>
<th>Algo.</th>
<th>Clusters</th>
<th>F1</th>
<th>NMI</th>
<th>RI</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means</td>
<td>2-30</td>
<td>0.464</td>
<td>0.465</td>
<td>0.659</td>
<td>0.694</td>
</tr>
<tr>
<td>GMM</td>
<td>2-30</td>
<td>0.442</td>
<td>0.422</td>
<td>0.644</td>
<td>0.653</td>
</tr>
<tr>
<td>DP-means</td>
<td>19</td>
<td>0.462</td>
<td><strong>0.667</strong></td>
<td>0.765</td>
<td><strong>0.850</strong></td>
</tr>
<tr>
<td>DPM</td>
<td>14</td>
<td>0.451</td>
<td>0.107</td>
<td>0.502</td>
<td>0.475</td>
</tr>
<tr>
<td>MCNC</td>
<td>14</td>
<td><strong>0.480</strong></td>
<td>0.589</td>
<td><strong>0.811</strong></td>
<td>0.782</td>
</tr>
<tr>
<td>k-means</td>
<td>14</td>
<td>0.431</td>
<td>0.471</td>
<td>0.704</td>
<td>0.673</td>
</tr>
<tr>
<td>GMM</td>
<td>14</td>
<td>0.376</td>
<td>0.311</td>
<td>0.539</td>
<td>0.604</td>
</tr>
</tbody>
</table>

Table 5.6: Performance of location clustering. As k-means and GMM cannot identify a suitable number of clusters $K$, we vary $K$ from 2 to 30 and report the average values. Results of k-means and GMM in case $K = 14$ are also reported.

5.3.5 Discovering the Who-When-Where Co-patterns with Missing Data

To demonstrate the pattern discovery ability of our framework on data with missing values, we varied the amount of missing data generated from the complete dataset in Section 5.3.2. More specifically, we randomly chose $m$ scans, where $m$ ranges from 80%, 60%, 40% and 20% of the total scans, and set their Bluetooth data as missing values. Each of these generated datasets was then inputted to the MCNC model and the performance metrics were calculated (similar to the complete data settings). The experiment results show that performance scores increase when there is less missing data, as shown in Figure 5.5. In other words, the more data which are observed, the better the patterns are. It shows that our framework can take into account each single observed data point to learn better patterns.
5.4 Concluding Remarks

We presented the multi-channel nonparametric clustering (MCNC) model for pattern discovery from multiple sources with missing values. Leveraging on the power of the Bayesian nonparametric modelling approach, this model can automatically grow the model complexity (patterns) from the data. In addition, the main advantage of our approach over existing methods is that it can handle data from multiple sources, process them simultaneously and treat them equally. Furthermore, our approach can also deal with data with missing values. We demonstrate the approach on the StudentLife dataset, a real-world dataset collected from mobile phones, to discover who-when-where co-patterns that have their values in context-awareness applications. The obtained results showed the merit of our model, including the ability to learn complex hidden patterns from multiple data channels with missing values. More importantly, our framework offers the flexibility to interactively explore the high-order relationship among heterogeneous data sources while the existing methods are unable to do so.

Figure 5.5: Experiments with missing data. The lower the proportion of missing data, the better the clustering performance. DPM cannot handle missing data, thus these data points will be discarded. In contrast, MCNC can utilise the missing data points by modelling other available channels.
Chapter 6

Learning Hierarchical Representation in Noisy, Heterogeneous and Multi-Channel Data

In chapters 5, we presented the MCNC (multi-channel nonparametric clustering) model which is able to simultaneously manipulate multi-channel data, such as Bluetooth and WiFi data captured from mobile phones. This model was developed on top of the regular Dirichlet process mixture (DPM) model and extended its base measure being a product-space. Theoretically, a key challenge of the product-space approach is to ensure the marginality property of the model, i.e., if we marginalise one (or some) random variables, the model is still able to recover the distributions of the remaining random variables. We also demonstrated our MCNC model on synthetic data and on a real-world dataset. The experiment results showed that the MCNC model achieved high performance on standard clustering metrics in comparison to baseline methods.

Unfortunately, the learning capability of the MCNC model is still limited on flat data. In other words, it is unable to learn the richer structure, such as hierarchical structure of the data. In this chapter, we develop a novel Bayesian nonparametric model to discover hierarchical latent patterns in pervasive noisy, heterogeneous and multi-channel data.
6.1 Motivation

Ubiquitous computing relies on embedding devices to capture information in daily activities from various channels, such as Bluetooth, WiFi, and accelerometers. These data are usually presented in various formats (e.g., continuous values, binary values, or categorical values). In addition, ubiquitous data in many situations are organised into groups. For example, let us examine data collected from the smartphones of students in a university. At the first level, data can be grouped by student since the data of one student reflect his own behaviour and may be different to the behaviour of other students. At the second level, data can be grouped by school since the data of students from different schools may reflect different behavioural patterns. Usually, useful patterns in such ubiquitous, heterogeneous and hierarchical data, such as abstract contexts and daily life activities from mobile data, are not directly observed from the raw signals. They need to be learned or inferred. However, most machine learning methods run into trouble when extracting these latent patterns due to the diversity and heterogeneity of data presented as the hierarchical structures. Despite decades of research, this still remains one of the most challenging problems.

Recent advances in Bayesian nonparametric machine learning, in particular the theory of topic models has provided an elegant solution to these challenges. However, none of the existing methods has addressed the problem of inferring latent multifaceted activities and contexts from heterogeneous data such as those collected from mobile devices. For example, the latent Dirichlet allocation (LDA) model (Blei et al., 2003) and its Bayesian nonparametric version, the hierarchical Dirichlet processes (HDP) (Teh et al., 2006) are tremendously successful latent topic models, which extract latent patterns in the form of probability distributions. Unfortunately, although the statistical foundation is generic, these models were originally developed to deal with a single data channel, often to model words in document corpus.

In this chapter, we extend the theory of HDP to extract richer, high-order latent patterns from heterogeneous data through the use of a richer base measure distribution being the product-space. This approach allows heterogeneous data from multiple sources to be exploited simultaneously and takes their correlating information into account during the learning process. The model has all the advantages of nonparametric modelling to automatically discover the space of latent patterns and activities from the data. In addition, its major strength lies in the product-space approach which allows the model to deal with multi-channel heterogeneous data, even if the
6.1. Motivation

We name our model the product-space hierarchical Dirichlet processes (PS-HDP).

We demonstrate the key properties and advantages of the proposed framework on a synthetic dataset as well as on the StudentLife dataset (Wang et al., 2014), a real-world dataset collected from mobile phones. We discover the latent structures of identity–location–time (a.k.a who–where–when) patterns, which are known to be one of the most fundamental context acquisition settings for a wide range of applications such as mobile context-aware applications (Schilit and Theimer, 1994) or understanding human dynamics (Pentland, 2007). To this end, we first extract Bluetooth and WiFi-related data along with their timestamps from the StudentLife dataset. The Bluetooth data are used to infer nearby people (identity), whilst the WiFi data are used to infer the significant locations of users (location). Then, we feed these data into our PS-HDP model to discover who–where–when patterns at multiple levels including (1) the global level (i.e., patterns occur across all participants) and (2) the local level (i.e., patterns occur for a specific participant). We also visualise and provide an analysis of the activities and patterns learned from our model to demonstrate the merit of our proposed framework. We further quantitatively evaluate and report the performance of the PS-HDP model using standard metrics including F1-score, normalised mutual information (NMI), rand index (RI) and purity. These metrics are then compared to those from popular clustering methods, including k-means (MacQueen, 1967), non-negative matrix factorization (NNMF) (Lee and Seung, 1999), Gaussian mixture model (GMM) (McLachlan and Peel, 2004), DP-means (Kulis and Jordan, 2012), and Affinity Propagation (AP) (Frey and Dueck, 2007). Lastly, we demonstrate the ability of the PS-HDP model in learning activities with missing data, a common problem encountered in pervasive and ubiquitous computing applications. The experiment results show that the more data are observed (i.e., less missing data), the more F1-score increases.

In short, our main contributions in this chapter are threefold. First, we propose a product-space extension of the Bayesian nonparametric model and the theoretical guarantee of the statistical marginalisation consistency. Our model is able to simultaneously model multi-channel heterogeneous data from multiple sources and discover hierarchical, multifaceted latent structures from data that can be explained jointly. Second, we demonstrate an application for learning the hidden activities from heterogeneous data collected from multiple sensors, which is important in the area of pervasive and ubiquitous computing. We apply our proposed model on
the StudentLife dataset, a real-world dataset collected from mobile phones. Third, beyond the discovery of activities in the form of who-when-where, our framework is generic and readily applicable to infer any arbitrary number of activity types which can be ubiquitously found in many modern mobile datasets (such as type of phone call, what is being discussed and so forth). Furthermore, recent advances in probabilistic inference, such as stochastic variational inference (Hoffman et al., 2013) can be derived so that our proposed framework can be scaled to big datasets.

6.2 Additional Related Background

In addition to literature survey in chapter 2, we present in this section relevant additional related work on machine learning approaches which are effectively applicable on pervasive data. In particularly, we focus on Bayesian nonparametric approaches for pervasive data with its complex and heterogeneous characteristics as presented in chapter 2.

6.2.1 Pattern Discovery from Heterogeneous Data

The discovery of hidden patterns from multi-channel heterogeneous data has been a challenge in machine learning and data mining. As machine learning algorithms are typically designed to work with only one specific data type (e.g., continuous, discrete), most of the previous studies treat each data channel separately. For example, in the combined mining method (Cao et al., 2011), $K$ different miners have been applied on $K$ data sources to get $K$ corresponding sets of patterns. After this, a merger is used to combine these sets to obtain global patterns (i.e., patterns from all data sources). Unfortunately, this approach is usually time-consuming, and more importantly, is unable to exploit the correlating information between these data sources during the learning process to create better patterns.

Recently, researchers have been trying to leverage the advances of the Bayesian nonparametric approach to propose unifying models to learn and discover patterns from multiple data sources. One common strategy is to model one data source as the primary data (called content), while treating other data sources as secondary data (called contexts). Contexts are viewed as distributions over some index space and both contents and contexts are modelled jointly. For example, Phung et al.
(2012) propose an integration of the HDP and the nested Dirichlet process (nDP) with shared mixture components to jointly model contexts and contents respectively, where mixture components could be integrated under a suitable parameterisation. The authors showed that their model achieved good results in the field of computer vision. In another multilevel clustering work, Nguyen et al. (2014c) used documents as primary data, and used other information (e.g. time, authors) as contexts. In this model, secondary data channels should be collected in group-level, called group-level contexts (Nguyen et al., 2015). More recently, Huynh et al. (2015) developed a Bayesian nonparametric approach to model correlation structures among multiple and heterogeneous data sources. Choosing a data source as the primary data (content), they induce a mixture distribution over the data using HDP. Other data sources, also generated from HDP, are treated as context(s) and assumed to be mutually independent given the content. However, in some applications, choosing one data source to be the primary data is not an easy task. Our approach differs from this one as it treats all data channels equally and hence does not need to specify contexts and content.

Another common issue with a mobile dataset is missing observations. A traditional approach for dealing with missing data is to employ imputation methods based on the statistics of the data. Different imputation methods have been proposed (e.g., mean, median, conditional mean substitution, closest fit, maximum likelihood, etc.) for different missing data mechanism (e.g., missing completely at random, missing at random, or not missing at random). Interested readers are referred to (Little and Rubin, 2014) for more details.

In addition to the imputation approach, many other approaches were also proposed to deal with missing values. Grzymala-Busse and Hu (2000) reviewed and compared nine approaches to missing attribute values in data mining. They preferred the C4.5 approach (Quinlan, 2014) and the method of ignoring examples with missing attribute values. Recently, the missing data problems in machine learning were deeply studied in the work of Marlin (2008), where different learning methods with missing data were presented and discussed. He also presented the collapsed Gibbs sampler for the DPM model with random missing values, showing the ability of Bayesian nonparametric approaches in dealing with missing data.
6.3. Proposed Models

6.2.2 Discovery of Interaction and Mobility Patterns from Bluetooth and WiFi Data

Bluetooth data are widely used in proximity detection problems. In particular, mobile Bluetooth data are usually used to detect surrounding people to discover interaction patterns. Do and Gatica-Perez (2011, 2013) used large daily life Bluetooth data captured by smartphones to create a dynamic social network. Then, by using their proposed probabilistic model, they discovered different social contexts (such as group meetings or dinner with the family) and interaction types (e.g. office interaction, personal interaction). In another approach, Nguyen et al. (2013) used HDP to discover interaction types from Bluetooth data from honest social signals captured by sociometric badges. Then, they clustered mixture proportions from HDP using the Affinity Propagation algorithm to extract contexts and communities.

While Bluetooth data are useful for discovering surrounding people, WiFi data are usually used to infer locations. A smartphone is able to scan for surrounding WiFi hotspots. Each WiFi hotspot has a unique identified fingerprint (i.e. its MAC address). Assuming that the location of a WiFi hotspot never changes, one can use its ID as an indicator of a location. From the raw WiFi scans, different approaches are used to discover locations. Dousse et al. (2012) used OPTICS clustering to group similar scans to a cluster representing a place. However, with the OPTICS algorithm, the number of clusters must be provided beforehand. Nguyen et al. (2014b), in contrast, used the AP algorithm to discover interesting locations without the need to specify the number of clusters.

6.3 Proposed Models

As presented in section 2.1, data captured from mobile phones or wearable devices usually come from various sensors integrated in the devices. Moreover, data captured in the wild are usually disrupted and as a result, missing data becomes a challenge in learning and modelling. In this section, we present our proposed model which is able to handle heterogeneous data as well as missing data.
6.3.1 Product-space Hierarchical Dirichlet Processes

Hierarchical Dirichlet processes (HDP) model (Teh et al., 2006) is a powerful topic model to learn latent topics and patterns. However, it was originally designed for a single data channel and is thus unable to handle heterogeneous data. To overcome this limitation of the original HDP, we present the product-space hierarchical Dirichlet processes (PS-HDP) model which is an extension of the HDP model using a richer structure for the base measure being a product-space. It can be described as follows.

The data consist of $J$ groups, each of which contain $N_j$ data points. Each data point $i$ (in group $j$) is a collection of observations, denoted as $\{x_{ji}^1, x_{ji}^2, ..., x_{ji}^C\}$ from $C$ data sources which can be heterogeneous from different distributions. For example, in case of the StudentLife dataset (section 5.3.2), we can assume a group is a user and a data point is an interaction of the users. Each interaction includes some heterogeneous channels of information such as $x_{ji}^1$ is a timestamp (Gaussian distribution), $x_{ji}^2$ is a Bluetooth signal (multinomial distribution), and $x_{ji}^3$ is a WiFi signal (multinomial distribution).

6.3.1.1 Stochastic Representation

Let $H_1, H_2, ..., H_C$ be the base measure (for each data source) generating the global atom $G_0$ from the Dirichlet process with concentration parameter $\gamma$ as $G_0 = \langle G_0^1 G_0^2 ... G_0^C \rangle \sim \text{DP}(\gamma, H_1 \times H_2 \times ... H_C)$ where $H_1 \times H_2 \times ... H_C$ is the product of the base measure. Then, each group $j$ will have a local atom $G_j$ drawn from the Dirichlet process with the concentration parameter $\alpha$ and the global atom $G_0$ as $G_j = \langle G_j^1 G_j^2 ... G_j^C \rangle \sim \text{DP}(\alpha, G_0)$. In other words, each random group-specific mixture distribution $G_j$ shares the same base probability measure $G_0$. Next, the atom for each data point $i$ in group $j$ at channel $c$ is iid drawn as $\theta_{ji}^c \sim G_j^c$ and the observation is generated subsequently $x_{ji}^c \sim F_c(\theta_{ji}^c)$.

6.3.1.2 Stick-breaking Representation

For posterior inference, we characterise the above stochastic view using stick-breaking representation. We draw the global weight $\beta \sim \text{GEM}(\gamma)$ and the topics (or patterns) $\phi_k^c \sim H_c, \forall k=1, ..., \infty, \forall c=1...C$ such that the global atom $G_0 = \sum_{k=1}^{\infty} \beta_k \delta(\phi_k^1, ..., \phi_k^C)$. For each group $j$, the local weight $\pi_j \sim \text{DP}(\alpha, \beta)$ such that $G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta(\phi_{jk}^1, ..., \phi_{jk}^C)$. 

Then, the data point label $z_{ji} \sim \text{Mult}(\pi_j)$. Finally, the observation is generated using the corresponding topic $\phi_{z_{ji}}$ as $x_{ji} \sim F^c(\phi_{z_{ji}})$, $\forall j = 1, 2, ..., J$, $\forall i = 1, ..., N_j$, $\forall c = 1, ..., C$.

Figure 6.1 shows the graphical models of the PS-HDP from the stochastic view and the stick-breaking view.

### 6.3.1.3 Posterior Inference

Our framework is a Bayesian model. For posterior inference, we utilise collapsed Gibbs sampling (Liu, 1994). To integrate $\pi_j$ and $\phi_k$ due to the conjugacy property (Diaconis and Ylvisaker, 1979), the two latent variables $z_{ji}$ and $\beta$ need to be sampled.

**Sampling $z_{ji}$**. We assign a data point $x_{ji}$ to its component $\phi_k$. The conditional distribution for $z_{ji}$ is influenced by a collection of words associated with topic $k$. 
across the documents:

\[ p(z_{ji} = k \mid \mathbf{x}, \mathbf{z}_{-ji}, \alpha, \beta, H) = p(z_{ji} = k \mid \mathbf{z}_{-j}, \alpha, \beta) \times \]
\[ p(x_{ji} \mid z_{ji} = k, \{x_{j'i'} \mid z_{j'i'} = k, \forall (j'i' \neq ji)\}, H) \]
\[ = \begin{cases} 
(n_{jk}^{-ji} + \alpha \beta_k) \times f_k^{-x_{ji}}(x_{ji}) & \text{used } k \\
\alpha \times \beta_{\text{new}} \times f_{k_{\text{new}}^{-x_{ji}}}(x_{ji}) & \text{new } k.
\end{cases} \]

The first term is recognised as the Chinese Restaurant Franchise (number of data points in group \( j \) follows topic \( k \)). The second term is the predictive likelihood \( f_k^{-x_{ji}}(x_{ji}) \) which can be analytically evaluated using the conjugate property (Diaconis and Ylvisaker, 1979).

**Sampling \( \beta \).** We sample the global mixing weight follow the approach presented in (Teh et al., 2006) where we sample \( \beta \) jointly with the auxiliary variable \( \mathbf{m} \) (0 ≤ \( m \) ≤ \( n_{jk} \)):

\[ p(m_{jk} = m \mid \mathbf{z}, \mathbf{m}_{-jk}, \beta) \propto \text{Stirl}(n_{ij}, m_{jk}) (\alpha \beta_k)^m \]
\[ p(\beta \mid \mathbf{m}, \mathbf{z}, \alpha, \gamma) \propto \beta^{-1}_{\text{new}} \prod_{k=1}^{K} \beta_{\sum_{j} m_{jk} - 1}^{-1} \]

**Sampling hyperparameters \( \alpha \) and \( \gamma \).** To make the model robust in identifying the unknown number of clusters, we resample the hyperparameters in each Gibbs iteration. The lower concentration parameter \( \alpha \) is described in (Teh et al., 2006). The upper concentration parameter \( \gamma \) follows the techniques of Escobar and West (1995).

### 6.3.2 Marginalisation Property

At the intuition level, our extension to HDP can broadly be attributed to a factorised likelihood to the HDP. However, this can be deceiving as a trivial extension due to the nonparametric Bayesian setting whose marginalisation property is not immediately obtained. Unlike a parametric model, since our prior distribution for the mixture components is a draw from a stochastic process, the marginalisation consistency (i.e., to ensure that marginalizing over other data channels will result in a consistent
marginal distribution on a data channel of interest) requires us to integrate all the way up to the base measure. Marginalisation consistency is particularly important in the case of missing data to ensure the correctness of the model wherein the missing values are usually ‘filled up’ with the expectation evaluated over the conditional distribution of the hidden variables given the observed variables. We state the main results in this section.

Let $H$ be a base measure over some measurable spaces $(\Theta, \Sigma)$ and $\alpha > 0$ is a positive number called concentration hyper-parameter. Furthermore, let $\mathcal{P}$ be the set of all probability measures over $(\Theta, \Sigma)$, suitably endowed with some $\sigma$-algebra $\sigma(\mathcal{P})$, then a Dirichlet process (DP) is a probability measure over $(\mathcal{P}, \sigma(\mathcal{P}))$ such that a draw $G$ from it is a probability measure on $(\Theta, \Sigma)$ whose resulting (random) vector $(G(A_1), G(A_2), \ldots, G(A_m))$ is distributed according to a Dirichlet distribution parameterised by $(\alpha H(A_1), \alpha H(A_2), \ldots, \alpha H(A_m))$ where $\{A_1, \ldots, A_m\}$ is any arbitrary partition of $\Theta$. We further say that this DP is parameterised by the base measure $H$ and concentration parameter $\alpha$ and write $G \sim \text{DP}(\alpha H)$. As it becomes clear, a DP is then a distribution over distributions. It can also be viewed as a stochastic process whose index set is $\Sigma$ since each $G(A_i)$ is a random variable with $A_i \in \Sigma$.

Our situation at hand (HDP) is a setting of grouped data consisting of $J$ groups and $N_j$ data points within each $j$-th group. To properly define a hierarchical prior over this grouped data, we start with the DPM prior which we defined in section 5.2.2.

**Definition 6.1.** Consider $J$ groups of data, each $j$-th group consists of $N_j$ data points, a Hierarchical DP Mixture (HDPM) prior is a probability measure over $\Theta^J \times \Sigma_j = 1 N_j$ with the usual product sigma algebra $\Sigma^{N_1} \times \ldots \times \Sigma^{N_J}$ such that for every collection of measurable sets $\{ (S^{(j)}_i) : S_{ji} \in \Sigma, j = 1, \ldots, J, i = 1, \ldots, N_j \}$

\[
\text{HDPM}\left( \{ \theta_{ji} \in S^{(j)}_i : j = 1, \ldots, J, i = 1, \ldots, N_j \} \right| \gamma, \alpha H) = \int \left\{ \prod_{j=1}^J \prod_{i=1}^{N_j} G\left( S^{(j)}_i \right) \text{DP}(G \mid \gamma G_0) \right\} \text{DP}(dG_0 \mid \alpha H)
\]

Using a similar proving strategy for a single group case, we obtain a marginalisation consistency for our proposed extension to HDP. Again, using the same notation described earlier where $\Theta = \Theta_1 \times \Theta_2$, $\Sigma = \Sigma_1 \times \Sigma_2$, $H^*$ is measure over the product space with $H_1$ and $H_2$ being the marginals respectively, then:
Proposition 6.1. Denote by \( \theta_{ji} \) the pair \( \left( \theta_{ji}^{(1)}, \theta_{ji}^{(2)} \right) \), it holds that

\[
HDPM \left( \left\{ \theta_{ji}^{(1)} \in S_{j}^{(j)} : j = 1, \ldots, J, i = 1, \ldots, N_{j} \right\} | \gamma, \alpha H \right) =
HDPM \left( \left\{ \theta_{ji}^{(1)} \in S_{j}^{(i)} \times \Theta_{2} : j = 1..J, i = 1..N_{j} \right\} | \gamma, \alpha H^{*} \right)
\]

for every collection of measurable sets \( \{ (S_{ji}) : S_{ji} \in \Sigma_{1}, j = 1, \ldots, J, i = 1, \ldots, N_{j} \} \).

We note that while this theorem states the marginalisation consistency for our model with the product space of two measures, it can be trivially generalised to any number of base measures using induction.

6.4 Experiments

In this section, we first present our experiment with a synthetic dataset. Then, we describe the StudentLife dataset, which is used in our experiments. Next, we present the discovery of identity–location–time patterns using our proposed model with complete data. We also analyze the discovered patterns at both global and local levels. After that, we present quantitative comparisons with the baselines. Finally, we demonstrate the ability of the PS-HDP model to deal with missing data.

Our experiments were run on a PC equipped with an Intel Xeon E5-2460 CPU (8 cores, 2.6Ghz) and 16GB RAM. We implemented the PS-HDP model using C#. We ran a Gibbs sampling over 500 iterations for posterior inference. On average, each iteration cost around 19 seconds.

6.4.1 Experiments with Synthetic Data

To demonstrate the model properties and its advantages in modelling multiple data sources simultaneously, we experimented with synthetic data where we know the ground truth. We verified the capability of the PS-HDP to identify the appropriate co-patterns existing in the data with multiple sources. We generated 50 groups, each group having 120 data points, and each data point comprising two data channels. All data points \( \left( x_{i}^{(1)}, x_{i}^{(2)} \right) \) were generated from one of \( K = 6 \) co-clusters as shown in Figure 6.2. These 6 co-clusters are mutual combinations of 3 multinomial distributions (top row) and 2 univariate Gaussian distributions (bottom row). The
multinomial distributions were used to generate the first channel $x_i^{(1)}$, while the Gaussian distributions were used to generate the second channel $x_i^{(2)}$. This synthetic data represents a common situation in ubiquitous computing applications where the data channels can be in a continuous or discrete domain. Although we demonstrate the modelling for two channels, our PS-HDP model is generic to model any number of channels which occur in real-world applications.

![Figure 6.2](image1.png)

**Figure 6.2**: A synthetic example of two channels of discrete and continuous data. There are 6 true co-clusters comprising 3 multinomial distributions and 2 Gaussian distributions. Our proposed PS-HDP correctly recovers 6 true clusters.

![Figure 6.3](image2.png)

**Figure 6.3**: The HDP model estimates $K = 3$ topics on the first channel of synthetic data. It cannot recover the 6 true co-patterns as it does not utilise the data from second channel (Gaussian distribution).

We fed this synthetic data to our PS-HDP model which correctly recovered these true clusters and co-patterns as shown in Figure 6.2. In contrast, HDP can only utilise single channel information, thus it failed to learn the co-patterns existing in
6.4. Experiments

the data. Because the multinomial observations from the true clusters 1 and 2, 3 and 4, 5 and 6 were merged, HDP resulted in only $K = 3$ clusters (Figure 6.3). Observing single channel data does not well reflect the whole data. Thus, HDP identified $K = 3$ instead of $K = 6$ which is the true number of clusters.

6.4.2 Discovery of Who–When–Where Patterns with Complete Data

We fed data from the StudentLife dataset (section 5.3.2) to our PS-HDP model, where the Bluetooth and WiFi channels were modelled by multinomial distributions, and the time channel was modelled by Gaussian distributions. The inference of our PS-HDP model resulted in 17 patterns (clusters). Each pattern has three elements - a Gaussian distribution and two multinomial distributions. The Gaussian distribution (parametrised by a mean and a variance) represents the distribution of time at which the pattern occurs. The two multinomial distributions represent the probability of each Bluetooth ID/WiFi ID participating in the pattern. Each distribution of Bluetooth IDs reflects a proximity pattern between users, and each distribution of WiFi IDs reflects a location.

Table 6.1 visualises 7 significant patterns of proximity (who) - location (where) - time (when) discovered by our PS-HDP model. Both Bluetooth IDs and WiFi hotspot IDs are displayed using a tag cloud, where the size of each number reflects the contribution of the corresponding ID to the pattern. As a result, there could be some IDs with very small probability which would not be seen on the plot. A pattern of time is represented by the probability density function of the corresponding Gaussian distribution.

By making a comparison between patterns in Table 6.1, we can find some interesting information for pervasive analysis. For example, we can see that patterns 1 and 4 represent a proximity group (dominated by the user who owns the device corresponding to Bluetooth ID #5), at the same location (Sudikoff building). However, their corresponding timestamps are different; one at early morning (e.g., 5am) and one after working hour (e.g., 7pm), respectively. Another interesting example concerns patterns 14 and 16. These two patterns represent events that happen in similar locations but the proximity groups are slightly different (the only mutual significant user is #2). In our opinion, these two patterns would be hard to discover if one
### Table 6.1: The patterns of proximity (who), location (where) and time (when) discovered by PS-HDP.

<table>
<thead>
<tr>
<th>ID</th>
<th>Proximity</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td><img src="image1.png" alt="Graph" /></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td><img src="image3.png" alt="Graph" /></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td><img src="image5.png" alt="Graph" /></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td><img src="image6.png" alt="Graph" /></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td><img src="image7.png" alt="Graph" /></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td><img src="image8.png" alt="Graph" /></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td><img src="image9.png" alt="Graph" /></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td><img src="image10.png" alt="Graph" /></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td><img src="image11.png" alt="Graph" /></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td><img src="image12.png" alt="Graph" /></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td><img src="image13.png" alt="Graph" /></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td><img src="image14.png" alt="Graph" /></td>
</tr>
</tbody>
</table>
treats each data channel (i.e. Bluetooth and WiFi) independently. This shows the advantage of our model over traditional approaches.

To facilitate the exploration of the discovered patterns, we created an applet\(^1\) which displays patterns in the form of a network (as shown in Figure 6.4). When a user clicks on a node (e.g. a location topic node), it will emphasise the top 5 significant WiFi IDs that contribute the most to this location topic, and hides other nodes. It also connects this location topic node to a corresponding proximity topic node and time node. This applet makes it easier to see which WiFi IDs belong to a specific location topic, or which Bluetooth IDs belong to a specific proximity pattern.

To compare the multi-channel patterns discovered by the PS-HDP model and the single-channel patterns discovered by HDP, we ran HDP for the WiFi channel to discover the location patterns. The HDP model also includes a latent indicator \(z_{ji}\) for each observation \(x_{ji}\), representing the pattern that this observation is assigned, similar to \(z_{ji}\) in Figure 6.1b. We used this latent indicator \(z_{ji}\) to compute a confusion matrix between the patterns discovered by the PS-HDP and the location patterns discovered by HDP. We plot this confusion matrix in Figure 6.5. The cell \((m,n)\) in this matrix is the number of data points that are assigned to cluster \(m\) by PS-HDP and to cluster \(n\) by HDP. For the ease of visualisation, we normalised the matrix to make each row sum to 1. This matrix represents the correlation between the multi-channel contexts and the location contexts. As we can see in the figure, the 5 multi-channel contexts 7, 11, 12, 15, 17 happen at location 7. The usual behavior of PS-HDP is that it runs on multi-channel data in comparison with HDP which runs on single-channel data. The correlating information from other channels can split a cluster of single-channel data into smaller clusters. For example, a student might attend to two different lectures in the same building but with two different groups of classmates. If we engage in pattern discovery on location data, we will only get one pattern – the location of the building. However, if we take advantage of the proximity data from different groups of classmates, it can help to split the data into two smaller groups.

\(^1\)Readers are referred to appendix A for more details about the applet.
Figure 6.4: All 17 discovered patterns are displayed as a network. Each location topic (a green node, prefixed by L) is connected to the 5 WiFi IDs (red nodes) that contribute the most to that location topic. Similarly, each proximity topic (a blue node, prefixed by P) is connected to the 5 Bluetooth IDs that contribute the most to the topic. Corresponding location and proximity topics are linked together. They are also connected to a time node (a lime green node) showing the time of the event.

6.4.3 Multilevel Pattern Analysis

We further analyze multilevel patterns at global (for all participants) and local (for each participant) levels. We aim to discover which patterns are regular for all users (e.g., which particular users often interact with each other, where the locations are,
and when they meet). Similarly, we learn the local regular patterns for individuals (e.g., with whom does this user often interact, at which locations, and when). We ask similar questions for infrequent patterns at the multilevel. To answer these questions, we utilise global weight $\beta$ to analyze the level of global patterns and the mixing proportion $\pi_j$ to analyze the level of local patterns for user $j$.

### 6.4.3.1 Global Pattern Analysis

We rely on $\beta$ to find interesting patterns globally. Figure 6.6 shows the pattern that has the largest $\beta_k$ value (pattern number 5) and the pattern that has a smallest $\beta_k$ value (pattern number 15), as well as the $\beta_k$ values of all 17 patterns.
6.4. Experiments

6.4.3.2 Local Pattern Analysis

After running the PS-HDP model, we obtained a mixing proportion vector for each participant and cluster assignment for each data point. Using this information, we can rebuild patterns for each participant using only the data points scanned by his phone. To help find significant patterns for each participant, we also built a small application which lets the user choose a user (represented by a Bluetooth ID) and the application visualises the two most interesting patterns (i.e. one has the largest and one has the smallest $\beta_k$ value) inferred from the data of that participant.

Furthermore, we computed a similarity matrix between 18 users using the Euclidean distance of a user’s mixture proportions. After this, we fed this similarity matrix into the Affinity Propagation clustering algorithm (Frey and Dueck, 2007) to automatically group users together. We also computed the linear correlation coefficients from the users’ mixture proportions ($\pi_j$ in Figure 6.1b) as shown in Figure 6.7. We can see that users with similar correlation coefficients have been grouped together (e.g. group of users 1, 2, 3, or group of users 14, 15, 16, 17) showing that these users are highly correlated and have similar activities.

6.4.4 Evaluation of Performance

After running the PS-HDP model, each data point was assigned to a cluster through the indicator $z_{ji} = k$. In addition, from the ground truth, the location of each data point can be known. We rely on this information to evaluate the clustering perform-
6.4. Experiments

![User correlogram](image)

Figure 6.7: Correlogram of users’ mixture proportions.

...ance of our approach. To obtain the baselines, we ran some quantitative clustering algorithms including k-means (MacQueen, 1967), non-negative matrix factorization (NNMF) (Lee and Seung, 1999), Gaussian mixture model (GMM) (McLachlan and Peel, 2004), DP-means (Kulis and Jordan, 2012), and Affinity Propagation (AP) (Frey and Dueck, 2007) on the same dataset and use their quantitative clustering results to compare the performance of the PS-HDP. We note that the k-means, NNMF, and GMM methods require a number of cluster $K$ to be specified beforehand, and this $K$ can affect the performance score, while our model can automatically learn the number of patterns from the data. Therefore, to be fair, we ran k-means with different $K$ values and calculated the average performance scores. The range of $K$ was selected around the values of $K$ which were automatically discovered by the DP-means and AP algorithms. Here, we varied $K$ from 2 to 40. Table 6.2 shows the performance scores of these algorithms. Overall, the PS-HDP model achieves better scores in comparison with the other clustering algorithms.
6.4. Experiments

<table>
<thead>
<tr>
<th>Algo.</th>
<th>Clusters</th>
<th>F1</th>
<th>NMI</th>
<th>RI</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means</td>
<td>2-40</td>
<td>.405</td>
<td>.527</td>
<td>.737</td>
<td>.711</td>
</tr>
<tr>
<td>NNMF</td>
<td>2-40</td>
<td>.390</td>
<td>.603</td>
<td>.810</td>
<td>.765</td>
</tr>
<tr>
<td>GMM</td>
<td>2-40</td>
<td>.513</td>
<td>.614</td>
<td>.783</td>
<td>.769</td>
</tr>
<tr>
<td>DP-means</td>
<td>19</td>
<td>.410</td>
<td>.028</td>
<td>.468</td>
<td>.267</td>
</tr>
<tr>
<td>AP</td>
<td>9</td>
<td>.190</td>
<td>.130</td>
<td>.702</td>
<td>.480</td>
</tr>
<tr>
<td>PS-HDP</td>
<td>17</td>
<td>.646</td>
<td>.415</td>
<td>.841</td>
<td>.685</td>
</tr>
</tbody>
</table>

Table 6.2: Performance of different clustering algorithms on the StudentLife dataset.

6.4.5 Discovery of Who–When–Where Patterns with Missing Data

To demonstrate the pattern discovery ability of our framework on data with missing values, we created four settings with differing amounts of missing data from the complete StudentLife dataset (section 5.3.2). More specifically, from the complete dataset, we randomly chose \( m \) data points (with \( m \) repeatedly set to 80%, 60%, 40% and 20% of the total data points) and set their Bluetooth data as missing values. Each of these generated datasets was then fed to the PS-HDP model and performance metrics were calculated (similar to the complete data settings). The experiment results show that the F1-score increases consistently when less missing data occurs (as shown in Table 6.3). In other words, the more data there is, the better the patterns are.

<table>
<thead>
<tr>
<th>Missing</th>
<th>Clusters</th>
<th>F1</th>
<th>NMI</th>
<th>RI</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>80%</td>
<td>18</td>
<td>.317</td>
<td>.553</td>
<td>.770</td>
<td>.759</td>
</tr>
<tr>
<td>60%</td>
<td>19</td>
<td>.331</td>
<td>.593</td>
<td>.784</td>
<td>.799</td>
</tr>
<tr>
<td>40%</td>
<td>16</td>
<td>.338</td>
<td>.588</td>
<td>.781</td>
<td>.788</td>
</tr>
<tr>
<td>20%</td>
<td>17</td>
<td>.398</td>
<td>.619</td>
<td>.796</td>
<td>.808</td>
</tr>
</tbody>
</table>

Table 6.3: Performance of PS-HDP with respect to different amount of missing data.
6.5 Concluding Remarks

We presented a Bayesian nonparametric model, called the Product Space HDP, which is extended from the HDP model by using a richer structure for the base measure being a product-space. Its major strengths and advantages are firstly inherited from the Bayesian nonparametric approach including the ability to automatically grow the model complexity, hence it does not need to know the number of clusters beforehand; and secondly lie in the product-space approach which can deal with multiple heterogeneous data sources and with missing elements. The difference of our approach over existing ones is that it treats data from all sources equally, therefore does not need to specify the content and contexts channels. We applied the proposed model to one of the most fundamental problems in pervasive and ubiquitous computing: learning hidden activities from heterogeneous data collected from multiple sensors. We demonstrated the proposed model with a synthetic dataset as well as with a real-world StudentLife dataset to discover who–when–where patterns. The discovered patterns were then analyzed, visualised, compared and contrasted with the ground truth, and the performance was quantitatively evaluated using standard metrics. The obtained results showed the advantages of the PS-HDP model, including the ability to learn complex hidden patterns with superior performance over popular existing clustering methods.
Chapter 7

Data-Driven Understanding of Human Dynamics with Sequential Bayesian Models

In the previous chapter, we presented the product-space hierarchical Dirichlet processes (PS-HDP) model, a unified Bayesian nonparametric model for dealing with multi-channel heterogeneous hierarchical data. The model was developed through the extension of the base measure of the HDP model being a product-space. The richer base measure allows multiple data channels with different data types to be handled at the same time. Moreover, the model can also handle missing data, which is one of the big challenges and is frequently seen in pervasive applications. Through intensive experiments, we showed that the PS-HDP model can effectively discover latent patterns in both synthetic and real datasets.

Although it has the ability to work with multi-channel complex data, the PS-HDP model is not designed to learn hidden patterns in sequential data directly. It was not designed to model time constraints, which is an important characteristic of sequential data. In this chapter, we develop a new model, further extending our research by combining the strength of the hidden Markov model (HMM) and the multi-channel nonparametric clustering (MCNC) model, to handle sequential multi-channel heterogeneous data.
7.1 Introduction

Recall from chapter 2, a hidden Markov model (HMM) (Rabiner and Juang, 1986) can be considered a special case of a mixture model where the hidden variables, which control the mixture component to be selected for each observation, are related through a Markov process rather than being independent of each other. This generalisation enables HMM to effectively model time series or sequential data, where data points at timestamp \( t \) are dependent (only) on data points at timestamp \( t - 1 \). We refer readers to section 2.2.2.2 for more detail on HMM.

However, the HMM model is based on some strong Markov assumptions which consequently limits the flexibility of HMM. For example, one of the assumptions of HMM is that data is generated by some discrete state variable which can take on one of several values. In other words, the number of hidden states of a specific HMM has to be pre-defined (Beal et al., 2002). In real-world problems, especially in pervasive applications where data continuously arrive in a stream, the number of hidden states is not known, thus it is impossible to specify in advance.

The infinite hidden Markov model (iHMM) (Beal et al., 2002) is a nonparametric generalisation of HMM which is designed to overcome the aforementioned limitation of the HMM. In iHMM, a Dirichlet prior is placed over the hidden states to allow the number of hidden states to grow to an infinite number and to be learned from the data, hence it does not require the number of states to be specified beforehand. However, the iHMM did not gain momentum until the work of Teh et al. (2006) where the hierarchical Dirichlet processes (HDP) model is used as a nonparametric prior for the building block of iHMM (described as the HDP-HMM stochastic process), and a more efficient Gibbs sampler is provided.

Unfortunately, like most of the existing topic models, the iHMM is designed to work with only one kind of data at a time (i.e., discrete or continuous), hence it cannot be applied on multi-channel data. Thus, there is a need for some iHMM extensions so that it is able to handle multi-channel and heterogeneous data. In this chapter, we apply the same principle as that of the MCNC and PS-HDP models to the iHMM model, i.e., making the base measure of the infinite hidden Markov model (iHMM) being a product-space, so that it has the ability to deal with sequential multi-channel data. The new model, named the multi-channel infinite hidden Markov model (MC-iHMM), is demonstrated on both synthetic and real-world heterogeneous...
multi-channel data – the StudentLife (Wang et al., 2014) dataset. The experiment results show that our proposed MC-iHMM model can achieve better performance compared to existing baselines.

Our contributions in this chapter are two-fold. First, we contribute to the theory development in the field by proposing a product-space approach for the \(i\)HMM model with an infinite capacity to handle data channels. The new model has a solid mathematical foundation to ensure its robustness. Furthermore, the model not only handles multi-channel data but also heterogeneous (mixed type) and missing data. Additionally, it can be used for both clustering as well as classification tasks due to its sequential nature. Second, on the application side, to the best of our knowledge, we are the first to use a Bayesian nonparametric (BNP) sequential model to discover the who-when-where co-patterns. This is an important problem in pervasive computing as location, person, and time and activities are the most important bits of information to develop context-aware systems. Moreover, since the proposed model is based on \(i\)HMM, it can handle sequential data as well as it can grow with data complexity. In contrast to parametric models where the number of clusters needs to be known in advance, our MC-\(i\)HMM model can adapt itself to the given data and automatically infer the number of clusters. Thus, it delivers a promising truly online and scalable algorithm for streaming data which is often seen in pervasive computing.

7.2 Bayesian Models for Sequential Data

In this section, we present briefly the \(i\)HMM model (Beal et al., 2002) and its inference method. Specifically, we follow the \(i\)HMM model representation and its inference scheme introduced by Teh et al. (2006), namely a hierarchical Dirichlet processes hidden Markov model (HDP-HMM).

7.2.1 Model Representation for HDP-HMM

The \(i\)HMM, proposed by Beal et al. (2002), is a Bayesian nonparametric version of HMM to overcome the limitations of HMM. In \(i\)HMM, a Dirichlet prior is placed over the hidden states, allowing the number of hidden states to grow to an infinite number and to be learned from data, hence it does not need to specify the number of states beforehand. However, as the stochastic process of \(i\)HMM was not well-defined,
the model was not used widely until the work on hierarchical Dirichlet processes (HDP) by Teh et al. (2006). Specifically, Teh et al. (2006) reformulated iHMM under the view of the HDP framework, in which HDP is used as a nonparametric prior for the building block of iHMM. It is known as a hierarchical Dirichlet processes hidden Markov model (HDP-HMM) stochastic process. Therefore, the unknown number of states in HMM is identified in the same way as HDP. Figure 7.1 shows the stochastic representation of the HDP-HMM model.

![Figure 7.1: The stochastic view of the infinite hidden Markov model.](image)

The stick-breaking representation of HDP-HMM is illustrated in Figure 7.2 in which the parameters are drawn from the following distributions:

\[
\begin{align*}
\beta & \sim \text{GEM} (\gamma) \\
\pi_k & \sim \text{DP} (\alpha, \beta) \\
\phi_k & \sim H \quad k = 1, 2, \ldots, \infty \\
z_t & \sim \pi_{z_{t-1}} \quad t = 1, 2, \ldots, T \\
y_t & \sim F (\phi_{z_t})
\end{align*}
\]

where \( \{y_t\}_{t=1}^T \) is the data observation at time \( t \), \( z_t \) is the latent variable indicating the component \( k \) that generates \( y_t \), \( \pi_k \) is a vector of the mixture proportion of components, and \( \beta \) is a vector of the stick-breaking weights.
7.2.2 Inference for HDP-HMM

In this section, we briefly present Gibbs sampling to perform the inference for the HDP-HMM. In particular, we need to sample the hidden variables including $z_t$, $\beta$, $\alpha$ and $\gamma$. While sampling the stick-breaking parameter $\beta$ and hyperparameters $\alpha, \gamma$ are the same as for HDP, sampling of $z_t$ needs to be re-calculated. Specifically, the hidden variable $z_t$ is sampled using the following conditional probability:

$$p(z_t = k \mid z_{-t}, y, \beta, H) \propto p(y_t \mid z_t = k, z_{-t}, y_{-t}, H) \times p(z_t = k \mid z_{-t}, \alpha, \beta) \quad (7.1)$$

The first term is the likelihood of the observation $y_t$ given component $\phi_{zt}$. In other words, this likelihood can be expressed as $\int_{\phi_k} p(y_t \mid z_t = k, \phi_k) p(\phi_k \mid y_{-t}, z_{-t}, H) d\phi_k$ which can easily be analyzed under the conjugate property. The second term in Equation 7.1 is simply the Chinese restaurant process of transition. Let $n_{ij}$ be the number of transitions from state $i$ to state $j$ and $n_{*j}$ be the number of all transitions to state $j$. Similarly, let $n_{i*}$ be the number of all transitions departing from state $i$. 

Figure 7.2: The stick-breaking view of the infinite hidden Markov model.
7.2. Bayesian Models for Sequential Data

The CRP likelihood under the Markov property can be analyzed as:

\[ p(z_t \mid z_{-t}, \alpha, \beta) \propto p(z_t = k \mid z_{t-1}, \alpha, \beta) \times p(z_t = k \mid z_{t+1}, \alpha, \beta) \]

from state t-1 to state t

\times p(z_t = k \mid z_{t+1}, \alpha, \beta)

from state t to state t+1

We then have four cases to compute this probability:

\[ p(z_t \mid z_{-t}, \alpha, \beta) \propto \begin{cases} 
(n_{z_{t-1}, k} + \alpha \beta_k)^{n_{k, z_{t+1}} + \alpha \beta_{z_{t+1}}} & k \leq K, k \neq z_{t-1} \\
(n_{z_{t-1}, k} + \alpha \beta_k)^{n_{k, z_{t+1}} + 1 + \alpha \beta_{z_{t+1}}} & z_{t-1} = k = z_{t+1} \\
(n_{z_{t-1}, k} + \alpha \beta_k)^{n_{k, z_{t+1}} + \alpha \beta_{z_{t+1}}} & z_{t-1} = k \neq z_{t+1} \\
\alpha \beta_{z_{t+1}}^\text{new} & k = K + 1 
\end{cases} \]

7.2.3 Applications of iHMM

Since its birth, the iHMM model and its variants have been successfully applied to numerous applications. Goldwater et al. (2006) used HDP-HMM to segment an audio stream into a sequence of words. They proposed a statistical approach to word segmentation based on the HDP-HMM where the latent states of HMM correspond to words. Fox et al. (2008) demonstrated iHMM as an effective method for speaker diarisation. Speaker diarisation is the problem of segmenting an audio recording into time intervals associated with individual speakers. The posterior inference of the iHMM yields an estimation of the number of speakers participating in a meeting and a diarisation of the audio stream. In the computer vision field, Stepleton et al. (2009) presented a block diagonal iHMM to video gesture classification tasks and a musical theme labelling task. This model is a generalisation of the iHMM of which the transitions between hidden states are in a nearly block-diagonal structure. In identifying such a structure, the model classifies, or partitions, the data sequence according to behaviors in an unsupervised manner. Nguyen et al. (2012) made use of iHMM to video segmentation for the problem of anomaly detection. More recently, Nguyen et al. (2017) extended iHMM for discriminative settings to enhance the dissimilarity among hidden states.
7.3 Multi-Channel Infinite Hidden Markov Model

To deal with sequential multi-channel data, we propose the multi-channel infinite hidden Markov model (MC-iHMM) model as described in the following. For simplicity, we assume that our data are comprised of two heterogeneous channels, hence each data point $y_t$ is a 2-tuple $(y^{(1)}_t, y^{(2)}_t)$ where $y^{(1)}_t$ and $y^{(2)}_t$ can be completely different in the data types (e.g., one is continuous and the other is discrete). We employ a product-space measure $H_1 \times H_2$ to generate the observations $(y^{(1)}_t, y^{(2)}_t)$. Each base measure $H_1$ or $H_2$, through their corresponding likelihood functions $F_1$ and $F_2$, is responsible for generating parameters to explain $y^{(1)}_t$ and $y^{(2)}_t$ respectively.

From the generative view (a.k.a stochastic view), for each time $t$, we draw a sample $G \sim \text{DP}(\lambda, H_1 \times H_2)$, which is now a pair $(\theta^{(1)}_t, \theta^{(2)}_t)$. Then, we generate data as $y^{(1)}_t \sim F_1(\cdot|\theta^{(1)}_t)$ and $y^{(2)}_t \sim F_2(\cdot|\theta^{(2)}_t)$ respectively. From the stick-breaking view, our stick-breaking representation now becomes $G = \sum_{k=1}^{\infty} \pi_k \delta_{\psi_k}$, where each atom $\psi_k$ is now a 2-tuple $\psi_k = (\phi_k, \lambda_k)$ where $\phi_k$ (s) are i.i.d drawn from $H_1$ and $\lambda_k$ (s) from $H_2$. The construction of stick-breaking weights $\pi_k$ (s) remains the same as described by Sethuraman (1994). Figure 7.3 and 7.4 show the graphical model of MC-iHMM from the stochastic view and from the stick-breaking view, respectively.

![Figure 7.3: The stochastic view of MC-iHMM model.](image)

For posterior inference, we also use collapsed Gibbs sampling as described by Liu (1994). Thus, from the stick-breaking view of the MC-iHMM model, we can see that the variables which need to be sampled are $\gamma, \beta, \alpha$ and $z_t$. Of these, the stick-breaking parameter $\beta$ and hyperparameters $\alpha, \gamma$ are sampled in the same way as for
HDP described by Teh et al. (2006). Here, we present how to sample the variables $z_t$. Specifically, the hidden variable $z_t$ is sampled using the following conditional probability:

$$p(z_t = k \mid z_{-t}, y, \beta, H) \propto p(y_t \mid z_t = k, z_{-t}, y_{-t}, H) \times p(z_t = k \mid z_{-t}, \alpha, \beta)$$ \hspace{1cm} (7.2)

The first term is the likelihood of the observation $y_t$ given the component $\psi_{z_t} = (\phi_{z_t}, \lambda_{z_t})$. In other words, this likelihood can be expressed as:

$$\int_{\phi_k, \lambda_k} p(y_t \mid z_t = k, \phi_k, \lambda_k) p(\phi_k \mid y_{-t}, z_{-t}, H_1) p(\lambda_k \mid y_{-t}, z_{-t}, H_2) d\phi_k d\lambda_k$$

which can be easily analyzed under the conjugate property. The second term in Equation 7.2 is simply the Chinese restaurant process of transition. Let $n_{ij}$ be the number of transitions from state $i$ to state $j$ and let $n_{ej}$ be the number of all transitions to state $j$. Similarly, let $n_{ei}$ be the number of all transitions departing from state $i$. The CRP likelihood under the Markov property can be analyzed as:
7.4. Experiments

\[ p(z_t \mid z_{-t}, \alpha, \beta) \propto p(z_t = k \mid z_{t-1}, \alpha, \beta) \times p(z_t = k \mid z_{t+1}, \alpha, \beta). \]

We then have four cases to compute this probability:

\[
p(z_t \mid z_{-t}, \alpha, \beta) \propto \begin{cases} 
(n_{z_{t-1}, k} + \alpha \beta_k) \frac{n_{k, z_{t+1}} + \alpha \beta_{z_{t+1}}}{n_{k+1} + \alpha} & k \leq K, k \neq z_{t-1} \\
(n_{z_{t-1}, k} + \alpha \beta_k) \frac{n_{k, z_{t+1}} + \alpha \beta_{z_{t+1}}}{n_{k+1} + \alpha} & z_{t-1} = k = z_{t+1} \\
(n_{z_{t-1}, k} + \alpha \beta_k) \frac{n_{k, z_{t+1}} + \alpha \beta_{z_{t+1}}}{n_{k+1} + \alpha} & z_{t-1} = k \neq z_{t+1} \\
\alpha \beta_{k_{\text{new}}} \beta_{z_{t+1}} & k = K + 1 
\end{cases}
\]

where \( \alpha, \beta \) are sampled following the method of Teh et al. (2006).

7.4 Experiments

To demonstrate the capacity of our proposed model MC-\(i\)HMM, we run intensive experiments on the model using synthetic data as well as real-world data. The experiments on the synthetic data help us verify the expected properties of our model on clean data with clearly known patterns, while the experiments on the StudentLife dataset demonstrate the applicability of our model to real applications.

7.4.1 Experiments with Synthetic Dataset

In this section, we run our proposed model on a synthetic dataset to demonstrate the model properties and advantages. More specifically, we want to show the capability of our MC-\(i\)HMM model in identifying the appropriate co-patterns existing in data with multiple sources and of mixed types. Hence, our synthetic data were generated for these purposes. We describe the process of data generation as follows.

To demonstrate that our model can simultaneously handle data from multiple sources and of mixed types, we generated data in two channels, in which the data in the first channel were discrete and the data in the second channel were continuous. Additionally, we assumed that data can be grouped in some clusters. Hence, we created \( K \) atoms \( \{\psi_1, \ldots, \psi_K\} \) and used them to generate data points. Each atom \( \psi_k \) is a 2-tuple \( \psi_k = (\phi_k, \lambda_k) \) where \( \phi_k \) is a multinomial distribution and \( \lambda_k \) is a Gaussian distribution. Moreover, to demonstrate its ability to learn better clusters by exploiting information from all data channels, we designed these \( K \) atoms in a
way such that some of them have similar multinomial distributions, but different in Gaussian distributions. In our experiments, for visualisation purposes, each multinomial vector of size $35 \times 1$ is reshaped to a $7 \times 5$ matrix representing a dot matrix of Latin characters. Figure 7.5 shows the true atoms that we use to generate our synthetic data.

We also define a transition matrix $A$ of size $K \times K$, where each element $A_{ij}$ defines the probability of transitions from cluster $i$ to cluster $j$. These $K$ atoms and the transition matrix were then used to generate $N$ data points $\{x_1, \ldots, x_N\}$, each data point $x_i$ is also a 2-tuple $(x_i^{(1)}, x_i^{(2)})$ as follows. For each data index $i$, we drew cluster index $z_i$ from transition matrix $A$ and the previous cluster index $z_{i-1}$ (assumed that $z_0 = 1$). Then, we used atom $\psi_{z_i}$ to generate $x_i$ where $x_i^{(1)}$ is i.i.d drawn from $\phi_{z_i}$ and $x_i^{(2)}$ is i.i.d drawn from $\lambda_{z_i}$. More specifically, we generated $N = 3000$ data points using $K = 6$ true atoms, as shown in Figure 7.5, and the true transition matrix $A$ shown in Figure 7.6.

![Figure 7.5](image)

**Figure 7.5:** The true $K = 6$ patterns used to generate 2-channel data. Data of the first channel are generated from multinomial distributions and data of the second one are generated from univariate Gaussian distributions. Note that some distributions are replicated. There are only four unique multinomial distributions and three unique univariate Gaussian distributions.

We applied our MC-iHMM model on the above synthetic data to verify the capability of MC-iHMM to identify the appropriate co-patterns existing in the multi-channel heterogeneous data. As seen in Figure 7.7 and 7.8, our model correctly recovers six true patterns (Figure 7.7) as well as the true transition matrix (Figure 7.8).
In contrast, as the iHMM model can only handle single-channel data, we cannot feed all the generated synthetic data to iHMM. Instead, we had to choose one of two channels as the input to iHMM. Figure 7.9 and 7.10 show the patterns and the transition matrix learned by iHMM on the data of the first channel (discrete data with multinomial distribution). The iHMM model was unable to learn the six true patterns, instead it discovered only four patterns corresponding to the four unique letters (Figure 7.9). This phenomenon is easy to understand, since the iHMM model did not use the information from the data of the second channel which is the key to discovering patterns in a finer manner. Without this channel, the multinomial observations from true clusters 2 and 4 were generated from the same multinomial distribution (letter A), hence they were merged into one cluster. In the same way, multinomial observations from true clusters 5 and 6 were also merged together. These merges of patterns are also reflected in the learned transition matrix (Figure 7.10).

We then evaluated quantitatively the clustering results using popular metrics for clustering analysis including purity, normalised mutual information (NMI), pair-counting F-measure (F1-score), and rand index (RI). The evaluation results are presented in Table 7.1. The MC-iHMM model achieves perfect scores as it correctly groups all data points, whereas the iHMM model obtains average scores as it can not correctly detect the six true clusters, but only four clusters corresponding to the four unique multinomial distributions.

Finally, we studied the learning process of the MC-iHMM and iHMM model. Figure 7.11 shows the average likelihood of the two models over 50 iterations. We can see...
that the two models quickly converge on the synthetic data. However, the learning process of the MC-iHMM model is more robust than that of the iHMM model.

### 7.4.2 Experiments with Real-world Dataset

The StudentLife dataset is a real-world pervasive dataset collected from the smartphones of 48 students at Dartmouth College over a 10-week spring term in 2013. We refer readers to section 5.3.2 for more details about the dataset and how we processed it for our experiments. In this section, we present some more visualisations about the locations where the scans took place. As location labels (such as a building name or a residential area) are used as ground truth for our quantitative evaluations, it is important to understand these location labels and their uncertainties.
7.4. Experiments

Figure 7.8: Transition matrix learned by the MC-iHMM is similar to the true transition matrix.

Figure 7.9: Patterns learned by iHMM on the first data channel (multinomial distribution). iHMM discovered four patterns instead of six true patterns. It merged true patterns number 2 and 4 as well as 5 and 6.

We first investigate the relation between WiFi hotspots and the locations. In the StudentLife dataset, each location is identified by a unique ID, namely locID, and each WiFi hotspot is identified by its MAC address, which is then converted to a unique WiFiID. It is common knowledge that one location could have many WiFi hotspots. However, in an ideal situation, each WiFi hotspot should belong to one and only one location. Unfortunately, after investigating the WiFi scans of the StudentLife dataset, we found that many WiFi hotspots were labelled with more than one location. Figure 7.12 shows the histogram count of the number of unique WiFiIDs against the number of unique locID assigned to a WiFiID. We can see that there are over 3,000 WiFiIDs which have been assigned to two different locIDs, over
1.600 WiFiIDs which have been assigned to three different locIDs, and over 1.000 WiFiIDs which each of them being assigned to four different locIDs. This indicates that many locations overlap, and the labelling process is not consistent. This led to multiple labels being assigned to one WiFiID. Therefore, we must consider uncertainty when using locIDs as ground truth to quantitatively evaluate the performance of the location discovery via WiFiIDs.

The StudentLife dataset also contains the GPS data. We used this GPS information to visualise the positions of the scans on a map retrieved from Google Maps (Figure 7.13). Note that the GPS information was only used for visualisation purposes and was not used in our experiments. As can be seen from Figure 7.13, in addition to the overlapping of locations, there is also fragmentation of some locations such as

Figure 7.10: Transition matrix learned by iHMM.

Figure 7.11: Average likelihood of iHMM and MC-iHMM models over 50 iterations.
7.4. Experiments

Figure 7.12: Histogram count of WiFi ID vs. number of location ID assigned to a specific WiFi ID.

“North-Main” and “East-Wheelock”. This is because the labels are assigned to street names where WiFi hotspots are located.

To obtain a deeper understanding of the location trace of each user which could potentially reveal the daily routines of the user, we visualised the scans of each user on a grid. Since the StudentLife dataset was collected over a 10-week duration, we map the location on a grid of 10 rows and 504 columns (7 days $\times$ 24 hours $\times$ 3 intervals of 20 minutes). Each row represents a week and each column represents an interval of 20 minutes. For each corresponding 20-minute interval, we associated it with a location label where the user spends the most of time during the interval. We used different colors to encode location labels. Figure 7.14 shows the grid maps of three selected users: u00, u10 and u17. We found that the data collected by each user varied significantly. Data from user u00 were mostly collected during the working hours of week days and no data was collected from him at night time or on weekends. In contrast, data collected from user u10 and u17 are much more complete. From the grid maps of these two users, we can easily determine the mobility routines of user u10, while routines of user u17 are not clear.
7.5 Concluding Remarks

In this chapter, we presented a novel model, named the multi-channel infinite hidden Markov model (MC-iHMM), to discover user’s activity patterns in sequential multi-channel heterogeneous data. We first presented the theory to build the foundation and support for our MC-iHMM model. Then, we conducted experiments on synthetic data as well as real-world data. The experiment results on the synthetic data showed the strength and advances of our model over the iHMM model in terms of handling sequential mixed-type data. We also applied the model on the StudentLife dataset, a real-world pervasive dataset containing multiple types of data. We achieved comparable results on this dataset compared to some popular baseline algorithms.

Despite achieving promising results on real-world data, we still have some work to do on our MC-iHMM model. In particular, we would like to investigate further the quantitative results of our model to see why it did not achieve better results, as
expected on the StudentLife dataset. In addition, we also want to conduct more qualitative analysis to show the advances of our model in a more intuitive manner. This will form the basis of our future work and interesting results are anticipated.

Figure 7.14: The grid maps of ground truth locations of the three selected users.
Pervasive computing has become parts of our daily lives, bringing both benefits and challenges. In particular, data collected from pervasive devices are potentially useful for a wide range of applications such as healthcare, context-aware systems, or understanding human dynamics. However, how to discover useful information from pervasive data still remains a great challenge to machine learning and data mining research. This thesis has developed machine learning methods to discover hidden patterns and co-patterns from complex heterogeneous data acquired in a pervasive setting. The content of this thesis strengthens the connection between two growing fields in computer science: machine learning and pervasive computing. In particular, it has systematically explored the use of Bayesian nonparametric models to learn patterns and co-patterns from pervasive signals. To the best of our knowledge, this thesis is the first attempt to learn co-patterns from heterogeneous, multi-channel, and sequential pervasive data using a Bayesian nonparametric approach. In this chapter, we provide a brief summary of the thesis and then discuss our future perspectives on the work presented in the thesis.

### 8.1 Summary

The content of this thesis can be summarised as follows.

Chapter 3 contributed to the literature on pervasive healthcare by investigating machine learning methods for the purpose of anomaly detection from electrocardiogram
(ECG) signals collected in the wild from low-cost wearable devices. In particular, we focused on the problem of handling noisy signals collected from wearable ECG devices. Noisy signals make the detection of ECG fiducial points as well as the segmentation of ECG signals less accurate. As a consequence, the accuracy of abnormal heartbeat detection is also diminished as it relies on the detection of ECG fiducial points and segments. To tackle this problem, we proposed a temporal normalisation method of ECG segments which could provide a set of robust features even with low quality ECG signals collected from wearable devices. Moreover, because our normalisation method is implemented with a temporal window, its training process can be done online and can adapt to situational changes. We demonstrated our methods on electrocardiography (ECG) data recorded from low-cost Shimmer wearable ECG devices and data from the MIT-BIH Arrhythmia database. The experiment results showed that our approach achieved a higher F1-score than rule-based methods used by clinical physicians.

Chapter 4 addressed the problem of understanding human dynamics from pervasive signals. In particularly, we discovered the significant locations and daily routines of individuals from WiFi signals captured by their smartphones. We recognised that WiFi data have a high time coverage in comparison to other location data, such as GPS, hence WiFi data can provide more information about the travel routines of individuals. Our approach employed the Affinity Propagation (AP) algorithm to group near-by WiFi access points into groups. Based on the assumption that WiFi access points are immobile, we considered each group a significant location. This approach did not use any geographical information or fingerprint data. We chose the AP algorithm for data clustering since this algorithm can automatically discover the number of clusters from the data, thus it can adapt well to pervasive data. We also analysed the sequence of significant visited places of each individual to learn his/her daily routines. Knowing the daily routines of individuals is core to understanding human dynamics. We demonstrated our approach on the Nokia Mobile Data Challenge (MDC) dataset. The experiment results showed that we could gain insights into human dynamics including the discovery of typical location patterns and the outlier days when the user had unusual location trajectories.

In chapter 5, we tackled the challenges of dealing with multi-channel and heterogeneous data as well as missing data. We proposed a novel model, termed the multi-channel nonparametric clustering (MCNC) model. This nonparametric (BNP) model is able to extract high-order latent co-patterns that occur across different data
8.1. Summary

channels. This latent co-pattern discovery ability is enable through the machinery extension of current BNP models by using a richer product-space base measure distribution. This richer base measure allows the MCNC model to be able to simultaneously explore data from multiple sources and extract hidden co-patterns, even if data have missing elements. Additionally, the MCNC model can also automatically discover the space of latent patterns from the data, thus it does not require the number of patterns to be specified in advance. We demonstrate the properties of our MCNC model on a synthetic dataset as well as on the StudentLife dataset, a real-world pervasive dataset collected from smartphones, to discover who-when-where patterns. These patterns provide the co-occurrence information on people, time and location, which is a key input for human dynamic understanding and context-awareness applications.

In chapter 6, we extended the MCNC model to address a special case of pervasive data. We presented the product-space hierarchical Dirichlet processes (PS-HDP) model to deal with multi-channel heterogeneous data which can be grouped hierarchically. The PS-HDP model combines the strengths of the MCNC model to simultaneously extract latent patterns from multi-channel heterogeneous data and the strengths of the hierarchical Dirichlet processes (HDP) model to group data in a hierarchical fashion.

In chapter 7, we presented the multi-channel infinite hidden Markov model (MC-iHMM) to dynamically discover patterns from multi-channel sequential heterogeneous data. It combines the strengths of the MCNC model and the infinite hidden Markov model (iHMM) in learning patterns from sequential data. In short, the core features of the PS-HDP and the MC-iHMM models are based on the product-space approach to deal with multi-channel heterogeneous data. However, their model architectures were designed to adapt to two different clustering problems: hierarchical data clustering and sequential data clustering. We demonstrated the two models on both synthetic and real-world pervasive datasets. We evaluated their clustering performances using standard metrics including purity, F1-score, normalised mutual information (NMI), and rand-index (RI). The results showed that our models achieved comparable clustering performance to those of baseline methods. We also provided interactive tools to visualise the who-when-where patterns as a network so that it can facilitate the exploration of our experiment results.
8.2 Future Work

Overall, this thesis has addressed the key challenges in the discovery of hidden patterns and co-patterns from heterogeneous, complex and dynamic pervasive signals. In this section, we discuss some ideas and possible extensions of the presented work.

In our work on arrhythmia detection from wearable ECG signals (chapter 3), we focused on the problem of dealing with noisy data. Due to the time limit and the lack of volunteer participants, we only collected ECG signals from members of our lab, none of whom were likely to have had serious cardiac problems. For more thorough experiments, we can undertake a new data collection on a larger scale by employing more participants with a wide range of cardiac problems. Additionally, another research direction is how to combine signals from motion sensors to enhance arrhythmia detection. For example, we can use accelerometer data to identify the activities which the subject is doing (e.g., running or sitting). This activity information can help reduce the number of false positives in arrhythmia detection.

One drawback of our work on significant location discovery (chapter 4) is the fragmentation of locations. The reason for this might be that we did not used time information for clustering. Thus in future work, we could add the time information into the clustering process. We could also examine the graph clustering approaches to detect groups of WiFi access points.

Our work on the product-space approach (chapter 5–7) addressed the challenges of pervasive signals. More specifically, we proposed the MCNC model to discover latent patterns from multi-channel heterogeneous data (with or without missing elements). Then, we extended this model to address two clustering problems: hierarchical data clustering and sequential data clustering. The main drawback of our models is that they rely on collapsed Gibbs sampling for posterior inference. However, collapsed Gibbs sampling is well-known as a computationally expensive method, thus it is inappropriate for large-scale data. To make our models scalable, we could replace Gibbs sampling by variational inference or decayed MCMC sampling methods, which usually run faster than Gibbs sampling. In particular, these methods can compute posterior inference in an incremental progress, thus they are also suitable for the online discovery of latent patterns. By employing the aforementioned inference methods, we can apply our methods proposed in chapters 5, 6, 7 to large-scale data such as the Nokia MDC dataset.
Appendix A

The Interactive *Who-When-Where* Applet

The multi-channel nonparametric clustering (MCNC) model and the product-space hierarchical Dirichlet process (PS-HDP) model are designed to discover co-patterns from complex heterogeneous data. To assist users explore the co-patterns discovered by our models, we create an applet which visualises these co-patterns as a network. This applet allows users to interact with each individual co-pattern and to see the relationship between components in this co-pattern.

In this appendix, we use the *time-location-proximity* (a.k.a *who-when-where*) co-patterns discovered by our PS-HDP model in chapter 6 to demonstrate our applet. Recall that the PS-HDP model discovered 17 co-patterns from the StudentLife dataset. These patterns are visualised as a network using Gephi\(^1\). Furthermore, we use the JavaScript GEXF Viewer for Gephi\(^2\) as the user interface to allow users to interact and explore the *who-when-where* co-patterns. Figure A.1 shows an example of the applet displaying the aforementioned patterns. In this figure, each individual time, location, proximity topic as well as each individual Bluetooth ID and WiFi ID are represented as nodes with different colors corresponding to different node types. In more detail, proximity topics (prefixed by *P*) are blue nodes, location topics (prefixed by *L*) are red nodes, time topics are mud-green nodes, Bluetooth IDs

---

\(^1\)https://gephi.org/

\(^2\)https://github.com/raphv/gexf-js. This JavaScript Viewer works best on Firefox browser.
Figure A.1: The network of who-when-where co-patterns from the StudentLife dataset.

are green nodes and WiFi IDs are pink nodes. The size of each location or proximity node represents the mixture proportion of the corresponding topic, whereas the size of each time topic represents the standard deviation of its Gaussian topic distribution. For example, we can see that the node sizes of P5, L5 are bigger than those of P4, L4. This implies that the mixture proportion of the who-when-where topic 5 is larger than that of topic 4. Furthermore, the size of an edge which connects a Bluetooth/WiFi IDs to a proximity/location node represents the contribution of this Bluetooth/WiFi IDs to the corresponding proximity/location topic.

When a user clicks on a proximity, location or time topic node, the nodes in the same co-pattern are also highlighted. Figure A.2 shows an example where location topic P4 is chosen. In this figure, Bluetooth IDs 7, 5, 10, 64 and 14 are highlighted, and the edge which connects P4 and Bluetooth ID 5 is the thickest. This means that proximity topic P4 represents a group of people which includes Bluetooth IDs 7, 5, 10, 64 and 14, and of these, Bluetooth ID 5 presents the most frequently in
Figure A.2: Relation of a proximity topic and its corresponding location and time topics.

This group. Moreover, P4 also connects to time topic 19:48, which has a small size, implying that this group of people (7, 5, 10, 64 and 14) usually meet each other around 19:48 with high confidence.

Proximity topic P4 is associated with location topic L4. To explore L4, users can click on it and the significant WiFi IDs of the topic L4 will be highlighted. In Figure A.3, we can see that WiFi hotspot IDs 1, 2, 10, 13 and 14 are highlighted, and the edge to WiFi ID 2 has the largest size. This means that L4 represents a location where WiFi hotspot IDs 1, 2, 10, 13 and 14 are located, and is more likely to be close to WiFi hotspot 2.

This demonstration is packed in a compressed ZIP file and can be downloaded using the link: https://goo.gl/82ak6L. After downloading the ZIP file, extract its content and open the file index.html file using the Firefox browser.
**Figure A.3:** Relation of a location topic and its corresponding proximity and time topics.
Bibliography


S. Goldwater, T. L. Griffiths, and M. Johnson. Contextual dependencies in unsu-


N. T. Nguyen, D. Phung, H. Bui, and S. Venkatesh. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In Proc. of


P. Nurmi and S. Bhattacharya. Identifying meaningful places: the nonparametric way. In J. Indulska, D. J. Patterson, T. Rodden, and M. Ott, editors, Proc. of


D. Price. How to read an electrocardiogram (ECG), May 2010.


Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.
Title: MCNC: Multi-Channel Nonparametric Clustering from heterogeneous data
Conference Proceedings: Pattern Recognition (ICPR), 2016 23rd International Conference on
Author: Thanh-Binh Nguyen
Publisher: IEEE
Date: Dec. 2016
Copyright © 2016, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
Title: Unsupervised inference of significant locations from WiFi data for understanding human dynamics

Author: Thanh-Binh Nguyen, et al.

Publication: Proceedings

Publisher: Association for Computing Machinery, Inc.

Date: Nov 25, 2014

Copyright © 2014, Association for Computing Machinery, Inc.

Order Completed

Thank you for your order.

This Agreement between Thanh-Binh Nguyen ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") consists of your license details and the terms and conditions provided by Association for Computing Machinery, Inc. and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

printable details

License Number 4350521243786
License date May 15, 2018
Licensed Content Publisher Association for Computing Machinery, Inc.
Licensed Content Publication Proceedings
Licensed Content Title Unsupervised inference of significant locations from WiFi data for understanding human dynamics
Licensed Content Author Thanh-Binh Nguyen, et al
Licensed Content Date Nov 25, 2014
Type of Use Thesis/Dissertation
Requestor type Author of this ACM article
Is reuse in the author's own new work? No
Format Print and electronic
Portion Full article
Will you be translating? No
Order reference number
Title of your thesis/dissertation Making Sense of Pervasive Signals: a Machine Learning Approach
Expected completion date Dec 2017
Estimated size (pages) 180
Attachment
Requestor Location 41/22-28 Canterbury Street

Flemington, Victoria 3031
Australia
Attn: Nguyen

Billing Type Credit Card
Credit card info Visa ending in 9020
Credit card expiration 03/2019
Total 10.62 AUD

ORDER MORE  
CLOSE WINDOW

Copyright © 2018 Copyright Clearance Center, Inc. All Rights Reserved. Privacy statement. Terms and Conditions. Comments? We would like to hear from you. E-mail us at customercare@copyright.com
Title: Individualized arrhythmia detection with ECG signals from wearable devices

Conference Proceedings: Data Science and Advanced Analytics (DSAA), 2014 International Conference on

Author: Thanh-Binh Nguyen

Publisher: IEEE

Date: Oct. 2014

Copyright © 2014, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.