USER BEHAVIOUR MODELLING, RECOGNITION AND ANALYTICS IN PERVASIVE COMPUTING

GUANQING LIANG

Ph.D

The Hong Kong Polytechnic University

2016

The Hong Kong Polytechnic University

Department of Computing

User Behaviour Modelling, Recognition and Analytics in Pervasive Computing

Guanqing Liang

A thesis submitted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

August 2015

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

(Signed)

Guanqing Liang (Name of Student)

Abstract

Recent years have witnessed the unprecedented growth of the adoption of sensor-rich devices such as smartphone and smartwatch, along with the large scale deployment of a variety of sensor networks in ambient environments. With the help of those ambient sensors, a large amount of users' digital traces can be collected, which opens up new opportunities for user behaviour modeling, recognition and analytics. User behaviour modeling, recognition and analytics are one of the key components in pervasive computing, which underpins a variety of significant applications: smart healthcare, business intelligence, context-aware applications, etc. Although a significant amount of research effort has been devoted in this topic, how to effectively model, recognize and analyze user behaviour still remains an open problem.

In this thesis, we focus on two major research topics. First topic is how to accurately model and recognize certain categories of user behaviours based on ambient sensor data. In particular, we focus on the study of three kinds of behaviours which play critical roles in both physical and psychological heath, including: sitting posture, eating habit and social interaction. Second topic is how to identify and exploit the correlation between user behaviours and other user states such as emotion. Specifically, we conduct correlation analytics between mobility and social circle, stress and sitting posture, and exploit the correlation relationship to build up new recognition model. The details are as follows.

Firstly, we aim to classify people's sitting posture based on pressure sensors-embedded

seat cushion. Due to intrusiveness, high cost or low generalization accuracy, current solutions for siting posture recognition are impractical. In this work, we design Postureware, an accurate, low-cost and non-intrusive sitting posture recognition system. In particular, Postureware incorporates very thin pressure sensors to offer non-intrusive experience, an effective sensor placement solution to reduce cost, a set of user-invariant features and an ensemble learning classifier to improve generalization ability. The results show that Postureware can achieve 99.6% ten-fold cross validation accuracy and 84.7% generalization accuracy only with 10 sensors. In addition, we further evaluate the system utility by developing three applications, including unhealthy sitting posture monitoring, sitting posture-based game interface and wheelchair control.

Secondly, we study the problem of recognizing people's eating behaviour using off-the-shelf smartwatch and smartphone. However, very few works have been developed for long-term eating behaviour monitoring by means of a noninvasive platform. In particular, we exploit the accelerometer of smartwatch to derive user's eating behaviour, including: eating schedule, food cuisine and food item. Besides, we leverage the collaboration between smartwatch and smartphone to reduce the energy consumption of smartwatch, and thus enabling long-term monitoring. More specifically, we propose a context-aware data collection method to conserve energy, a novel set of accelerometer features that are able to capture key characteristics of eating behaviour patterns, and a light-weight decision tree-based classification algorithm. We evaluate our approach using real-world traces and the experimental results demonstrate our work is able to monitor individual's eating behaviour in a non-invasive and energy-efficient manner.

Thirdly, we aim to model and recognize social activity based on the sensor data collected from smartphone. Most of the existing works in social activity recognition are based on the patterns of individual user such as location pattern, vocal pattern, etc. However, we observe that social activity is associated with certain group, which inherently exhibits the patterns with respect to multiple users. In this work, we introduce the concept of social circle, which reveals the behaviour pattern associated with mul-

tiple users in social activities. Here, a social circle refers to a set of users frequently gathering to conduct certain social activities. Based on the social circle concept, we present CircleSense, an accurate and efficient smartphone-based system for social activity recognition. In particular, social circle is extracted from the social proximity information obtained by Bluetooth device discovery. To further improve the system accuracy, we apply metric learning technique to extract social circle from social proximity information. To evaluate the system performance, we conduct extensive experiment based on the dataset collected in real world from 16 subjects. The experiment result shows that CircleSense outperforms the existing methods in terms of the recognition accuracy.

In addition to user behaviour modelling and recognition, we study the problem of correlation analytics between user behaviours and other user states such as emotion. We find that there exists correlation between human mobility and social circle, as well as stress and sitting behaviour. By leveraging the correlation relationships, we improve the accuracy of human mobility prediction and stress measurement. In particular, we study the problem of human mobility prediction based on social context. We first conduct correlation analytics on 10-day Wi-Fi traces collected from 111K devices in a large shopping mall. We found that dwell time of repeat visitor exhibits a low degree of variation. Interestingly, visitor dwell time is positively correlated with the size of social group during the visit. By exploiting the above findings, this work presents an accurate user dwell time prediction model that incorporates time and social context, dwell time and leave time history. Evaluation results show that the proposed model is able to provide high accuracy of predicting user dwell time and outperform the baseline methods. Last but not least, we aim to measure stress based on seating pressure distribution on the chair. In particular, we collect seating pressure data from 15 participants using a seat cushion which is deployed with 20 pressure sensors. Through correlation analysis, we identify a number of seating pressure features that are associated with stress, including: average seating pressure, pressure imbalance, etc. Based on the associated

features, we build up a stress detection framework to classify whether participants are stressed or not. The result show that the stress detection framework can achieve 86% accuracy using kNN classifier.

Publications

Journal Paper

1. **Guanqing Liang**, Jiannong Cao and Jiaqi Wen, "How You Sit Reveals How You Feel: Non-intrusive Stress Detection Based on Seating Pressure", submitted to *IEEE Transactions on Affective Computing* (2015).

2. **Guanqing Liang**, Jiannong Cao, Lei Yang and Xuefeng Liu, "CircleSense: Accurate and Energy-Efficient Social Activity Recognition based on Smartphones", Submitted to *Pervasive and Mobile Computing* (2015).

3. Lei Yang, Jiannong Cao, **Guanqing Liang** and Xu Han, "Cost-aware Service Placement and Load Dispatching in Mobile Cloud Systems", to appear in *IEEE Transactions on Computers* (2015).

4. **Guanqing Liang** and Jiannong Cao, "Social context-aware middleware: A survey", *Pervasive and Mobile Computing* 17 (2015): 207-219.

Conference Paper

1. **Guanqing Liang**, Jiannong Cao and Xuefeng Liu, "Smart World: A Better World", *12th IEEE International Conference on Ubiquitous Intelligence and Computing* (UIC 2015), Beijing, China, August 10-14, 2015. [Best Video Award]

2. **Guanqing Liang**, Jiannong Cao, Xuefeng Liu and Xu Han, "Cushionware: a practical sitting posture-based interaction system", *CHI Interactivity* 2014. ACM, Toronto, Canada, 591-594.

3. **Guanqing Liang**, Jiannong Cao and Weiping Zhu, "CircleSense: A pervasive computing system for recognizing social activities", *IEEE International Conference on Pervasive Computing and Communications* (PerCom 2013), San Diego, USA. March, 2013.

Acknowledgements

I thought I would settle down after I got my bachelor degree back in 2011. But Kelcy and Juan had set great examples for me that professor does make a difference in students' life, which inspired me to pursue my PhD degree so as to become a professor in university.

First and foremost, I would like to express my deepest appreciation to my supervisor, Prof. Jiannong Cao for his mentoring and guidance. It is he who encourages and challenges me to be a critical thinker, an effective writer and a logical presenter. More importantly, he transforms me to be a more proactive, responsible and mature man.

I would like to thank my co-supervisor Prof. Grace Ngai, Prof. Rocky Chang and Prof. Alvin Chan for giving me many invaluable advices in both research and life. I would also like to thank Dr. Walter S. L. Fung. He not only challenges me to stay focused, but also inspires me to become an interesting person with great passion for life.

My sincere thanks also goes to Prof. Tarek Abdelzaher for giving me an opportunity to do research as a visiting scholar at University of Illinois at Urbana-Champaign from Oct 2014 to May 2015. I want to express my gratitude to my colleagues: Shen Li, Shaohan Hu, Shiguang Wang and Hongwei Wang. Besides, I would like to thank my housemates: Peter, Derek, Alex, Michael, Mark, Trevor and Marcus.

I would like to thank my colleagues and buddies in Hong Kong including Xuefeng Liu, Peng Guo, Chao Ma, Junbing Liang, Zongjian He, Tao Li, Lei Yang, Joanna Siebert,

Gang Yao, Weiping Zhu, Jun Ma, Yaguang Huangfu and other members of IMCL lab. We get through ups and downs together; we support one another; we share our life with each other.

Last but not least, I would like to thank my parents for their unconditional love and support. They not only give me life, but also freedom! I also want to thank my girlfriend Yang Zhu, who always supports me and encourages me to pursue the truth and whatever is genuine.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Pervasive Computing

The vision of pervasive computing described by Mark Weiser in the seminal paper [Wei99] is "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Specifically, a pervasive computing environment is characterized as a place which is saturated with computing and communication capability, yet gracefully integrated with users. In essence, the disappearance of technologies in pervasive computing indicate the minimal user consciousness.

To raise the minimal user consciousness, it is of great importance to understand and leverage context to anticipate users' needs and to act in advance. Context can be divided into four categories: computing context, user context, physical context and time context [CK00]. In particular, user context includes user's profile, location and behaviour etc. In this thesis, we focus on understanding user behaviour, which is one of most important user context. Specifically, user behaviour understanding encompasses the modelling, recognition and analytics of user behaviour.

Understanding user behaviour opens up a lot of novel opportunities for healthcare, business intelligence and internet of things applications, which is drawing increasing attention from both academy and industry. For example, according to world health organization, more than 1.9 billion adults are overweight and over 600 million of them are obese by 2014 [WHO15]. Obesity mainly results from unhealthy life style: unhealthy eating behaviour and sedentary behaviour. Recognizing and monitoring eating and sitting behaviour not only helps raise people's awareness, but also offer behaviour logs to medical staff for diagnosis. Furthermore, analysing the association among different behaviours can provide insight for behaviour intervention and behaviour modeling.

1.2 Understanding User Behaviour

User behaviour understanding encompasses the modelling, recognition and analytics of user behaviour. User behaviour modelling and recognition focuses on accurate recognition of user behaviour based on a carefully designed behaviour model. User behaviour analytics focuses on identifying the patterns of behaviour and the association between behaviours.

1.2.1 User Behaviour Modelling and Recognition

The objective of user behaviour modelling and recognition is to first model user behaviour using a set of features and then recognize the user behaviour with reasoning techniques.

How to model user behaviour depends on the understanding of the behaviour. Each behaviour can be characterized using different set of features. For instance, social activity could be featured using location and time. However, spatial or temporal features might not be adequate to represent social activities. Therefore, social activity can be

modeled based on social circle and time. Here, social circle refers to a set of people that appear together frequently in a social activity. User behaviour modelling is critical to the performance of behaviour recognition. If the user behaviour can be modelled in a robust and effective way, then use behaviour recognition can achieve high accuracy with very simple reasoning techniques.

Based on the behaviour model, behaviour recognition uses inference techniques to derive high-level behaviour context from the low-level sensor data. Generally, behaviour recognition can be performed by two techniques: data-based technique and knowledge-based technique. Data-based technique recognizes user behaviour via learning of dataset. Data-based technique can be further divided into generative learning, discriminative learning and heuristic approach. Knowledge-based technique takes advantages of domain knowledge to infer user's behaviour. Knowledge-based approach consists of three categories, including mining-based, logic-based and ontology-based.

Both data-based and knowledge-based techniques have pros and cons. Data-based technique is good at handling uncertainty and temporal information. However, it requires large amount of datasets for learning, and does not support scalability and reusability. Rather than counting on data, knowledge-based technique is based on rules or patterns extracted from domain experience. As a result, it can be easy to start without dataset, and supports reusability and scalability. But it cannot handle uncertainty and temporal information. In this thesis, we adopt data-based techniques as our recognition techniques, due to their advantages on dealing with uncertainty.

1.2.2 Behaviour Analytics

User behaviour analytics focuses on identifying the patterns of behaviour and the association between behaviours. On one hand, user behaviour analytics aim to find out the patterns of behaviour in time, space or frequency domain. On the other hand, use

behaviour analytics conduct association analysis to measure the correlation between different behaviours. In this thesis, we focus on correlation analytics among behaviours and other user states such as stress. Take stress detection as an example. First, we conduct correlation analytics between stress and seating behaviour. Then we identify some seating pressure features that can reveal user's stress state. Finally, we construct a stress classification model that can automatically identify whether user is stressed or not. Ultimately, behaviour analytics is used to refine the behaviour model and recognition process.

1.3 Motivation of the Thesis

We argue that a practical user behaviour modelling, recognition and analytics system should meet the following requirements in terms of model effectiveness, system cost and user experience, etc.

- **High-accuracy**. First and foremost, the system should be able to accurately infer user behaviour based on the sensor data. The accuracy of system depends on the feature extraction and model selection. Thus, how to extract user-invariant features and how to select appropriate model require further investigation.
- **Non-intrusive**. Last but not last, the system should be non-intrusive to let users feel comfortable. Otherwise, the system with bad user experience will be discarded.
- **Energy-efficient**. Since most of the user behaviour recognition systems are based on energy-limited mobile platforms such as smartphone and smartwatch, they are expected to be energy-efficient to enable long-term monitoring.
- **Low-cost**. The system should be affordable, so that most of the people can benefit from it.

- **Real-time**. Some system need to be real-time to support certain applications.

However, challenges arise due to the conflicts between the above requirements. For example, high accuracy usually indicates high energy consumption and high cost. Thus, to design a practical user behaviour modelling, recognition and analytics system, we need to strike the balance among the system accuracy, energy consumption and system cost.

First, we focus on modelling and recognising three particular kinds of behaviour: sitting posture, eating habit and social activity. Here, we aim to optimise the balance between accuracy, cost and energy consumption.

In sitting posture recognition problem, we focus on arbitrating the conflicts between accuracy and cost. Current solutions based on wearable sensors or camera for siting posture recognition, however, are impractical due to intrusiveness, high cost or low accuracy. To achieve a non-intrusive and privacy-preserving solution, researchers have deployed pressure sensors on the chair to infer sitting posture [XLH+11][TSP01]. But the main limitation of these solutions are the high cost (around 3000 USD), since they rely on high-fidelity pressure sensor array with more than two hundred sensors [Tek]. Note that paper [MKF+07] presents an idea to use 19 pressure sensors for sitting posture recognition. However, it only achieves 78% accuracy. The objective of this work is to enable a practical sitting posture recognition technique that is accurate, low-cost and non-intrusive and real-time. We introduce Postureware, a sitting posture recognition system that can identify ten common sitting postures accurately in real-time using ten pressure sensors that are deployed inside a seat cushion.

We aim to resolve the conflicts between system accuracy and energy-consumption in eating behaviour inference. Monitoring eating behaviour has great potential for helping people controlling weights and improving health conditions, and thus drawing increasing attention from both industry and research communities. However, very few works have been developed for long-term eating behaviour monitoring by means of a non-

invasive platform. In particular, we exploit the accelerometer of smartwatch to derive user's eating behaviour, including: eating schedule, food cuisine and food item. Besides, we leverage the collaboration between smartwatch and smartphone to reduce the energy consumption of smartwatch, and thus enabling long-term monitoring. More specifically, we propose a context-aware data collection method to conserve energy, a novel set of accelerometer features that are able to capture key characteristics of eating behaviour patterns, and a light-weight decision tree-based classification algorithm. We evaluate our approach using real-world traces and the experimental results demonstrate our work is able to monitor individual's eating behaviour in a non-invasive and energy-efficient manner.

We study the problem of social activity recognition and focus on improving the system accuracy. A fair amount of research efforts have been devoted to activity recognition based on various sensors embedded in smartphone. Some studies utilize accelerometer to measure user's locomotion state and then derives his/her physical activities based on locomotion patterns [ZLC+08][MLEC07]. In addition to locomotion pattern, location pattern has been extracted to infer certain kinds of social activities. In papers [EP06][ZY11], authors use smartphone to obtain the location of users based on Wi-Fi/GPS, and then identify the social activities based on the context of location. However, the aforementioned approaches only consider the behaviour patterns of an individual user. We observe that social activity is associated with a community, which inherently exhibit the behaviour patterns with respect to multiple users rather than individual users. Thus, in this work, we propose a social circle-based social activity recognition model.

Second, we study the correlation analytics between behaviours and other user states, and then exploit the correlation relationship to build up better models. By exploiting the correlation relationship between behaviours and other user states, we can improve the system accuracy and non-intrusiveness. In this part, we aim to optimise the accuracy and non-intrusiveness.

We aim to improve the accuracy of user dwell time prediction model by exploiting social context. Note that predicting the dwell time of users based on some contextual information is not novel (e.g., arrival time, past dwell time). Works [MSCN13][VDN11] [DGP12], have all studied such cases. However, none of the previous works have tried to quantify the influence of fundamental factors upon user dwell time. It is worth emphasising that quantifying the influences of factors towards user dwell time is significant. On one hand, it advances the understanding of user movement behaviour. On the other hand, it provides guidelines for the design of dwell time prediction model. Moreover, social context which does impact user mobility behaviour, has not been considered into the previous dwell time prediction models. Thus, we conduct the correlation analytics and propose social circle-based user dwell time prediction model.

We are interested to explore the possibility of detecting stress based on the seating pressure distribution on a chair, in the attempt to strike the balance between system accuracy and non-intrusiveness. Several technologies have been developed to measure stress, including surveys, physiological signal measurement (blood pressure [VvDdG00], heart rate variability [DNG+00], skin conductance [HMP11] [SAS+10], cortisol [DK04] [vEBNS96]). However, these methods are intrusive, which require the cognitive attention of the users. We argue that an ideal stress detection system for long-term monitoring should be unobtrusive, without posing additional stress upon people. A less intrusive approach to measure stress is analysing the user behaviours such as typing pattern [HPRC14] and mobile phone usage [LLLZ13], which correlate with stress. Our idea is motivated by the embodied theory of cognition [Cla97], which indicates that affective states of people are manifested in their posture channels. Specifically, this work seeks to measure stress based on seating pressure patterns.

In this thesis, we will elaborate the aforementioned problems in detail and design the solutions to tackle them.

1.4 Contributions of the Thesis

The contributions of this thesis mainly lie in modeling, recognition and analytics of user behaviour in pervasive environment. Fig 1.1 shows the framework of this thesis. In particular, the main body of the thesis consists of two parts: user behaviour modeling and recognition, correlation analytics. The first part focuses on how to accurately model and recognize certain categories of user behaviours based on ambient sensor data. In particular, we study three kinds of behaviours which play critical roles in both physical and psychological heath, including: sitting posture, eating habit and social interaction. Second part aims to identify and exploit the correlation between user behaviours and other user states such as emotion. Specifically, we conduct correlation analytics between mobility and social circle, stress and sitting posture, and exploit the correlation relationship to build up new recognition model. In the following, we will present the contributions of this thesis in detail.

1.4.1 Contributions in User Behaviour Modeling and Recognition

In user behaviour modeling and recognition, we focus on three kinds of behaviours including: sitting posture, eating habit and social interaction. 1) Current solutions for siting posture recognition, however, are impractical due to intrusiveness, high cost or low generalization accuracy. In this work, we design Postureware, an accurate, low-cost and non-intrusive sitting posture recognition system. We design and implement a non-intrusive system for sitting posture recognition, which can accurately recognize sitting posture in real-time (8Hz) with low cost (150USD). We then propose an information-theoretic sensor placement solution, enabling the system to achieve the same recognition accuracy using much fewer number of sensors. Specifically, the system can recognize 10 categories of sitting postures with 99.6% ten-fold cross validation accuracy with 10 sensors. Furthermore, we design an accurate sitting posture recognition

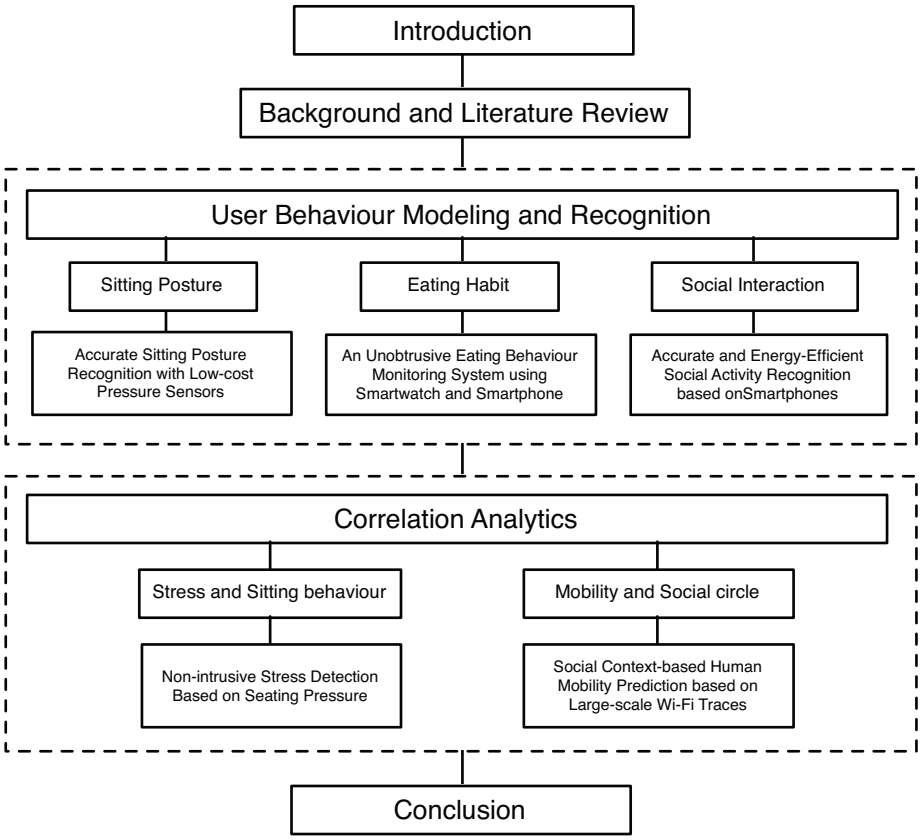


Figure 1.1: The outline of the thesis

model, which incorporates pre-processing, user-invariant feature extraction and AdaBoost classification, being able to resolve the challenges posed by user diversity and thus improving generalization accuracy. The system can achieve 84.7% generalization accuracy with 10 sensors. Finally, we implement a prototype system and develop two applications to evaluate the system utility. Unhealthy sitting posture recognition application can achieve 99.9% precision accuracy from the offline dataset. In second application, more than 80% of the users can finish the racing game smoothly using sitting posture.

2) We study the problem of monitoring an individual's eating behaviour using off-the-shelf smartwatch and smartphone. However, very few works have been developed for long-term eating behaviour monitoring by means of a noninvasive platform. The primary contributions of this paper are as follows. We first present a context-aware data

collection method to conserve energy based on the collaboration between smartwatch and smartphone. We then introduce a new set of accelerometer features that can capture key characteristics of eating behaviours. To further reduce the energy consumption of the system, we present a light-weight classification algorithm for eating event classification. We evaluate our approach using real-world traces and the experimental results demonstrate our work is able to monitor individual's eating behaviour in a non-invasive and energy-efficient manner.

3) Most of the existing work in social activity recognition are based on the patterns of individual user such as location pattern, vocal pattern, etc. However, we observe that social activity is associated with a community, which inherently exhibits the patterns with respect to multiple users. In this work, we introduce the concept social circle to identify a distinctive behaviour pattern of social activities. Compare with other patterns extracted from individual user, social circle is able to characterize social activities more accurately. Based on social circle, we develop a practical smartphone-based system called CircleSense for recognition of a generic categories of social activities. Finally, we evaluate the system with a 16-participant dataset which is collected from the deployed android phones. The experimental results demonstrate that CircleSense outperforms the existing methods.

1.4.2 Contributions in Behaviour Correlation Analytics

In the second part of the framework, we study the problem of correlation analytics among behaviours. In particular, we find that there exists correlation between human mobility and social circle, as well as stress and sitting behaviour. Furthermore, by leveraging the correlation relationships, we improve the accuracy of human mobility prediction and stress measurement.

First, we study the problem of user dwell time prediction by exploiting social context. We design StayPredictor, a system to predict user dwell time at a shopping mall based

on Wi-Fi traces. Our solution is evaluated on 10-day WiFi traces collected from 111K devices, and being able to achieve 32.6% relative error. We propose a correlation analytics framework to quantify the impact of factors towards user dwell time. Specifically, this framework can measure the linear and non-linear dependency relationship between dwell time and other factors. To our knowledge, this is the first work to identify and quantify the impact of social context towards user dwell time. We believe that the findings could provide new insights for human mobility modelling. Moreover, we present an ensemble method for dwell time prediction. By incorporating multiple models that are built from different context information, the proposed ensemble method is able to achieve higher prediction accuracy over single models.

Next, We are interested to explore the possibility of detecting stress by analysing the seating pressure distribution on the chair. In particular, we collect seating pressure data from 15 participants using a seat cushion which is deployed with 20 pressure sensors. Through correlation analysis, we identify a set of effective seating pressure features in both temporal and spatial domain that are associated with stress, including: average seating pressure, pressure imbalance, etc. Based on the associated features, we build up a stress detection framework to classify whether participants are stressed or not. The result show that the stress detection framework can achieve 86% accuracy using kNN classifier.

1.5 Organization of the Thesis

The structure of this thesis is shown in Fig.1.1. Chapter 1 presents the introduction of the thesis. Background and related work is presented in Chapter 2. The main body of the thesis is covered from Chapter 3 to Chapter 7. In particular, the main body can be divided into two parts: user behaviour recognition and correlation analytics.

In the first part, we present our work in user behaviour recognition. In Chapter 3,

we study the problem of sitting posture recognition based on pressure sensors. We propose an accurate, low-cost and non-intrusive solution, Postureware, which incorporates very thin pressure sensors, an effective sensor placement solution, a set of user-invariant features and an ensemble learning classifier. In Chapter 4, we study the problem of eating behaviour recognition using smartwatch and smartphone. In particular, we design a context-aware data collection method to conserve energy, a novel set of accelerometer features that are able to capture key characteristics of eating behaviour patterns, and a light-weight decision tree-based classification algorithm. We study the problem of social activity recognition in Chapter 5. We introduce the concept of social circle, which reveals the behaviour pattern associated with multiple users in social activities. Based on the social circle concept, we present CircleSense, an accurate and efficient smartphone-based system for social activity recognition.

In the second part, we present our work in behaviour correlation analytics. In Chapter 6, we study the correlation between human mobility and social circle. We found there exists correlation between user dwell time and the size of social group during the visit. By exploiting the above findings, this work presents an accurate user dwell time prediction model that incorporates time and social context, dwell time and leave time history.

We examine the correlation between stress and sitting posture patterns in Chapter 7. Through correlation analysis, we identify a number of seating pressure features that are associated with stress, including: average seating pressure, pressure imbalance, etc. Based on the associated features, we build up a stress detection framework to classify whether participants are stressed or not.

Finally, Chapter 8 concludes the thesis and discusses the future work.

Chapter 2

Literature Review

In this chapter, we present the literatures review on user behaviour modeling, recognition and analytics. In user behaviour modeling and recognition, we focus on three user behaviours: sitting posture, eating habit and social activity. In the following, we present the state of the art in sitting posture recognition in Section 2.1, eating behaviour monitoring in Section 2.2 and activity recognition in Section 2.3. Next, we focus on correlation analytics between behaviours and other user states such as stress and mobility. The related works of human mobility analytics and stress detection are presented in Section 2.4 and Section 2.5 respectively.

2.1 Sitting Posture Recognition

Sitting posture recognition has received significant attention due to the miniaturization of sensors and the a wide spectrum of applications including: healthcare, human-computer interaction, etc. In the following, we focus on discussing how different sensors have been applied to detect user's sitting postures.

Wearable Sensor-based. Wearable sensor has been used to detect user's sitting

posture. In [KBA+07], the acceleration sensors are attached to the human body to recognize a set of posture including sitting, walking and standing. The acceleration data are collected and then the sitting posture can be identified using threshold-based decision rule. In another research work [SP13], Dunne *et al.* examine the use of clothing augmented with plastic optimal fibre to monitor the seated spinal posture. The proposed method is to first collect inertial data from fibre sensor and apply J48 decision tree learning algorithm to build up the recognition model. However, wearable sensor-based solution exhibits two drawbacks. First, the system is prone to error due to the sensor position drift. Second, it introduces intrusiveness and bring discomfort to users. In contrast, Postureware leverages pressure sensors to collect data associated with sitting postures, which are very thin (0.208 mm thick) and enable non-intrusive user experience.

Vision-based. Vision-based solution depends on visual sensors for sitting posture recognition. Authors in [BBT05] present the study to recognize human postures based on video sequences. Their approach combines the 2-D and 3-D model such that posture can be inferred independent of the camera viewpoint. Alternatively, some researchers in [Cam11] turn to use depth camera (Kinect) for sitting posture recognition. The main idea is to track the human skeleton first using depth camera, and then recognize the sitting posture based on the relative positions among shoulder, hip and knee. However, vision-based approach is easily compromised by the lighting level and raises serious privacy concerns. Furthermore, it requires line of sight condition, which may not applicable to some scenarios due to user's mobility. Postureware, however, can work in non-line-of-sight scenarios, and yet preserve user's privacy.

Pressure Sensor-based. This category of works focuses on exploring pressure sensors to detect user's sitting posture. Most of the research works [XLH+11] [KKNT08] [TSP01] [LA06] [FMGK12] [MAST10] rely on high-fidelity pressure sensor. Wenyao *et al.*[XLH+11] develop a system called e-cushion which mainly tackle the issues caused by sensor crosstalk. They first conduct data-preprocessing to mitigate the effect of

crosstalk, and apply dynamic time warping technique to classify different sitting posture. Paper [ZMT03] focuses on tackling the issues posed by offset, rotation and scaling of pressure distribution due to users' sitting habit and size. A data-preprocessing method is proposed to crop the posture area and resize the pressure distribution into a standard metric. A lot of research efforts have also been devoted to explore a variety of features to represent the sitting posture, including: Principle Component Analysis (PCA) [TSP01]; mean value, center value, peak value, the percentage of contact area and contour shape [LA06]; sum of sensor values, number of active sensors, longitudinal, lateral center of pressure, variance of sensor value [FMGK12]. However, above mentioned works rely on expensive high-fidelity sensor array and thus impeding the widespread adoption. The only attempt which we are aware of that uses a few number of pressure sensors (19 sensors) to perform sitting posture recognition is made by Bilge *et al.* in [MKF+07]. Nevertheless, this system only achieves a (ten-fold cross-validated) classification accuracy of 78%, which is not adequate to support many applications.

In summary, most of previous work either use intrusive wearable/visual sensors or relies on expensive high-resolution pressure sensor array for sitting posture recognition. Different from existing research, the proposed Postureware is a sitting posture recognition system that is accurate, low-cost and non-intrusive. Specifically, Postureware is based on a few number of pressure sensors and able to achieve high accuracy with respect to a diversity of users in terms of weight, sitting habit, yet providing comfortable and privacy-preserving user experience.

2.2 Eating Behaviour Monitoring

Eating behaviour monitoring can be considered as a subfield of human activity analysis. Eating behaviour monitoring has a lot of potential to help people lose weight and keep healthy, which is drawing increasing attention from both industry and research

communities.

To perform eating behaviour monitoring, the simplest way is dietary recall method, which relies on the users to manually provide eating behaviour information by the end of each day. However, this approach suffers from several limitations. First, it is very burdensome to users, since they are required to consciously record what they eat each day. Second, it is erroneous due to the fact that some people like the elderly fail to recall the exact food they have eaten. Therefore, dietary recall method is not feasible for long-term eating behaviour monitoring.

To enable long-term eating behaviour monitoring, both industry and research community have been seeking ways to automatically monitor people's eating behaviour by exploiting various kinds of sensors. One straightforward method is to design a specialised utensil which user will use for meals. HAPIfork [for14] is an electronic fork to monitor and track eating habits. Specifically, it is able to measure the eating speed and trigger the alerts once user are eating too fast, since eating too fast gives rise to severe negative effects such as weight gain and digestive problems. However, the main limitation of HAPIfork is that it requires the user to use it during each meal. First, this is very inconvenient, since the user needs to bring it every day. Second, fork may not be a suitable utensil in some cuisines such as eastern cuisine. Azusa et al. [KLT+14] design and implement a sensing fork to track the eating behaviour of children. The sensing fork contains several sensors: three-electrode conductive probe, six-axis motion sensor and single pixel RGB colour sensor. In particular, the sensing fork is able to detect the eating actions: at-rest, holding, poking and biting. Besides, a number of preselected foods can be classified with the help of colour sensor. However, it exhibits the same limitations of HAPIfork.

Rather than monitoring the utensil, researchers turn to track the wrist motion of users to infer eating behaviour. Dong et al. [DHM09] proposes a system called BitCounter to count the bite of food taken. The system is based on a gyroscope sensor embedded in a wrist-worn device to collect signals associated with user's eating behaviour.

Through analysing the motion signal of the user's wrist, BitCounter can achieve 91% accuracy in detecting the bites of food taken. But there are several limitations of this work. First, the device requires the user to turn on the system during eating, which brings about inconvenience. Second, it is very energy-consuming since it collects data from gyroscope. Third, this work does not consider cuisine and food item detection. In [DSW+14], authors proposes a system to classify the eating and non-eating activities. The system is based on the accelerometer and gyroscope of a smartphone which is placed on wrist. Although fitness trackers such as fitbit [fit14] are wrist-worn devices and have been very popular, it is only limited to count steps, measure calories burned and etc.

In addition to wrist motion signal, acoustic signal has also been exploited to derive eating behaviour. Paper [PWF12] leverages acoustical signals collected by ear microphone to infer food intake activities and classify different kinds of food. More recently, a crowdfunding project BitBite [Bit14] proposes the similar idea to track and analyze eating habits with a smart ear clip. BitBite makes use of a microphone and additional sensors to track the eating activities based on the incoming sounds. BitBite is able to quantify the chewing rates, chewing qualities and eating schedule, etc. However, it requires the user to wear the device whenever they have meals, which is quite obtrusive. In [AKT07], authors present a system to infer eating and drinking activities as well as food category. However, the system requires users to wear an ear microphone, a collar-worn sensors and four inertial sensors on both upper and lower arms, which is very cumbersome and invasive.

Some research works have also utilised sensors such as piezoelectric sensors and camera for eating behaviour monitoring. Paper [KAS14] proposes a smart necklace which is based on an embedded piezoelectric sensor to estimate the volume of food intake. Furthermore, the system is capable of classifying three different kinds of food: sandwich, chips and water. Authors in [KY13] proposes a mobile food recognition system based on smartphone. The system is able to estimate calorie and nutritious of

foods based on food image taken by the smartphone camera. However, this system requires the user to take the picture of food each time.

Different from previous work, we propose iEat, the first non-invasive system that is based on smartwatch and smartphone to automatically record people's eating behaviour, including: eating schedule, food cuisine and food item. In particular, iEat is able to perform continuous and accurate eating behaviour monitoring in an energy-efficient manner.

2.3 Activity Recognition

2.3.1 Phone-based Activity Recognition

In general, existing phone-based activity recognition models can fall into three main categories: location-based model, motion-based model and hybrid model.

Location-based model. The location-based model attempts to infer the activity based on location pattern. In the Reality mining Project [EP06], Eagle et al. makes use of the GSM data obtained by mobile phones to determine users' three simple activity states: home, work or elsewhere. Some researchers [ZY11] propose the WiFi-based activity recognition model, using WiFi access points as location signatures to train the recognition model. However, the location-based model merely considers location and temporal patterns, but fails to recognize some activities that are held in unplanned and impromptu places. Furthermore, in practice, it is not always possible to have dense WiFi access points deployment for accurate indoor localization.

Motion-based model. Significant research effort have also been devoted to utilize the locomotion pattern captured by accelerometer embedded in the mobile phone for social activity recognition. In [MLEC07][MLF+08], Emiliano et al. leverages the accelerometer data to infer users' four simple physical activities: sitting, standing, walking or running.

In the paper [KWM11], the author targets a larger set of activities including walking, jogging, climbing stairs, sitting and standing. Users' transportation activity recognition: bike, bus, walk or driving has also been investigated in another study. Paper [ZLC+08] studies the use of GPS trace to recognize user's transportation activity. They applies the supervised learning approach to train the recognition model based on the traces. In paper [MLEC07], authors builds up the transportation activity recognition model using phones' GSM signal. However, the motion-based model is susceptible to the noise interference and varies between different users, thus making it difficult to establish a general recognition model. Moreover, the method tends to capture coarse activities, which makes it difficult to recognize complex activities such as social activities with varying locomotion state.

Hybrid model. In this model, hybrid collection of sensors including GPS, microphone, accelerometer are used [LXL+11] to obtain location, vocal and locomotion pattern to recognize a set of activities. The activities are comprised of both simple activities such as walk, run, stationary and complex activities such as meeting, studying, exercising, socializing. Noticeably, their work only covers a small set of social activities and is easily compromised by environmental noise. Moreover, the model is energy-consuming that makes it unamenable to operate on smartphones with limited battery capacity.

Although some patterns of individual user including location or vocal patterns have shown their potential in recognizing complex activities including meeting, classes, exercising. Different from previous work, our approach relies on social circle to extract the social patterns of multiple users involved in a generic set of social activities and takes advantage of temporal pattern of social activities, to recognize a representative set of social activities, without any infrastructure support.

2.3.2 Social Context Recognition

Social context modelling and recognition [LCLH14] has drawn a lot of attention in ubiquitous computing community. In general, social context refers to a set of information that characterizes multiple users such as social tie, social group or group dynamics. By exploiting social context information, a variety of social context-aware applications are made possible and have the potential to create a tremendous amount of economic and social value. Social context recognition study can be roughly divided into three categories: social tie inference [MCY+14] [YPT+11], group detection [SLJ+14] and group context inference [BGA+08] [RKL+12].

Social tie inference infers the social link between a pair of users based on similarity measurement in predefined metrics such as location [MCY+14], interest [ZMH+11]. Group detection focuses on clustering users. Paper [SLJ+14] presents GruMon, a group detection system by fusing location data and smartphone sensor data from the accelerometer, compass and barometer. They focus on identifying which individuals are traveling together without identifying their relationships. In particular, they use the correlations between the sensor and semantic data of different individuals to determine if they belongs to the same group.

After group detection, different groups can be identified. Then group context management models and recognizes the context that associates with a group such as group activities, interaction and group dynamics. T. Gu et al. [GWC+11] studies the activities of multiple users using a wireless body sensor network, including watching TV, making pasta, etc. They exploit a discriminative knowledge pattern which indicates significant changes among activity classes to conduct activity recogniser. In paper [LMH+13], Youngki Lee et al. propose SocioPhone, a face-to-face interaction monitoring platform based on multi-phone sensor fusion. In particular, their work identifies a number of meta-linguistic contexts of conversation, such as turn-takings, prosodic features, a dominant participant, and pace. In paper [LC13], authors present SocialWeaver, a

smartphone-based system which is able to perform conversation clustering and construct conversation networks among the users. The main idea is to conduct speaker classification by collaboratively sharing information within the proximity group.

This work focuses on the recognition of social activity which is a particular kind of social context associated with a group of people.

2.4 Human Mobility Analytics

Behaviour Analytics from Wi-Fi Probes Traces. Recently, customer intelligence based on Wi-Fi probe data analytics is drawing a substantial amount of attention, as it has great potential to uncover customer insights. A lot of retailers show great interests in understanding the visitor behaviour, their shopping patterns, and store performance metrics through analysing the collected Wi-Fi probe traces. In particular, using statistical techniques, the overall foot traffic, location of people within a store, dwell times and more can be obtained. However, existing commercial solutions to Wi-Fi probe mining [Euc14] [Tur13] focus on descriptive analytics rather than predictive analytics. Some researchers in academic community have analysing user behaviours and social statuses from Wi-Fi probe data. Paper [ME12] studies the problem of user spatio-temporal trajectory estimation, given the probe data collected from a set of sparsely deployed monitors. Researchers in [QZLS13] explores Wi-Fi probe dat to identify different user activities. Marco et al. [BEM+13] discovers the social structure of a large crowd and its socioeconomic status from smartphone probe information, including: topological properties of the social networks, homophily and social influence in vendor adoption and demographics of brand penetration, etc. Potential social relationship can also be identified by leveraging the WiFi probes [CMC+12]. Different from above works, the objective of this work is to understand and predict user dwell time in a place.

Human Mobility Prediction. Predicting human mobility has been a hot topic since the

location data of people becomes available due to the high penetration rate of mobile phones. Specifically, human mobility prediction mainly focus on studying three problems: where will user go next [SKJH06]; how long will user stay in a location [SMM+11]; how likely will a user stay in location given a certain time [KB11]. Our focus is on the second problem. Long Vu et al. [VDN11] propose a framework called Jyotish to predictive human movement based on Wi-Fi/Bluetooth traces. In particular, to predict the user duration in a location, Jyotish merely relies on the duration distribution obtained from the historical data. Researchers in [DGP12] present a probabilistic model for human mobility prediction. Specifically, the model for duration prediction incorporates the historical duration distribution and the leave time distribution. More recently, a framework for dwell prediction is designed in paper [MSCN13] by leveraging sensor data in smartphone. This framework requires the mobile devices to periodically upload the sensor data to the access points such as accelerometer, compass and light. Based on the collected sensor readings, the dwell prediction can be predicted using machine learning algorithm. Note that previous predictive models for dwell time are evaluated on the dataset collected mainly from campus students. Thus, those dataset may not reflect the real-world human movement.

Different from previous works, our work proposes a correlation analytics framework to quantify the influences of fundamental factors upon user dwell time, which is very critical to the performance of predictive model. Besides, our work is the first to identify and quantify the impact of social context towards user dwell time. Furthermore, our proposed model is evaluated on a 10-day Wi-Fi traces collected from 111K devices.

2.5 Stress Detection

This section provides an overview of previous methods for stress measurement and the related research on posture-based affection detection.

2.5.1 Stress Measurement

The most straightforward method of stress measurement is based on surveys. For instance, perceived stress scale (PSS) [CKM83] is developed to measure the degree to which situations someone feels stressful. The daily stress inventory [BWJR87] can measure the stress experienced by people by counting the number and intensity of stressful events. However, survey-based methods are very subjective, and demand the cognitive attention of users.

An alternative method to assess stress is to measure the physiological signals that related to stress, including: heart rate variability, skin conductance, temperature, respiration, blood pressure. However, physiological-based method is intrusive, which requires people carry the sensors all the time, which may not appropriate for long-term monitoring. Voice is also considered as a signal that reveals the stress states of people. Hong Lu et al. [LFR+12] presents a system that can detect stress of 14 participants with the accuracies of 81% and 76% for indoor and outdoor environments, respectively. In particular, eight acoustic features for stress have been extracted such as pitch deviation, spectral centroid and MFCCs, etc. The main drawback of this approach is that it poses privacy issue, since it requires recording the human speech.

A less intrusive approach is to measure stress through analysing the behaviours that correlate with the stress. Javier et al. [HPRC14] uses pressure-sensitive keyboard and a capacitive mouse to distinguish between stressful and relaxed conditions in a laboratory setting. The study shows that increased levels of stress significantly associate with the increasing typing pressure and amount of mouse contact. MoodScope is presented in paper [LLLZ13], which can infer the daily mood of a user by analysing communication history and application usage patterns of a mobile phone. Stress can also be measured based on the body movement and mobile phone usage. In paper [SP13], body movement can be quantified using a wrist sensor (accelerometer, skin conductance), while mobile phone usage includes: call, short message service, screen on/off, etc.

2.5.2 Affection Detection based on Posture

A fair amount of research works have studied the association between body language and affection. One of the underlying theories is called embodied theory of cognition [Cla97], which indicates that the affective states of people are manifested in their body movement and positions. Thus, the posture dynamics can provide the behavioural information about the affective state of the user.

Inspired by this theory, Sidney et al. [DCG07] developed a system that monitored the postures of 28 undergraduate students using the pressure mat mounted on the back and seat of the chair, and were able to differentiate between the boredom and flow states with accuracy of 47%. The main finding of their work is that the affective state of flow associates with heightened pressure posed on the seat of chair, while the boredom is manifested through a rapid change in pressure on the seat. In another study, Selene et al. [MP03] found evidence to support a relationship between the postural behaviour patterns and interest level of children who are working a computer-based learning task. In particular, through analysing the temporal posture sequence patterns of 10 children, their work can recognise three states related to a child's level of interest, with the accuracy of 82.3%.

To further identify more affective states, posture patterns combined with other modalities have been explored. For instance, Ashish et al. [KBP07] presented a system to detect the frustration state of people who are involved in the problem-solving activities in a computer, using multiple modalities including: face, postural movement, skin conductance and pressure on the mouse. Specifically, their work can recognise the frustration state of 24 participants with 79% accuracy. Specifically, our work focuses on analysing seating pressure of people in the context of stress detection.

Most similar work is done by Arnrich et al [ASLM$^+$10], which exploits seating pressure to detect stress. However, our work is different from theirs in terms of the analytics framework, seating pressure features and results. Our analytics framework is more

open than Arnrich et al's framework in terms of the candidate features and classifiers. In Arnrich et al's work, the features are limited to the spectra of the norm of the center of pressure, and only a single classifier called self-organizing map is evaluated. In our framework, a larger set of features in both time-domain and spatial-domain are extracted, and different types of classifiers are evaluated. Moreover, Arnrich et al's work does not conduct feature selection. We conduct correlation analysis and select the features with high correlation with stress. Feature selection is important, since it is able to filter out the random factors and thus improving the system generality. Our work can attain 86% accuracy, while Arnrich et al's work achieves 74% accuracy. Compared with Arnich et al's work, our method is more accurate and provides new understandings on how stress can be revealed by seating pressure.

Chapter 3

Accurate Sitting Posture Recognition with Low-cost Pressure Sensors

We study the problem of sitting posture recognition based on pressure sensors. Section 3.1 presents the overview of this work. We describe the design considerations in Section 3.2. In Section 3.3 and 3.4, we present the system overview and framework respectively. Evaluation is presented in Section 3.5. To evaluate the system utility, we develop three applications in Section 3.6. We discuss some research issues in Section 3.7 and conclude this work in Section 3.8.

3.1 Overview

In modern lifestyle, many people spend prolonged periods of time sitting. Research shows that long periods of physical inactivity raise the risk of heart disease, diabetes, cancer, and obesity. Moreover, poor sitting postures such as leaning forward cause upper limb and neck pain [Wik13]. To reduce risky sitting behaviours, monitoring people's sitting posture is greatly needed, which can raise people's awareness of sitting behaviours. In addition, the rehabilitation prognosis of stroke patients can be conducted

by measuring the sitting imbalance [SS90].

In addition to healthcare, sitting posture recognition also plays a vital role in human computer interaction. Sitting posture, along with sitting posture recognition techniques, can be extended as a human-computer interface. For example, people with disabilities can play games or control a wheelchair via sitting postures. Affective states can be further recognized based on the pattern of sitting posture [SK12]. Driver's drowsiness level and attention, can be inferred based on sitting posture information. Student's interest level towards a subject can be identified based on sitting posture along with other sensory inputs, which can facilitate learning [KMP01].

Current solutions for siting posture recognition, however, are impractical due to intrusiveness, high cost or low accuracy. A wearable sensor-based method attaches an inertial sensors on a user's back to collect motion data and identify the sitting posture [SP13]. This approach is intrusive and brings discomfort to users, since users are required to wear or attach those sensors. Vision-based methods use cameras to capture user sitting postures [Cam11]. Such method requires line-of-sight condition and may raise privacy concerns. To achieve a non-intrusive and privacy-preserving solution, researchers have deployed pressure sensors on the chair to infer sitting posture [XLH+11][TSP01]. But the main limitation of these solutions are the high cost (around 3000 USD), since they rely on high-fidelity pressure sensor array with more than two hundred sensors [Tek]. Note that paper [MKF+07] presents an idea to use 19 pressure sensors for sitting posture recognition. However, it only achieves 78% accuracy.

The objective of this work is to enable a practical sitting posture recognition technique that is accurate, non-intrusive, low-cost and real-time. We introduce Postureware, a sitting posture recognition system that can identify ten common sitting postures accurately in real-time using ten pressure sensors that are deployed inside a seat cushion.

During the design of the system, we address two main challenges: (i) How to place the minimum number of sensors to achieve the required recognition accuracy and cost. To

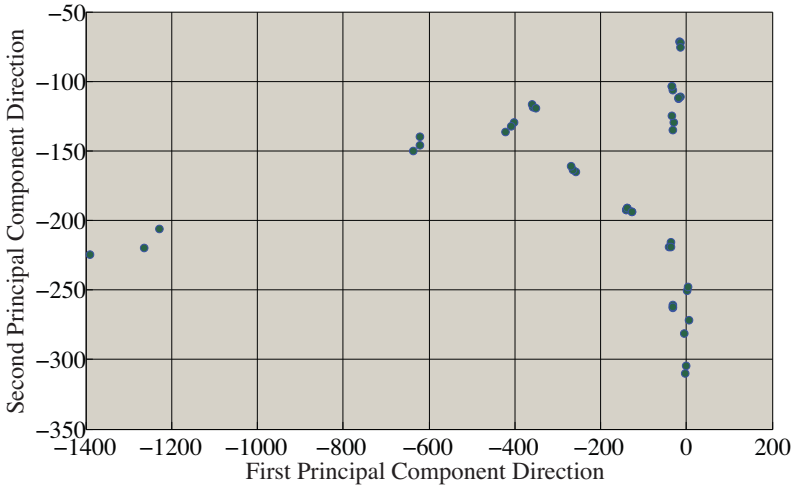


Figure 3.1: Sitting posture data from different users

address this challenge, we propose an information-theoretic metric to measure the sensor placement performance and then carry on a greedy algorithm to output the placement solution. This sensor placement solution is able to extract informative sensors yet with low computation cost. (ii) How to design an accurate sitting posture recognition model for different users. User diversity widens the intra-class differences and makes classification more challenging. Figure 1 visualizes the difference of sitting upright data samples from 15 people. We plot the first two PCA components on each axis of the figure based on the original data. It is surprising to see those sitting upright data samples are scattered distinctively. To this end, we propose a user-invariant sitting posture recognition framework, aiming to narrow the intra-class differences and widen the inter-class differences. Specifically, we conduct preprocessing to standardize the data; then use a set of user-invariant and distinctive features to model sitting postures; and finally use AdaBoost, an ensemble learning model to classify sitting posture.

The main contributions of this work are as follows:

- We design and implement a non-intrusive system for sitting posture recognition, which can accurately recognize sitting posture in real-time (8Hz) with low cost (150USD).

- We propose an information-theoretic sensor placement solution, enabling the system to achieve the same recognition accuracy using much fewer number of sensors. Specifically, the system can recognize 10 categories of sitting postures with 99.6% ten-fold cross validation accuracy with 10 sensors.

- We design an accurate sitting posture recognition model, which incorporates preprocessing, user-invariant feature extraction and AdaBoost classification, being able to resolve the challenges posed by user diversity and thus improving generalization accuracy. The system can achieve 84.7% generalization accuracy with 10 sensors.

- Finally, we implement a prototype system and develop two applications to evaluate the system utility. Unhealthy sitting posture recognition application can achieve 99.9% precision accuracy from the offline dataset. In second application, more than 80% of the users can finish the racing game smoothly using sitting posture.

3.2 Design Considerations

This section discusses the technical considerations that underpins the design of Postureware. Specially, the considerations include sitting posture set and system requirement.

3.2.1 Sitting Posture Set

The key function of Postureware is to identify a set of predefined sitting postures performed by people. In particular, Postureware focuses on ten of the most typical sitting postures found in office environment [Lue94], including: sitting upright; slouching; leaning back; leaning forward; leaning left; leaning right; left leg crossed; right leg crossed;

left leg crossed, leaning right; right leg crossed, leaning left (as shown in Figure 2). Another main reason to include those postures is because not just are they related to our demonstrative applications presented in the later section, but also chosen in previous works [MKF+07][TSP01], providing a benchmark for performance comparison of our system and previous works.

Due to diverse sitting habits, however, people sit differently even for the same posture. Take right leg crossed as an example, one may crosse right knee over left knee or cross their right ankle over the left knee. In order to make the selected sitting posture set more general, we consider several variant postures performed by people for certain posture. In our case, right leg crossed includes one crosses right knee over left knee; one crosses right ankle over the left knee.

Specifically, we target at fifteen sitting postures performed by the subjects, including: (1) sitting upright; (2) slouching; (3) leaning back; (4) leaning forward (angle<30 degrees); (5) leaning forward (angle>45 degrees); (6) leaning left (angle<10 degrees); (7) leaning left (angle>20 degrees); (8) leaning right (angle<10 degrees); (9) leaning right (angle>20 degrees); (10) left leg crossed in ankle; (11) left leg crossed in knee; (12) right leg crossed in ankle; (13) right leg crossed in knee; (14) left leg crossed, leaning right; (15) right leg crossed, leaning left.

Compared with the posture set in previous works [MKF+07][TSP01], the siting posture set in this work is more general and representative. As a result, it is more challenging to conduct the sitting posture classification.

3.2.2 System Requirement

To make it practical, we think the system should be able to provide effective functionalities and good user experience at a reasonable cost. Particularly, we highlight the following stringent requirements:

Figure 3.2: Sitting posture set: (a) sitting upright; (b) slouching; (c) leaning back; (d) leaning forward; (e) leaning left; (f) leaning right; (g) left leg crossed; (h) right leg crossed; (i) left leg crossed, leaning right; (j) right leg crossed, leaning left

- *Accurate*: the system should be able to recognize sitting postures performed by diverse users accurately. Accuracy should be the top priority among all the system performance metrics.

- *Real-time*: the system should be able to recognize people sitting posture in real-time, so that a variety of applications such as human computer interaction can be supported.

- *Non-intrusive*: the system should provide comfortable experience for users. Since a normal user usually spends a lot of time seating every day, it is of great importance to make it non-intrusive.

- *Low-cost*: the cost of the system should be low and afforded by most users for widespread adoption.

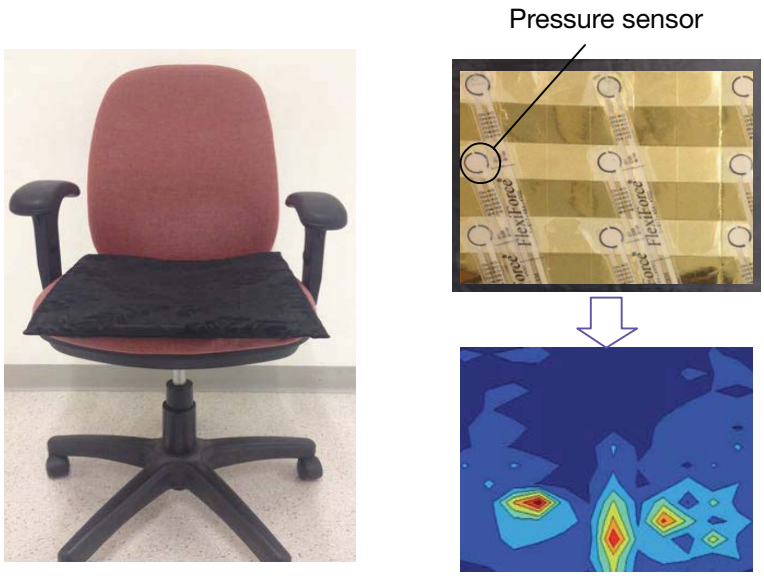


Figure 3.3: Postureware overview

3.3 System Overview

3.3.1 System Architecture

The delivery of Postureware system is a sensing mat that can recognize user's sitting posture. The sensing mat leverages the pressure sensors, as shown in Figure 3, to obtain the pressure distribution on top of the chair. Then the sitting posture can be inferred based on the distinctive pressure distribution pattern. Postureware contains four components: pressure sensors, sensor placement, sitting posture recognition model and application, which is shown in Figure 4. On top of sitting posture recognition, we develop three applications: first application is to monitor unhealthy sitting posture, second one is to use sitting posture to play a car racing game and the last application is to control wheelchair by changing sitting posture. In particular, Postureware contains two key components: (1) sensor placement; (2) sitting posture recognition model.

Sensor placement. To reduce system cost, we need to place as fewer number of sen-

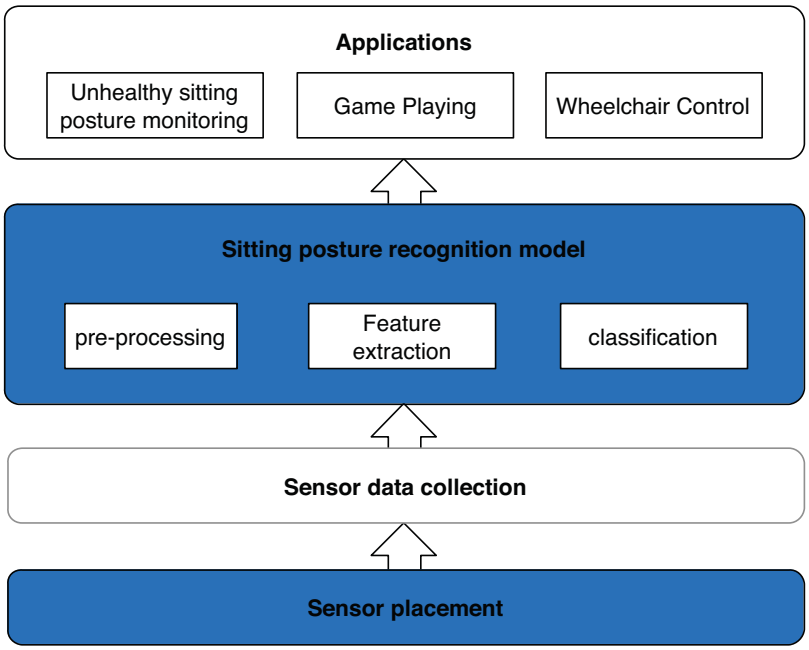


Figure 3.4: Postureware architecture

sors as possible, as each pressure sensor costs around 15 USD [Tek]. This component is designed to output a number of positions on the chair top to place sensors, aiming to minimize the number of deployed sensors while achieving the required recognition accuracy.

Sitting posture recognition model. Given the original pressure sensor data, a sitting posture recognition model is designed to infer the sitting posture accurately. Specifically, sitting posture recognition model consists of three components: pre-processing, feature extraction and classification. The first two components aim to represent sitting posture using distinctive and user-invariant patterns, whereas classification component is used to train the classifier to identify the posture based on features.

3.3.2 Problem Formulation

In this work, the research problem can be stated as follow: given a set of users, a set of predefined sitting postures, a number of pressure sensors to be deployed on top of

a chair, a required recognition accuracy; the objective is how to deploy the sensors and analyze the sensory data such that the sitting posture of the users can be recognized with the required recognition accuracy. More specifically, we formulate the problem from the supervised learning perspective.

Given:

- A set of users U, varying in weight, age, sitting habit (center of contact area).
- A set of pre-defined sitting postures P.
- N number of pressure sensors array S placed on top of chair.
- Dataset D collected from sensor array S, when all the users $\forall u_i \in U$ sits in all the postures $\forall p_j \in P$.
- Data sample from D is in the format of (x, y), where x is a N-dimension vector, denoting the sensor data from S; $y \in P$ is the corresponding sitting posture label.

Objective:

How to deploy given number N pressure sensors and construct a recognition function $f(x) : x \rightarrow y$ from D such that while user sitting on the chair, the sitting posture of user from U can be accurately recognized based on $f(x)$.

3.4 Postureware Framework

3.4.1 Sensor Placement

Sensor placement focuses on deploying as fewer number of sensors as possible to obtain the required accuracy of recognition. More specifically, the sensor placement problem is how to select a few number of sensors from the densely deployed sensor

grid. This problem has been studied in literature [MKF+07]. The evaluation metric for placement solution is how fit the placed sensors can reconstruct the features that are computed from high-fidelity sensor array. Generally, sensors on the boundary of contact area are preferred to select under this evaluation metric, since most of the selected features are related to the shape of contact area (e.g., size, position, center of contact area, distance of contact area to the edges of the chair).

Algorithm 1: Information-theoretic sensor placement algorithm

input:
dataset $D = \langle (s_1, y_1), ..., (s_m, y_m) \rangle$ with sensor data vector s_i and corresponding posture labels $y_i \in Y = \{1, ..., n\}$.
total sensor set S.
deployed sensor number N.
output:
selected sensor set t S_S.
Initialization:
selected sensor set $S_S = \emptyset$;
unselected sensor set $S_U = S$;
for $t = 1, ..., N$ **do**
    for $s_i \in S_U$ **do**
        $H(D) = -\sum_i P(D_i) \log P(D_i)$;
        compute $H(D, s_i) = H(D) - H(D|s_i)$;
    end
    $s_m = \arg\max_{s_i} H(D, s_i)$;
    $S_S = S_S \cup \{s_m\}$;
    $S_U = S_U \backslash \{s_m\}$;
end

However, we argue that this sensor placement solution can be further improved due to

the following reasons. First, due to diversity of people, features such as position and size of contact area between people and the chair varies, making the system difficult to generalize. Second, we found that discriminative sensors are normally located inside rather than on the boundary of contact area through observing the pressure distribution of sitting postures (see Figure 5). This observation implies that an effective sensor placement solution does not necessarily reconstruct the features well.

In order to place sensors in the positions that capture the distinctive data, we design an information-theoretic sensor placement algorithm. At a high level, this algorithm works like a greedy algorithm. First, it evaluates the performance of all the unselected sensors based on the information gain [Wik14]. Information gain of a sensor is measured by the mutual information between its sensory data and the data from all the unselected sensors. Then the algorithm will include the sensor with the highest information gain into the final sensor placement solution. The details of the algorithm is presented in algorithm 1. The rationale of this method is that larger information gain of a sensor indicates it contains more distinctive information, and thus improving the classification accuracy. This placement solution has two advantages: first, it is fast to compute; second, it is able to extract informative sensors, which is independent of the selected features and classification algorithms.

3.4.2 Pre-processing

Pre-processing is the first module of sitting posture recognition model. Once the pressure sensors are deployed, the pressure sensor data will be collected. However, the original data cannot be directly used for learning due to three main reasons. Firstly, the data is not calibrated, since the initial value of pressure sensor changes when no force is applied. Secondly, the data may not reflect user's sitting posture because user might not sit on top of all the deployed sensors. Thirdly, the data value varies due to user's diverse weight. In order to address the above issues, we design a pre-processing

module, aiming to calibrate, extract and standardize the original data.

Data calibration. To calibrate the sensor data, we first record the initial value of sensors when no force is applied. Then data can be calibrated simply by using the sensor data to subtract the initial value of corresponding sensor.

Contact area data extraction. Contact area data refers to the data of sensors that are activated by users while sitting on top of the chair. Contact area data extraction is to crop the sensors where covers all the data from the contact area between the subject and the chair. To extract contact area data, we adopt a threshold-based method that determines the sensors contacted by users.

Data normalization. Data needs to be further nomalized to mitigate the effect caused by various weights of users. In this work, we transform the data into standard normal distribution.

3.4.3 Feature Extraction

Feature extraction is to extract a set of features from the sensor data to represent sitting posture, which is critical to the recognition accuracy. Most of the features proposed in previous works [TSP01][MKF+07][LA06] are user-dependent, as they are measured in an absolute manner. For example, features such as the position and size of the bounding box of the pressure area, differers by people's sitting habit and weight.

The novelty of feature extraction in this work is that a set of both user-independent and distinctive features are proposed. Our philosophy of feature extraction is based on two principles. First principle is that feature should be computed in a relative manner. The insight of this principle is to enable the features to be user-invariant. For instance, the positions of top-k sensors will be considered as the relative positions of top-k sensors with respect to a reference point. Second principle requires the feature to indicate the pressure redistribution triggered by the sitting posture. This principle is inspired by

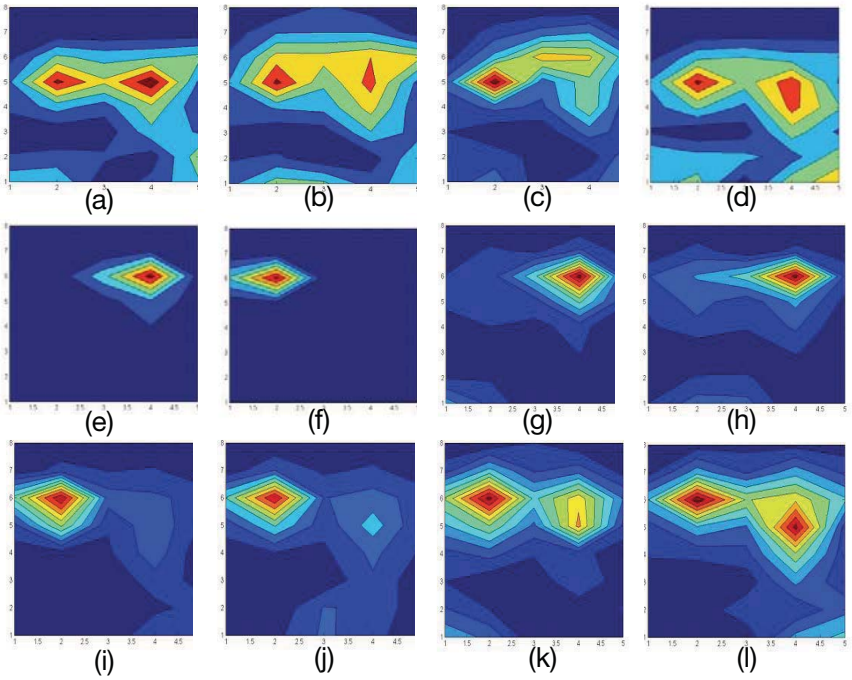


Figure 3.5: Pressure distribution of different sitting postures: (a) sitting upright; (b) slouching; (c) leaning back; (d) leaning forward; (e) leaning left; (f) leaning right; (g) left leg crossed in ankle; (h) left leg crossed in knee; (i) right leg crossed in ankle; (j) right leg crossed in knee; (k) left leg crossed, leaning right; (l) right leg crossed, leaning left

the observation from Figure 5: different postures in essence results in the pressure redistribution of sensors covered by human body.

Our features mainly represent two kinds of information(see Figure 6): (1) position-based feature, which represents the relative position of certain points. (2) ratio-based feature, which measures the pressure ratio between regions of points. To make the analysis more convenient, the seat cushion is considered as a two-dimension Cartesian coordinate system with x, y axes. The center of seat cushion is considered as origin, while the direction of y axis is set to the front direction of chair. If the location (x, y) has been deployed with a sensor, then it is denoted as $s(x, y) = 1$, and the corresponding sensor value is represented as $d(x, y)$. In particular, position-based feature include the following features.

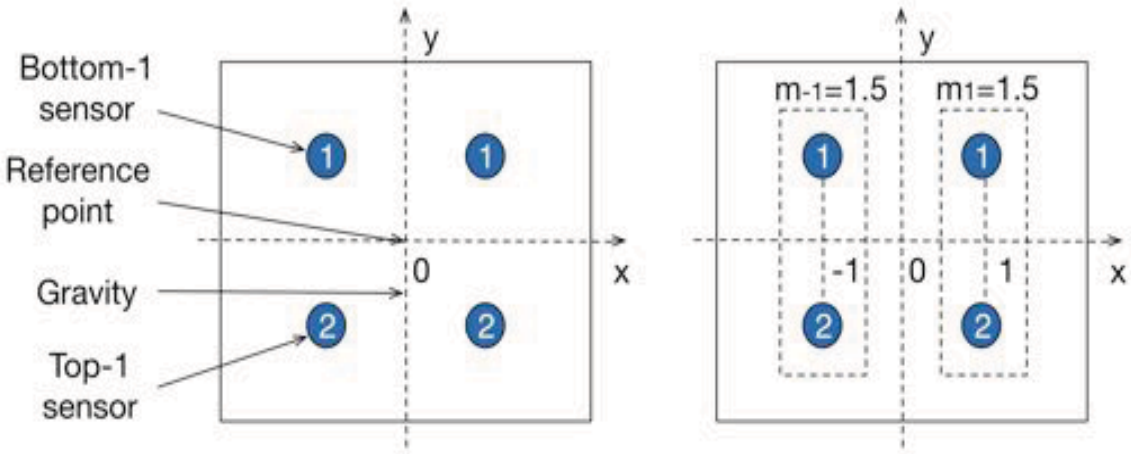


Figure 3.6: Feature extraction: each circle denotes a sensor and the number denotes the pressure value

Reference point. To make features user-independent, we need to set up a reference point and thus enabling the relative measurement with respect to the reference point. In our case, the center of sensor positions (o_x, o_y) is selected as the reference point. It can be calculated as follows:

$$o_x = (\sum_{\forall s(x,y)=1} x)/(\sum_{\forall s(x,y)=1} 1) \tag{3.1}$$

$$o_y = (\sum_{\forall s(x,y)=1} y)/(\sum_{\forall s(x,y)=1} 1) \tag{3.2}$$

Gravity. It refers to the center of pressure distribution on sensors. We denote the pressure data in position (x, y) as $d(x, y)$. Then the position of gravity (g_x, g_y) can be calculated based on the following equations:

$$g_x = (\sum_{\forall s(x,y)=1} x \cdot d(x,y))/(\sum_{\forall s(x,y)=1} d(x,y)) - o_x \tag{3.3}$$

$$g_y = (\sum_{\forall s(x,y)=1} y \cdot d(x,y))/(\sum_{\forall s(x,y)=1} d(x,y)) - o_y \tag{3.4}$$

Top-k sensors. Top-k sensors refer to k sensors with the highest pressure value. In particular, It extract three kinds of features: the center, the positions and variance of

positions of top-k sensors. The center of top-k sensors can be computed using the same equations of gravity. The variance of positions can be measured by summing up the variance of top-k sensor positions along x-axis and y-axis.

Bottom-k sensors. Bottom-k sensors means k sensors with the lowest pressure value. Similar to top-k sensors, three kinds of features are extracted from bottom-k sensors: the center, the positions and variance of positions.

Another category of features are ratio-based feature, which includes features as below:

X-axis ratio. This set of features R_x mainly measures the pressure difference along x-axis direction. Specifically, we measure the average value ratio among sensor blocks along with x-axis (vertical to chair front direction).

$$R_x = \{r_x | m_{x+n}/m_x, \forall_x m_x \neq 0\} n = 1, 2, ... \tag{3.5}$$

$$m_x = (\sum_{\forall y, s(x,y)=1} d(x,y))/(\sum_{\forall y, s(x,y)=1} 1) \tag{3.6}$$

where m_x represents the average pressure in the xth sensor row along x-axis direction.

Y-axis ratio. Similar to X-axis ratio, this set of features measure the pressure difference along y-axis direction.

After feature extraction, however, some features may be redundant or even ineffective. Therefore, we need to select a set of distinctive features to best represent the posture. To perform feature selection, we adopt one commonly used method called information gain [MBN02], which is independent of classification algorithm. The information gain of each feature with respect to the sitting posture will be calculated and feature with the highest value will be selected. Then keep repeatedly conduct previous step for the unselected features until classification accuracy has not increased.

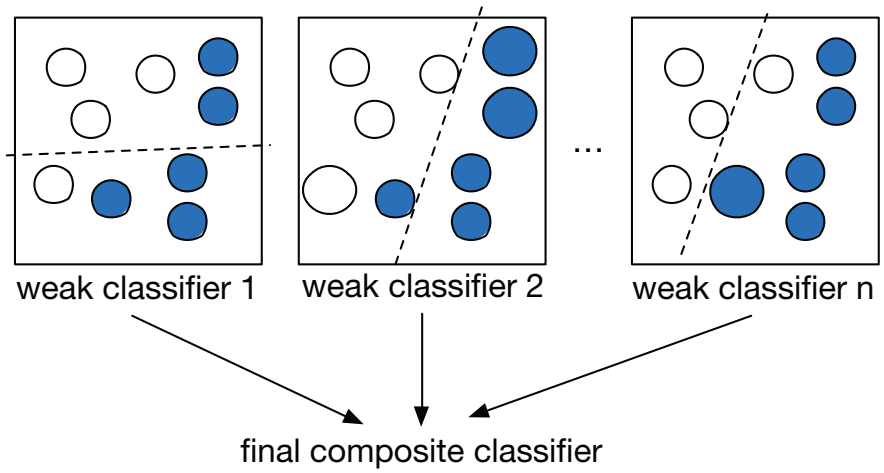


Figure 3.7: AdaBoost classification algorithm

3.4.4 Classification

The first two modules of sitting posture recognition model aim to construct representative features, while classification module is designed to train a classifier to identify the sitting posture based on the features. To further improve the generalization ability of the classifier, we adopt AdaBoost [FS96] as our classification algorithm. The main advantage of AdaBoost is that the error on the training dataset can be arbitrarily small and yet the generalization error is low.

Primarily, AdaBoost is an ensemble method, as shown in Figure 7, which combines a number of different classifiers. The main insight of AdaBoost is repeatedly running a given weak learning algorithm on various distributions over the training data, and then combine the classifiers trained in each turn into a single classifier. In this work, the input of AdaBoost algorithm is a training dataset $\langle (x_1, y_1), ..., (x_m, y_m) \rangle$ with feature vector $x_i \in X$ and corresponding posture labels $y_i \in Y = \{1, ..., n\}$; the output of the AdaBoost algorithm is a composite classifier $c_f(x)$. The details of AdaBoost algorithm is presented in Algorithm 2.

The selection of weak classifier algorithm is very critical to the performance of AdaBoost. We will evaluate the performances of several commonly used weak classifier

Algorithm 2: AdaBoost-based sitting posture classification algorithm

input:

Training dataset $\langle (x_1, y_1), ..., (x_m, y_m) \rangle$ with feature vector $x_i \in X$ and corresponding posture labels $y_i \in Y = \{1, ..., n\}$;

Weak learning algorithm *WL*;

Iteration number T.

output:

Final classifier: $c_f(x) = \arg \max_{y \in Y} \sum_{t:c_t(x)=y} \log \frac{1}{\varepsilon_t(1-\varepsilon_t)}$.

Initialization:

$D_1(i) = 1/m$ for all i, meaning all the data will be treated equally.

for $t = 1, ..., T$ **do**

1. Call *WL* and train the classifier $c_t : X \to Y$, given the data distribution D_t.

2. Calculate classification error ε_t: $\varepsilon_t = \sum_{i:c_t(x_i) \neq y_i} D_t(i)$;

If $\varepsilon_t > 1/2$, then set $T = t - 1$ and abort the loop.

3. Calculate the distribution at round $t + 1$:

$D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} \varepsilon_t/(1-\varepsilon_t) & \text{if } c_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$, where Z_t is the normalization factor.

end

algorithms including decision stump, naive Bayes and C4.5. The performance of different weak learning algorithms will be evaluated in the evaluation section.

- Decision stump. Decision stump is a learning model that is a one-level decision tree.
- Naive Bayes. Naive Bayes is a parametric classifier based on Bayesian Theorem with assumption that all features are independent.
- C4.5. C4.5 is a decision tree algorithm which is commonly used as a weak clas-

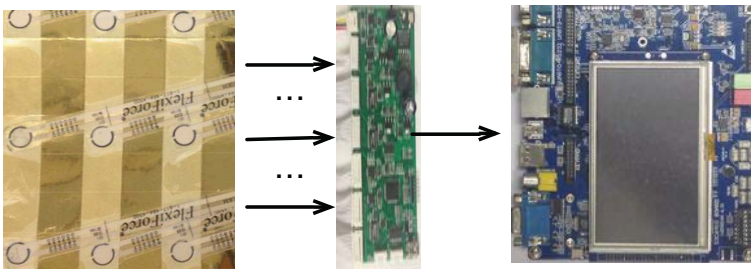


Figure 3.8: Sitting posture recognition hardware and dataflow

sifier in AdaBoost.

3.5 Evaluation

3.5.1 System Implementation

Postureware incorporates three main hardware components as illustrated in Figure 8: a pressure sensor array, a data sampling module and a data processing board. As discussed earlier, the pressure sensor array is used to obtain sensor value, whereas a sampling module is used to collect data with certain frequency. After receiving data from the data sampling module, the data processing board executes the algorithms of sitting posture recognition and delivers the identified sitting posture to applications.

Initially, pressure sensors is deployed in 5*8 grid and the distance between neighbouring sensor is 4cm vertical to the chair front direction and 6cm along the chair front direction. FlexiForce A201 [Tek] is chosen as our pressure sensor, because it's very thin (0.208 mm thick) and very stable (less than 3% linearity error), which can provide non-intrusive user experience. In data sampling unit, the pressure data is encoded as a 16-bit digital value and the data sampling rate is set as 15Hz. The sensors are equally divided into four groups. At each sampling period, we will sample one sensor in each group simultaneously. Data processing board is essentially an arm development board

(EBD2410), where runs the sitting posture recognition algorithms and then output the recognized sitting posture. We implement the algorithms using c++ language. The communication between the development board and other devices is done with socket programming.

3.5.2 Experiment Setup

We run a pilot study to evaluate the performance of Postureware. We have 15 people to be our experiment subjects, including 13 males and 2 females. The weight of those subjects ranges from 45kg to 85kg, while the height is between 150cm to 180cm. The diversity of subjects in terms of sex, weight and height makes the dataset representative and enables the system to have good generalizability.

All subjects are instructed to perform ten categories of sitting postures as shown in Figure 2. As discussed previously, people sit differently even for the same posture. Then subjects will perform the predefined fifteen sitting postures discussed in section II. Note that the subjects are intervened to conduct predefined set of postures, because we find that some of users can not perform some postures accurately such as sitting upright. We argue that intervention is necessary when instructing user to hold some postures, mainly because subjects may not know what a standard healthy posture looks like.

In order to measure the performance, we use three widely used evaluation metrics: precision, recall and F-measure. Let us consider a sample of activity A_1 in the test dataset. If the predicted activity is A_1, it will be counted as a true positive (TP). Otherwise, assume the predicted activity is A_2, then it would be counted as a false positive (FP). In addition, activity A_1 will be counted as a false negative (FN) since it is missing in the prediction. F-measure reflects the overall effect of both precision and recall. The metrics can be computed based on the following equations.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.7}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.8}$$

$$F = 2\frac{precision \times recall}{precision + recall} \tag{3.9}$$

To evaluate the system performance, we first divide the dataset into two parts: 10 percent of data will be used for training, and the rest 90 percent will be used for testing. We use 10-fold cross validation to measure the performances of classification models. In order to demonstrate the effectiveness of the proposed methods, we compare the performance of proposed methods against a set of benchmarks. The benchmarks of sitting posture recognition includes the widely adopted classifier approaches such as Naive Bayes, Support Vector Machine [B+06]. Those classifiers are implemented in open source machine learning software Weka 3.6 [HFH+09].

3.5.3 System Performance

This section presents the evaluation results of sensor placement solution, feature selection, sitting posture recognition and the system real-timeness.

Sensor Placement We are interested in figuring out the effective solutions for sensor placement. We first deploy 40 pressures sensors in 5*8 gird and then run the proposed sensor placement algorithm to select a number of sensors. Figure 9 shows the sensor placement solutions given different number of sensors. Specifically, we plot the sensor placement results when number of deployed sensors equals 5, 10, 15, 20. Interestingly, with the more sensors deployed, two areas will become denser: one is close to the front boundary of a chair, while the other is on the region around the center of the chair.

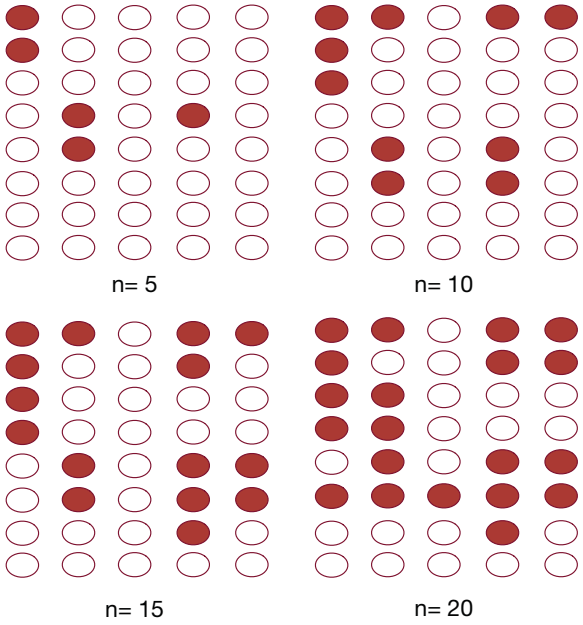


Figure 3.9: Sensor placement solution: all users

The results are reasonable because the pressure distribution in front boundary of chair provide more distinctive information to differentiate postures like left legs crossed in knees and left legs crossed in ankle. On the other hand, the region around the center of chair is mainly covered by the hip, offering distinctive information for different sitting postures.

Note that the sensor solution is unsymmetrical. We think the asymmetricity of placement could be attributed to three factors: posture set asymmetricity, subject diversity and sensor reading noise. First, those symmetrical sitting posture pairs are not performed in a symmetrical manner. For instance, a symmetrical posture pairs leaning left and leaning right, would exhibit asymmetrical sensor readings because it is very difficult for subjects to lean at a given angle. Second, the experiment subjects varies in weight and sitting habit, so that the sensor readings are asymmetrical. As a result, the sensor solution is unsymmetrical. Third, sensory data is noisy, which will impact the sensor placement solution.

To gain further insight, we also evaluate the impact of user weight towards sitting sensor placement. As shown in Figure 10 and Figure 11, it is interesting to find that the

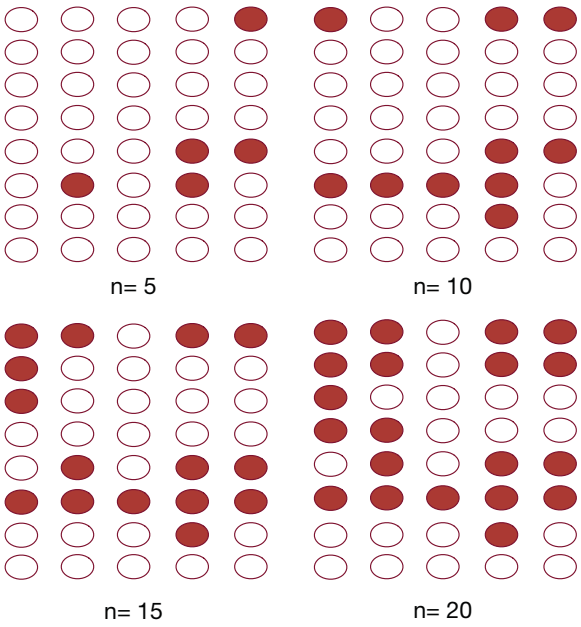


Figure 3.10: Sensor placement solution: users within (75kg, 85kg)

placement solution for user with weight (75kg, 85kg) prefers the region in the entire 6th row (close to chair back); while the solution for user with weight (45kg, 55kg) tends to deploy sensors on the region around the center of chair. According to the above observation, we find that the sensor placement depends on the weight of user. The rationale is that the pressure distribution on a chair top exhibits different patterns when users with different weights sit. More specifically, for a user with weight (45kg, 55kg), the pressure distribution on the hip area is more informative to differentiate different postures, whereas for a user with weight (75kg, 85kg), pressure distribution on the thigh area is more discriminative.

We also want to evaluate the impact of sensor number towards the recognition accuracy. Figure 10 plots the recognition accuracy of sitting postures as a function of the sensor number. As shown in the figure, with 10 sensors placed using the proposed method, it can achieve 99.6% ten-fold cross validation accuracy for 15 sitting postures. Furthermore, increasing sensor number significantly improve the recognition accuracy when the sensor number is less than 5.

In addition, Figure 12 also shows the comparison between the proposed method and

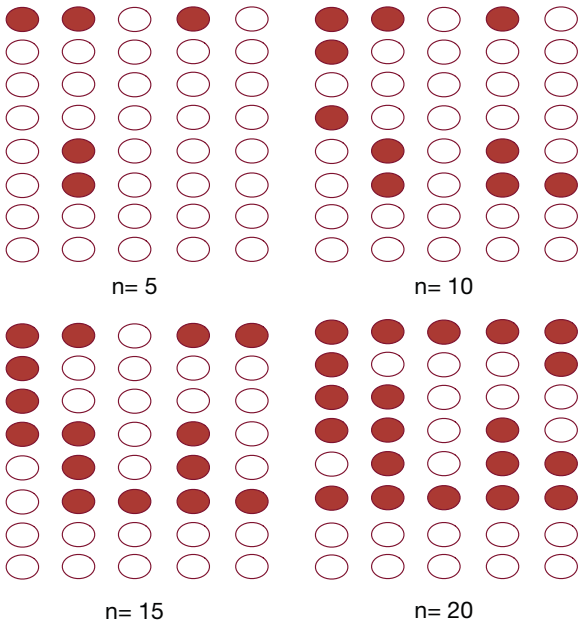


Figure 3.11: Sensor placement solution: users within (45kg, 55kg)

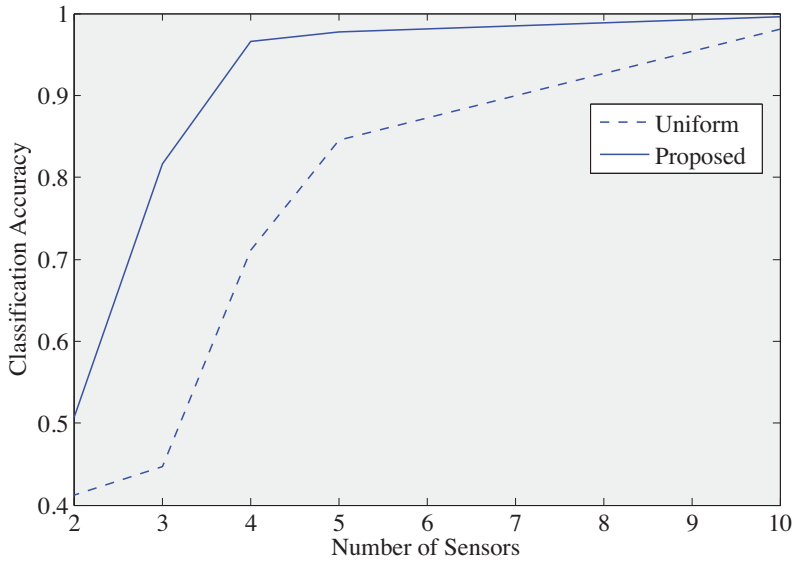


Figure 3.12: Sensor placement solution with respect to sensor number

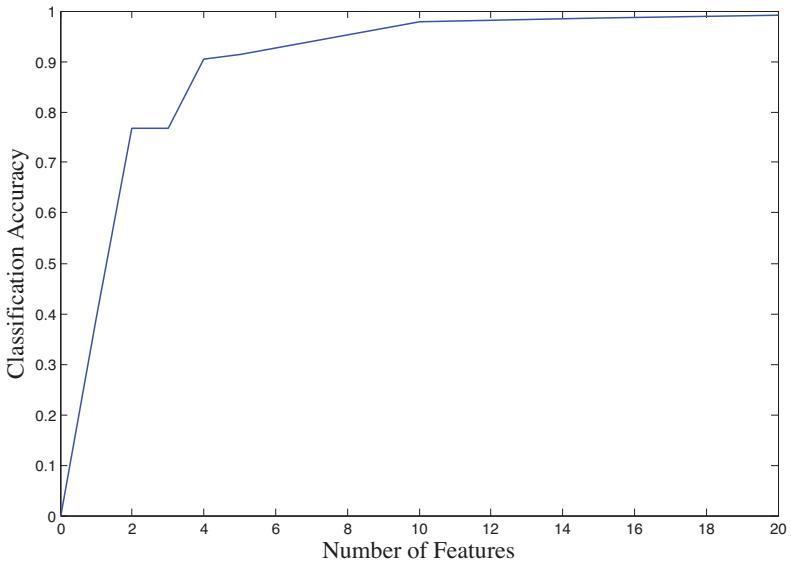


Figure 3.13: The impact of feature dimension on classification accuracy

the uniform method. The uniform sensor placement solution is obtained based on the uniformly generated random value ranging from 1 to 40, which indicates the deployed location. The key takeaway from the results is as follows: when the number of deployed sensors is small, the proposed method achieve much higher accuracy than the uniform method. However, the gap between the proposed method and the uniform method narrows as the number of sensors increase. The intuition behind this observation is that the more sensors are deployed, the more both solutions overlap in terms of sensors.

Feature Selection The objectives of feature selection experiment are two-folds: first objective is to rank the features in terms of the information gain metric; second objective is to evaluate the impact of feature number towards recognition accuracy. Based on information gain method, the gravity feature is considered as the most important feature out of 65 features, whereas the following four top-5 features all belong to ratio-based features, indicating the pressure ratio between the left side and right side of the chair. It is because that different sitting postures essentially results in the pressure redistribution on the chair top, which can be reflected by gravity feature and ratio-based features.

We will further evaluate the impact of feature number to the recognition accuracy. Figure 13 plots the recognition accuracy with respect to number of features when sensor

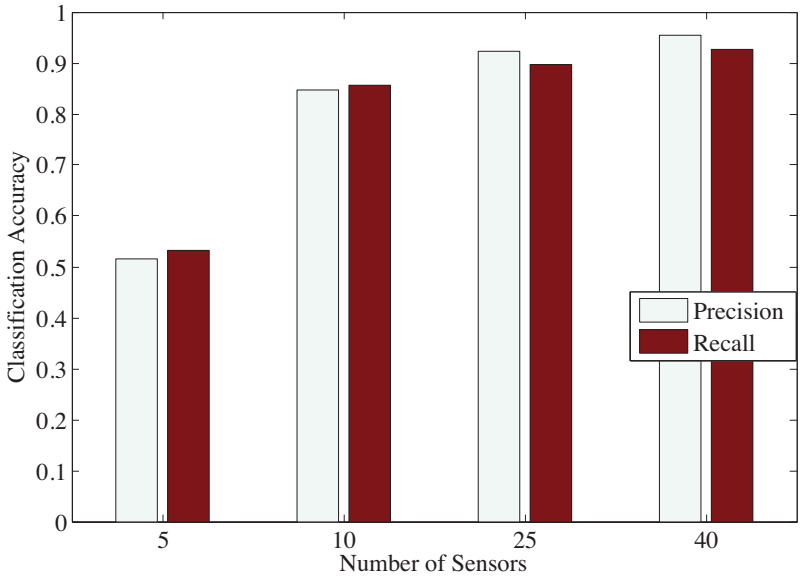


Figure 3.14: Generalization accuracy

number is 15. Note that the recognition accuracy generally improves when the feature number increases. Specifically, when the number of features is 5, the proposed method can achieve 91.4% ten-fold cross validation accuracy. The high recognition accuracy with a few number of features demonstrates that the extracted features are distinctive and user invariant, which can be used to differentiate different sitting postures effectively.

Sitting posture recognition result The main objective of the this experiment is to evaluate the performance of the proposed sitting posture classification algorithm. Support vector machine and naive Bayes act as the baseline methods. Table I summarizes the results when the number of deployed sensors are 10. AdaBoost with C4.5 weak classifier achieves the best performance with 99.6% precision accuracy and 99.6% recall accuracy, while AdaBoost with decision stump weak classifier has a very inaccurate performance. It is worth emphasizing that extracted features in this work are in essence discriminative such as the relative position and the pressure ratio, which generally prefers discriminative models like support vector machine than generative model like naive Bayes.

Generalization performance is a very important metric on how well the classification

Table 3.1: Sitting posture recognition confusion matrix

Classification Algorithm	Precision	Recall	F-measure
Naive Bayes	0.59	0.544	0.566
Support Vector Machine	0.88	0.864	0.868
AdaBoost(Naive Bayes)	0.616	0.568	0.566
AdaBoost(Decision stump)	0.026	0.14	0.042
AdaBoost(C4.5)	**0.996**	**0.996**	**0.996**

model can be applied to different users. To measure the generalization performance, we first train a classification model based on data of 14 subjects, and use the data of a new subject to evaluate the recognition performance. Figure 14 plots the generalization accuracy given different sensor number. When the sensor number is larger than 10, the generalization accuracy is quite high with precision larger than 84.7% and recall larger than 85.7%.

Table II shows the confusion matrix of generalization test. Surprisingly, the bottleneck of the recognition model lies the classification of sitting upright and leaning back. Most of the sitting upright postures are misclassified as leaning back. The main reason is that sitting upright is quite similar to leaning back in term of pressure distribution on chair top, as both postures result in a balanced pressure distribution. The main difference between theses postures may be the gravity of pressure distribution. However, sometimes the difference may not be evident because some people sit very close to the chair back, making sitting upright and leaning back very similar. The possible remedy could be adding a pressure cushion on the backrest, which can help differentiate the sitting upright and leaning back.

Real-time performance This experiment is to evaluate the real-time performance of system. Real-time performance is of critical importance to applications such as using sitting posture to control game or wheelchair. Postureware can recognize sitting posture in 8 Hz, which is able to support sitting posture monitoring since user's posture switch

Table 3.2: Generalization test confusion matrix. (SU) sitting upright; (SL) slouching; (LB) leaning back; (LF) leaning forward; (LL) leaning left; (LR) leaning right; (LC) left leg crossed; (RC) right leg crossed; (LCR) left leg crossed, leaning right; (RCL) right leg crossed, leaning left

	SU	SL	LB	LF	LL	LR	LC	RC	LCR	RCL
SU	**0**	1	167	0	0	0	0	0	0	0
SL	0	**158**	0	7	0	0	2	0	0	0
LB	90	0	**59**	21	0	2	0	0	0	0
LF	0	0	0	**340**	0	0	0	0	0	0
LL	0	0	0	0	**316**	0	0	0	0	0
LR	0	0	0	0	0	**341**	0	0	0	0
LC	0	0	0	0	0	0	**257**	0	0	0
RC	0	0	0	0	0	0	0	**401**	0	0
LCR	0	0	0	0	0	0	58	0	**123**	0
RCL	0	0	0	0	0	0	0	15	0	**176**

frequency is much lower than 8 Hz. Besides, it can also enable good user experience for sitting posture-based interaction, because we found that it takes more than 0.2s for most subjects to switch to another posture. In Postureware, the delay of sitting posture recognition is subject to the sensor sampling rate, because the execution of recognition can be done within a sampling interval (125 ms). Therefore, the real-time performance can be further improved with increased data sampling rate.

3.6 Applications

To further evaluate the system utility, we build three prototype applications, consisting of unhealthy posture monitoring and posture-based interaction [**?**].

3.6.1 Unhealthy Sitting Posture Monitoring

According to a report from Mayfieldclinic [May09], only sitting upright is considered a healthy posture, while the rest of 9 postures may potentially harm our health. To help promote healthy sitting behaviours, we develop an application that can monitor the unhealthy postures. Technically, the task is reduced to a two class classification problem.

Table III shows the confusion matrix of unhealthy posture recognition based on the dataset. Specifically, unhealthy postures can be identified with 99.9% and 99.8%. The reason why such high accuracy could be achieved is that the user has been involved in the training phases. One of the limitations in this application is that only the static posture is identified. In fact, a temporal sequence of postures deserves more attention, as it can capture fine-grained sitting behaviour which associates more with the health.

Table 3.3: Unhealthy sitting posture classification confusion matrix

	Healthy Posture	Unhealthy Posture	Recall
Healthy Posture	199	4	0.98
Unhealthy Posture	5	2825	0.998
Precision	0.975	0.999	

3.6.2 Posture-based Game Interaction

This application aims to use user's sitting posture to play games (as shown in Figure 15). In particular, we select the racing game named Need for Speed 4. In order to provide good user experience, we have imposed two conditions to use sitting posture to represent commands: (1) the sitting postures must be intuitive and simple so that human can easily perform it; (2) the sitting postures must be distinctive to be classified with high accuracy. Therefore, we associate certain postures with different control commands as follows: leaning left or right will turn the car left and right, while leaning

forward accelerate the car and leaning back decelerate or reverse the car.

Figure 3.15: Using sitting posture to control the racing game

The application has been demonstrated on the information day for undergraduate admissions 2014. Over 40 visitors try the prototype system and more than 80% of them can finish the game smoothly. To further quantify the utility of sitting posture-based interaction, we evaluate the interaction cost of sitting posture. The interaction cost of a given posture is measured by the time it takes to switch from sitting upright to the given posture and then switch back. Table IV summarizes the interaction cost when the leaning angle of those postures are set to 10 degrees. Leg/right leg crossed has the lowest interaction cost (0.8s), while leaning back has the highest interaction cost (1.2s). Note that there is a tradeoff between interaction cost and recognition accuracy. Low interaction cost means that the movement for posture switch is subtle, making it difficult for recognition.

Table 3.4: Sitting posture-based interaction cost

	Leaning forward	Leaning back	Leaning left/right	Leg/right leg crossed
Time	1s	1.2s	1.1s	0.8s

3.6.3 Posture-based Wheelchair Control

Second application takes the scenario of wheelchair control. This application will demonstrate how user can control the wheelchair naturally using sitting posture. (See Fig.16). To make posture-based control practical, we think two stringent requirements should be met: (1) the chosen sitting postures should be classified with a high accuracy; (2) the sitting postures can be performed by the users easily for a long time. To align with the above-mentioned requirement, we associate certain postures with different control commands as follows: sitting upright will stop the wheelchair, while sitting forward will trigger the movement of the wheelchair. Leaning left and leaning right will control the left and right turn of wheelchair respectively. Particularly, since a wheelchair holds being motionless for most of the time, sitting upright which is the normal state of a wheelchair seater, is selected to stop the wheelchair.

Figure 3.16: Using sitting posture to control the wheelchair

Two lessons have been learned after the pivot study of using postures to control the wheelchair. First lesson is that we should minimise the time for switching between different postures, given a certain chair speed and a required classification accuracy. Wheelchair control application is safety-critical, the time for different command switching should be stringently constrained. Otherwise, the seater is in great danger, as the

Table 3.5: System requirements for different application areas

Application area	Accuracy	Realtimeness	Posture set
Health care	Median	Median	Small
Safety-critical	High	High	Median
Game	High	High	Large

wheelchair may hit the obstacles if it takes a long time to switch from walking to stopping. Another lesson we have learned is the recognition system should be robust to handle unexpected postures. Unexpected postures usually occur when a user switch from a predefined posture to another one. Without a robust mechanism to deal with those unexpected postures, the system will become very unstable.

3.7 Discussion

Postureware shows that user's static sitting posture can be accurately recognized based on pressure sensors deployed on the chair top. However, some open problems still remain to be resolved.

System requirements vs Application areas. Application area actually determines the posture recognition system requirements including: accuracy, realtimeness and posture set. According to the Table, game applications demands the highest system requirements. Usually a computer game requires a user to input a sequence of various control commands accurately and timely. Therefore, to enable sitting posture as an input for game application, the system should be able to identify a large set of postures with high accuracy and high realtimeness. Healthcare application focuses on monitoring the unhealthy postures and sedentary behaviours. In practise, only a limited number of postures are considered to be appropriate such as sitting upright, so that a small set of postures need to be differentiated. Note that what harms the seater the most is holding inappropriate sitting postures for a prolonged time. Therefore, long-

term monitoring is expected in the system, rather than real-time monitoring. For some safety-critical applications like using sitting posture control wheelchair, high accuracy and realtimeness should be guaranteed in the system.

Robustness. In a real setting, people may hold a posture that does not belong to any of the defined postures, particularly while switching postures. For example, when people change from leaning forward to crossed legs, the posture they hold before crossing legs is not part of the posture set. The question is how to handle those postures that does not belong to the defined posture set. One straightforward approach is to smooth the recognition result and thus eliminating the unexpected results. The smoothing can be done using majority vote method to make final decision from classifier result during a fixed time window. However, there is a tradeoff between fault-tolerance and real-time performance. When the time window is set to a large value, the unexpected results can be more tolerated, but delay of the recognition increases. How to achieve the best balance between robustness and real-time performance requires further investigations.

Training Dataset. Training dataset is needed in current machine learning-based approach to determine the recognition model. However, training dataset collection is burdensome to users. Moreover, in some case, it is difficult for people to hold a predefined posture. Therefore, a better solution is to get rid of training dataset, which will aid the wider adoption of system. In order to achieve it, one possible way is to leverage the domain knowledge in human body to model sitting posture. We can simulate different human bodies and then come up with a general model which is able to map different pressure distribution to corresponding postures. Further exploration is required to understand the feasibility of this idea.

Transfer Learning. Another question is whether we can apply the recognition model learned in a particular chair to another chair. In our experiment, the chair top is made of clothes and soft. We think it is feasible to transfer the training model to new chair as long as it is within the same application domain. (e.g., same user and same posture set.) However, people may exhibit different sitting habits across various chairs, resulting

in different pressure distribution. Therefore, further studies are needed to understand the impact of different chairs towards learning models.

3.8 Summary

In this work, we present Postureware, a practical solution for sitting posture recognition based on pressure sensor array. To reduce system cost, we propose an effective sensor placement solution that can achieve 99.6% ten-fold cross validation accuracy based on 10 pressure sensors. To ensure the system generalization capability, we develop a robust sitting posture recognition framework, including user-invariant features and an ensemble learning classification model. Our system can achieve 84.7% generalization accuracy with 10 sensors. Finally, we present three prototype applications that demonstrates the system enables unhealthy sitting posture monitoring and sitting posture-based interaction.

In future work, we plan to extend the sitting posture recognition system to perform inter-disciplinary researches. In particular, we are interested in three directions: affection recognition, rehabilitation and self-improvement. First, we will investigate the correlation relationship among sitting behaviours with human emotion. By leveraging the correlation relationship, we can develop a sitting posture-based prediction model to recognize user emotion. Second direction is rehabilitation, which can help people with chronic spinal cord or traumatic brain injuries, to recover the sitting balance via game-based exercise. The third direction is self-improvement, which can provide feedback for dancers or meditators to maintain a good posture.

Chapter 4

An Unobtrusive Eating Behaviour Monitoring System using Smartwatch and Smartphone

In this chapter, we study the problem of eating behaviour recognition based on smartwatch and smartphone. Section 4.1 presents the overview of the thesis. In Section 4.2, system requirements and challenges are described. Section 4.3 and 4.4 describe the system framework and system design respectively. The details of system implementation are presented in Section 4.5 and evaluation results are described in Section 4.6. Finally, we discuss the research issues in Section 4.7 and conclude this work in Section 4.8.

4.1 Overview

Eating behaviour plays a critical role in people's health. Clinical studies suggest that eating patterns such as eating frequency, the temporal distribution of eating events during a day, breakfast skipping, and eating fast food, are related to obesity [MGW+12]

[GGAAB+13]. According to world health organization, more than 1.9 billion adults are overweight and over 600 million of them are obese by 2014 [WHO15]. The centre for disease control believes that the best way for prevention and treatment of such problems is to monitor behaviour and environment settings. Thus, monitoring eating behaviour has great potential for helping people controlling weights and improving health conditions, and thus drawing increasing attention from both industry and research communities.

Several recent attempts include HAPIfork [for14] and sensing fork [KLT+14]. They are electronic forks which rely on the motion sensors such as accelerometer and gyroscope to measure eating speed and eating schedule, etc. However, they require users to use those electronic utensils in each meal, which is impractical and brings a lot of inconvenience. Rather than tracking the utensil, another research aspect for eating behaviour monitoring is to measure the wrist motion. Dong et al. [DHM09] propose BitCounter, a system to detect and count bites of food taken based on a gyroscope sensor embedded in a wrist-worn device. However, there are several limitations of this work. First, the device requires the user to turn on the system during eating, which brings about inconvenience. Second, it is very energy-consuming since it collects data from gyroscope. Third, this work does not consider cuisine and food type detection.

This work presents iEat, a practical system to monitor an individual's eating behaviour using off-the-shelf smartwatch and smartphone. iEat is truly non-obtrusive: users just need to wear it on the dominant hand and the eating behaviour can be automatically detected and recorded. Besides, iEat is a customised eating behaviour monitoring system since different people have different ways of eating the same food. In particular, iEat uses the built-in accelerometer of the smartwatch to detect the eating events, including eating schedule, food cuisine and food item. The reason why we focus on accelerometer is that accelerometer has very low power consumption (10% power consumption of gyroscope), and thus enabling long-term monitoring. By providing the detailed eating behaviour information, iEat could be a promising platform for the analysis of eating pat-

tern and eating diagnosis. In particular, iEat enables the user to be aware of their eating habits, which has great potential to control weights and improve health conditions. Furthermore, iEat can provide eating behaviour data over a long period of time, which allows behaviour research to examine the relationship between eating behaviours and other behaviours.

During the design of iEat, we need to address two major challenges. First challenge is how to enable continuous eating behaviour monitoring given the energy constraint of smartwatch and smartphone. To resolve this challenge, our main idea is to reduce data collection time and data analysis computation. First, we present a context-aware data collection method which allows the smartphone to trigger the data collection of smartwatch based on user's environmental and behavioural context (e.g., inside a restaurant), so as to reduce energy consumption. Compared with smartwatch, smartphone is more sufficient in battery capacity and is able to support context detection. Second, we adopt a light-weight decision tree-based classification algorithm to further reduce the energy consumption of the system. Second challenge lies in how to accurately classify different eating events such as eating and non-eating, food cuisine and item based on wrist acceleration signals. We extract fragment-based, event-based and segment-based features which can effectively characterise the eating behaviour patterns.

The primary contributions of this work are as follows.

- We present a context-aware data collection method to conserve energy based on the collaboration between smartwatch and smartphone.

- We introduce a new set of accelerometer features that can capture key characteristics of eating behaviours.

- To further reduce the energy consumption of the system, we present a light-weight classification algorithm for eating event classification.

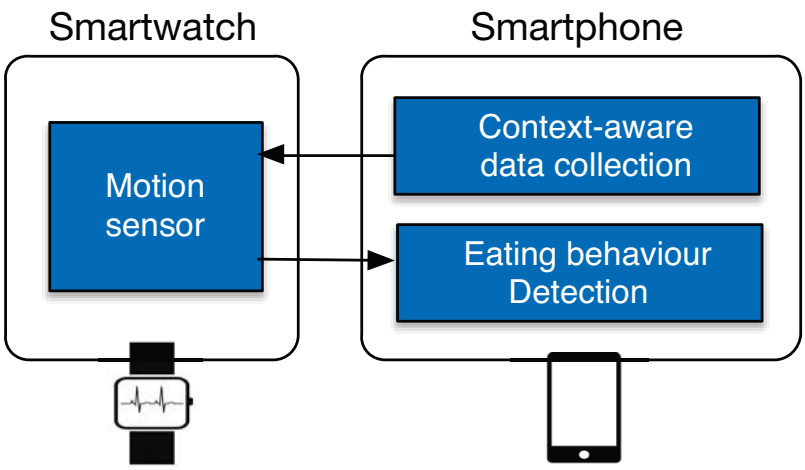


Figure 4.1: The architecture of iEat system

4.2 System Requirements and Challenges

iEat is designed to provide a long-term fine-grained eating behaviour monitoring. Specifically, we think iEat should meet the following requirements: (1) iEat needs to provide fine-grained eating behaviour information, including: eating schedule, cuisine, food items, etc. (2) The system should be energy-efficient due to the energy constraint of smartwatch and smartphone.

To meet the above mentioned requirements, two major research issues requires to be addressed. First, iEat needs to accurately distinguish eating and non-eating activities, classify different cuisines and food items solely based on the wrist acceleration signals. However, people behave differently at different time even having the same food. Thus, it is nontrivial to capture the effective features that characterize the eating motion patterns.

Second challenge lies in how to enable long-term monitoring given the energy constraint of smartwatch and smartphone. In our experiment, continuously collecting accelerometer data from smartwatch (Moto 360 Smart Watch) in 5 Hz for one hour will consume around 20 percent of battery life. Therefore, iEat is expected to have an energy-efficient data collection method and a light-weight data analysis algorithm.

4.3 System Framework

The primary objective of iEat is to build a system for long-term eating behaviour monitoring, which is able to detect fine-grained eating-related events. The architecture of iEat system is shown in Figure 5.2. The system relies on the coordination between smartwatch and smartphone. Smartwatch is responsible for collecting data from accelerometer and send to the smartphone. Smartphone accounts for triggering the start and the end of data collection of smartwatch and performing eating behaviour detection.

Context-aware data collection. To enable long-term monitoring, energy-efficient data collection is indispensable. In this work, we propose a context-aware data collection method. The main idea is to trigger data collection of smartwatch according to the environmental and behaviour context of users. For example, only when people are detected to be still and inside a dining place, will the data collection of smartwatch be turned on.

Eating behaviour detection. This is the major component of the system. Figure 4.2 illustrates the eating behaviour detection framework. First, the acceleration signal is sampled at the frequency of 25 Hz from the accelerometer of smartwatch. Second, the acceleration signal is fed to the preprocessing module. Preprocessing module will divide the signal into a series of frames and segments. Then the frame will be fed into feature extraction module, where different categories of features will be extracted. Finally, the extracted features will be fed to eating event detection module where the eating-related events will be detected.

iEat, as a portable platform for long-term eating behaviour monitoring, has the potential to help user improve their health and lose weight. For example, by providing the detailed eating habit information, iEat enables the user to be aware of their unhealthy eating habit and irregular eating schedule, so as to eat slow and at regular intervals. Furthermore, iEat can provide people's eating behaviour data over a long period of

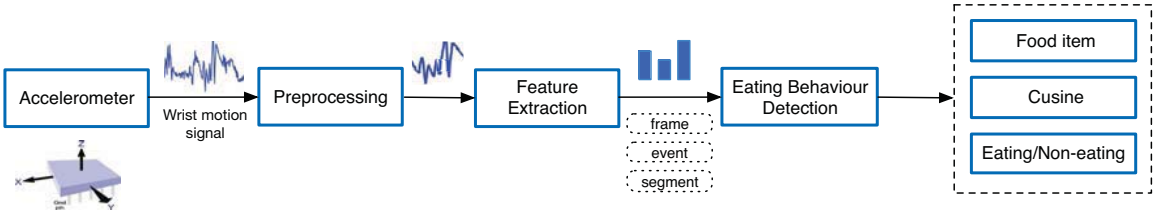


Figure 4.2: The framework of eating behaviour detection

time, which allows behaviour researchers to examine the association between eating behaviours and other behaviours.

4.4 System Design

4.4.1 Context-aware data collection

Our study shows that around 20 percent of the battery capacity is consumed after collecting accelerometer data from smartwatch (Moto 360 Smart Watch) in 5 Hz for one hour. Thus, if accelerometer is continuously sampled all day long, the battery of smartwatch will be depleted very soon, which compromises the usability of the system.

In order to enable long-term eating behaviour monitoring with smartwatch, an energy-efficient data collection method is of great necessity. The ideal scenario is that the data collection is turned on when user is eating, and turned off when user finishes eating. However, the challenge lies in how can we detect whether user is eating or not without collecting the accelerometer data from smartwatch. To tackle the above-mentioned challenge, we propose a context-aware data collection algorithm which triggers the data collection of smartwatch through smartphone according to user's behavioural and environmental context. We observe that during eating events, most people are sitting either in the restaurant or at home. Therefore, in the proposed context-aware data collection algorithm, only when people are detected to be still and inside a dining place, will the data collection of smartwatch be turned on.

Algorithm 3: Context-aware data collection algorithm

```
1  l: current GPS location
2  L_r: a set of dining places
3  while True do
4      if Walking == True then
5          Turn on GPS of smartphone;
6          Turn off data collection of smartwatch;
7      else
8          Turn off GPS of smartphone;
9          if l ∈ L_r then
10             Turn on data collection of smartwatch;
11         end
12     end
13 end
```

However, how to detect whether people are still or inside a dining place using smartphone? The naive way is to turn on the GPS of smartphone all the time and detect whether user is in a restaurant or not. However, sampling GPS data is very energy-consuming. To conserve energy, we synergize both step detector and GPS of smartphone to infer user's behavioural and spatial context. The key idea is to turn on GPS only if people is walking.

The proposed context-aware data collection method is summarised in Algorithm 3. To detect whether people is walking or not, we make use of the step detector embedded in smartphone (Nexus 5), which is very energy-efficient. Once the people is detected to be walking, the system will turn on the GPS of smartphone to attain the location. Meanwhile, the data collection of smartwatch will be turned off. If the people is detected to be still, then the GPS sensor will be turned off. In this case, if the current location of people belongs to one of the dinning places, then the data collection of smartwatch is activated.

4.4.2 Preprocessing

The objective of preprocessing is to divide the time series of data signal into frames and segments. Each frame or segment is a fraction of the original signal in a time window, which enables effective feature extraction. To extract frame and segment, we use a non-overlapping sliding window with a duration of 2 seconds, 30 seconds. At a sampling frequency of 25Hz, a single frame contains around 50 data samples and a single segment has 750 seconds. In particular, frame can capture the short-term motion patterns, while segment can capture the long-term motion patterns. In fact, the window length of a frame affects the performance of feature extraction and ultimately influences the system accuracy[PGK+09]. If the window length of a frame is too short, then a frame contains less distinctive information. The value of window length will be discussed in the evaluation section.

4.4.3 Feature Extraction

After the sensor data has been pre-processed, we then extract features from frames or segments, so as to capture the behaviour patterns of different eating activities. In particular, we extract features based on three levels of granularity: frame-based, event-based and segment-based. In the following, we will present the details of three categories of features.

Frame-based Features

The frame-based features are extracted from a single frame. For each frame, we will extract 36 kinds of features from three-axis acceleration data and the total number of features of each frame is 108. The frame-based features include heuristic features (e.g., SMA, RMS and mean rectified value), time-domain features (e.g., interquartile range, integral and zero-crossing rates), frequency-domain features (e.g., five first FFT com-

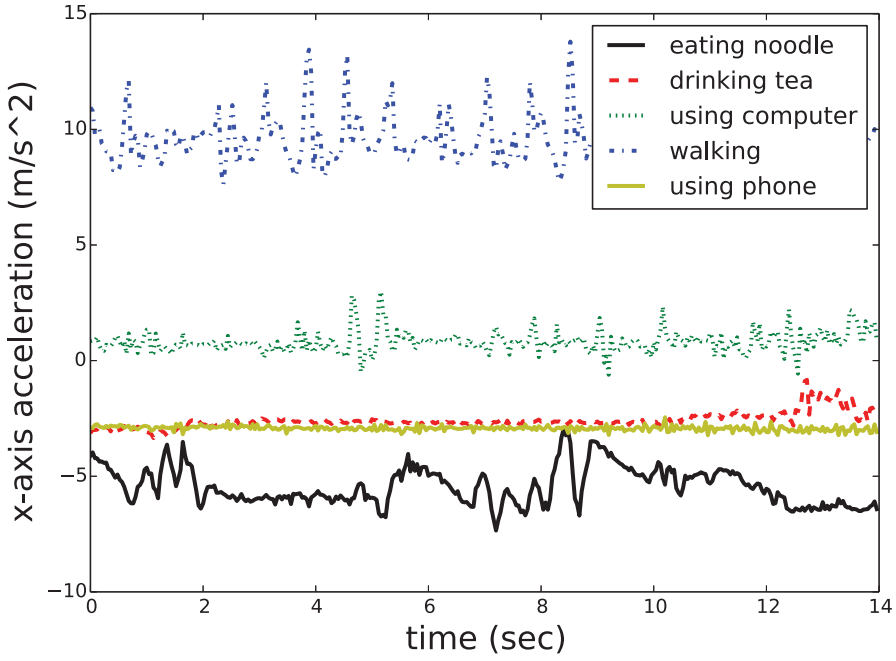


Figure 4.3: Wrist acceleration signals of eating and non-eating activities

ponents, energy and spectrum peak position) and wavelet-based features (e.g., DWT, Haar and wavelet magnitude). The frame-based features are able to capture characteristics of short-term wrist motion patterns of different eating activities. As shown in Fig. 4.3, non-eating activities such as walking and using computer have higher and positive acceleration in the x-axis direction of smartwatch, which indicates that mean value of x-axis acceleration is a discriminative feature to classify eating and non-eating activities.

Event-based Features

In addition to frame-based features, we extract event-based features that characterize the patterns of each eating cycle. In this work, one eating cycle includes picking up the food towards the mouth and then putting the hand down. We have discovered that when the food is being picked up, the wrist acceleration value will decease. Conversely, when the hand is lowered, the wrist acceleration value will increase. Thus, an eating

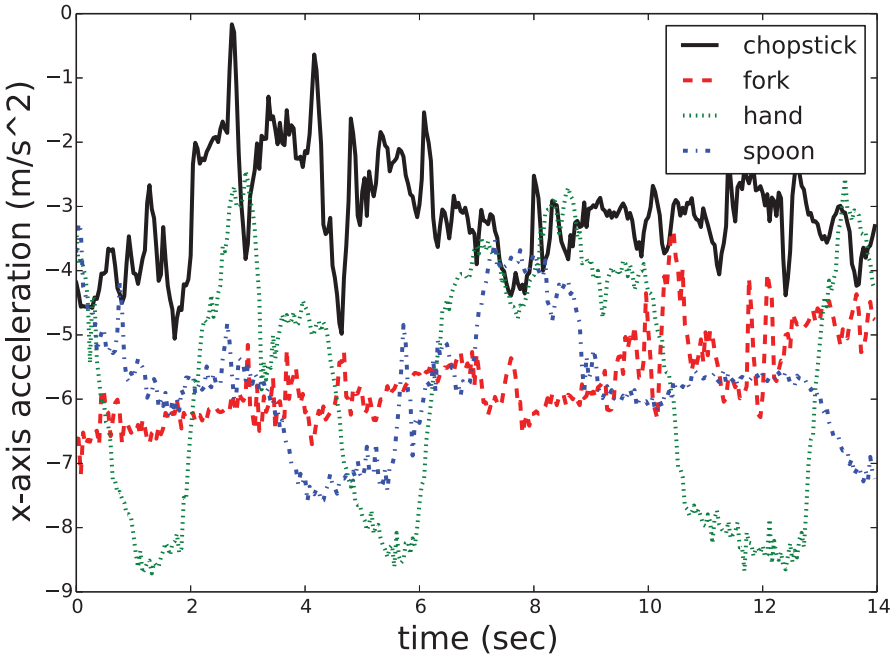


Figure 4.4: Wrist acceleration of using different utensils

cycle can be defined as an event that contains an interval of decreasing acceleration followed by an interval of increasing acceleration (see Fig. 4.4 and 4.5).

To capture the eating event-based features, we identify the valley area that associates with an eating cycle. We extract the valley areas by using a stream-based event detection algorithm to identify the significant changes in three-axis acceleration. Once a significant decrease has been detected, the corresponding time instant will be marked as the starting boundary of the valley area. We buffer the subsequent sensor signals until a significant increase in the three-axis acceleration is observed, and the corresponding time instant will be marked as the end boundary of the valley area. Currently we set the threshold to be 2 m/s^2 for identifying the significant changes. The value of significant change threshold will be discussed in the evaluation section. Once the starting and ending boundaries have been identified, we extract a set of statistical features that characterize the valley area, including: volume, intensity, length, kurtosis and skewness. Note that, there exists more than one eating cycles within a single segment. Thus, we average the value of each even-based feature within a segment.

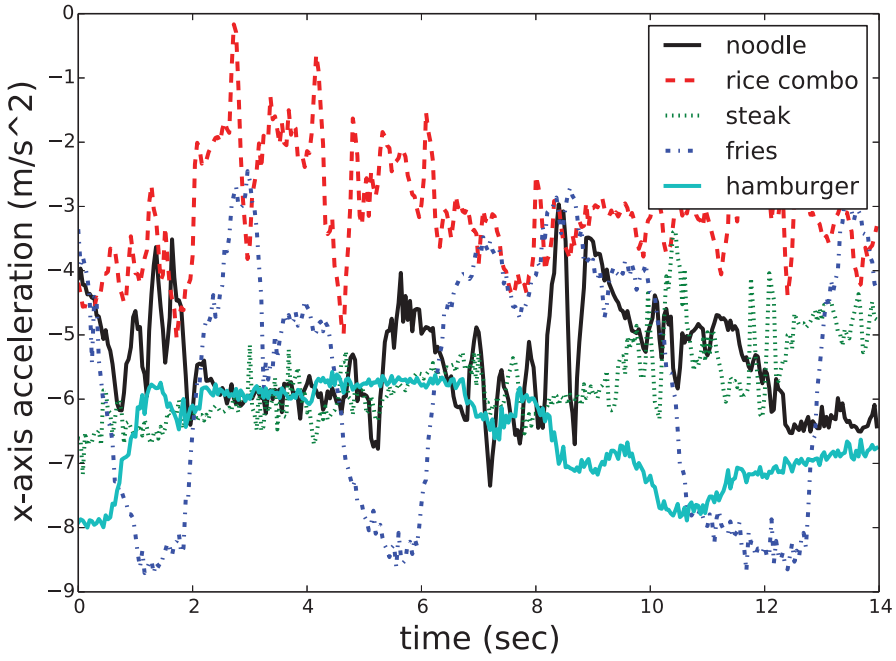


Figure 4.5: Wrist acceleration of eating different foods

Segment-based Features

On top of event-based features, we extract segment-based features that characterize the patterns of eating cycles over the segment. The segment-based features include the frequency and variance of individual event-based features. In total, we consider 12 segment-based features. From Figure 4.4 and 4.5, we can observe that holding different utensils or having different food items results in different patterns of event-based features. For example, having rice combo shows a more frequent occurrence of eating cycle than having noodle and hamburger.

In summary, we extract 108 frame features, 15 peak features and 12 segment features, resulting in an overall feature space consisting of 135 features. All the features are shown in Table 4.1.

Domain	Features
Heuristic	signal magnitude area, root mean square, mean rectified value
Time	max, min, mean, median, variance, std, skewness, kurtois, interquartile range, integral, mean-crossing rate, mean absolute deviation, correlation of each pair of axes
Frequency	FFT coefficient 1,2,3,4,5 HZ, spectral energy, spectral entropy, spectrum peak position, magnitude of FFT (first five component), DCT
Wavelet	DWT, Haar, wavelet entropy, wavelet magnitude
Event	volume, intensity, length, kurtosis, skewness
Segment	variance and frequency of event-based features

Table 4.1: Full feature list

Feature selection

Features extracted in the previous section could be redundant or irrelevant, which might degrade the system accuracy. Thus, to select the most effective features is of great necessity to improve the system accuracy. Here, we apply recursive feature elimination (RFE) [GWBV02] to conduct feature selection, since RFE can provide good performance with moderate computational efforts. The goal of recursive feature elimination is to select features by recursively considering smaller sets of features given an external estimator that assigns weights to features. The weights of features can be assigned with the coefficients of an estimator (e.g., linear model). First, the whole set of features will be trained using the estimator and then the weights will be assigned to each of them. Then, the feature with the smallest absolute weight will be pruned from the current set of features. The above procedure will be repeated recursively until it reaches the desired number of features to select.

4.4.4 Eating behaviour detection

Classifier

After the original signals have been transformed into the feature space, we need to train a classifier to separate the data of different classes. Different classifiers have pros and cons. Here, we choose decision tree as our main classifier due to its simplicity and high interpretability. In our evaluation section, we will compare decision tree with other popular classifiers such as naive bayes, support vector machine and adaboost ,etc.

Decision tree is a supervised learning classification model that is based on a set of decision rules for inference. Given the training set, a set of if-then-else decision rules can be learned to predict the value of a target variable. In general, the complexity of the model is associated with the depth of the tree. Decision tree model is very easy to implement and the algorithm complexity is linear to the depth of the tree.

Frame-based and Segment-based Classification

For eating and non-eating classification,we conduct classification on the basis of a frame. This is because a frame-based classification enables us to capture the start time and the end time of eating accurately and timely.

For cuisine and food classification, we conduct segment-based classification. In particular, we have two classifiers: frame-based and segment-based. Frame-based classifier is trained using frame-based features, while segment-based classifier is learned using event-based and segment-based features. Given a test segment, it will be first labeled using the segment-based classifier. Next, this segment will be divided into a number of frames and each frame will be labeled using the frame-based classifier. Finally, the final classification result is attained by averaging the results of the two outputs.

Domain	Features
Dominant hand	average rectified value, median, std, zero cross rate, mean absolute deviation, spectral entropy
Non-dominant hand	sma, integral, DWT('haar')

Table 4.2: Informative frame-based features for eating and non-eating classification

Eating and Non-eating Classification

The primary task of the eating and non-eating classification is to distinguish eating and non-eating activity on the basis of a frame. On top of the classification result, we can infer the start time and the end time of an eating activity. Intuitively, the signals are more informative when the smartwatch is worn in the dominant hand than non-dominant hand. However, we observe some interesting facts that allows us to leverage the wrist motion signals to identify eating activity even when the smartwatch is worn in people's non-dominant hand. There are two observations. First, people normally use domain hand to use phone, but they will use non-dominant hand to use phone while eating. Second, they tend to put the non-dominant hands under the table while eating.

The effective features after feature selection are listed in Table 4.2. In particular, we classify the features into two categories: dominant hand and non-dominant hand. The most informative dominant hand features include: average rectified value, median, std, zero cross rate, mean absolute deviation, spectral entropy. This is due to the significant changes of acceleration and deceleration of wrist signal while eating. The non-dominant hand features consist of sma, integral, DWT(haar), which is able to capture the motion patterns of non-dominant hand during eating activities.

To better convey the intuition, we plot the data points in feature space (see Fig.4.6). In particular, mean x value, mean x square value and variance of x value are selected as the main features. Obviously, eating and non-eating data points are very separable

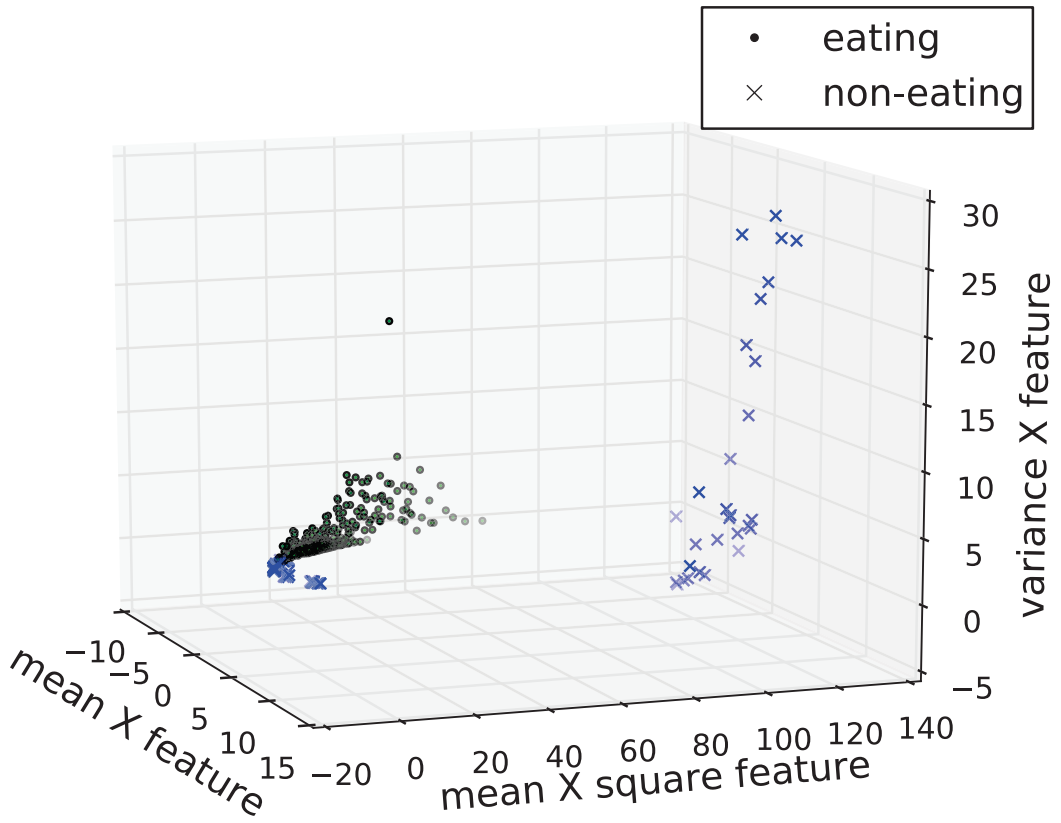


Figure 4.6: Eating and non-eating data points in three-dimension frame-based feature space

even under three-dimension feature space, which indicates that eating and non-eating classification can achieve high accuracy using simple classifier.

Cusine Classification

The cuisine classifier aims to distinguish different cuisines based on the frame-based, event-based and segment-based features. In particular, we focus on the classification of eastern cuisine, western cuisine and fast food. The intuition behind the cuisine classification is that we hold different utensils when we have different kinds of cuisines, which results in different patterns of the wrist motion. Here, we transform the cuisine classification into utensil classification problem. In particular, we classify three kinds of

utensils: chopsticks, fork, spoon and hand.

Domain	Features
Frame-based	variance, root mean square, average rectified value, std, zero cross rate
Event-based	volume, intensity, skewness
Segment-based	variance of event-based features in volume, intensity, skewness

Table 4.3: Discriminative features for cuisine classification

Table 4.3 lists the selected features which have the highest discriminative capability for cuisine classification. The effective frame-based features are variance, root mean square, average rectified value, std and zero cross rate. The selected frame-based features mainly characterize the variance and intensity of the wrist acceleration signal. The key event-based features include volume, intensity, skewness of the valley area, while the main segment-based features cover the variance of area features in volume, intensity, skewness. Fig. 4.7 shows how data points scatter in the feature space when the volume, intensity, skewnes of valley area are selected as major features. Clearly, the data within the same classes are close to each other, while the data from different classes are very separable.

Food Item Classification

The objective of food item classifier is to identify a predefined set of food items. Currently, the classifier supports recognising seven common food items: noodle, rice combo, dim sum, congee, steak, hamburger, fries. The food items are selected based on the survey from campus students.

Table 4.4 lists the most important features that can distinguish different categories of foods. The key frame-based features are variance, root mean square, average rectified value, std and mean absolute deviation. Obviously, the variance and intensity are

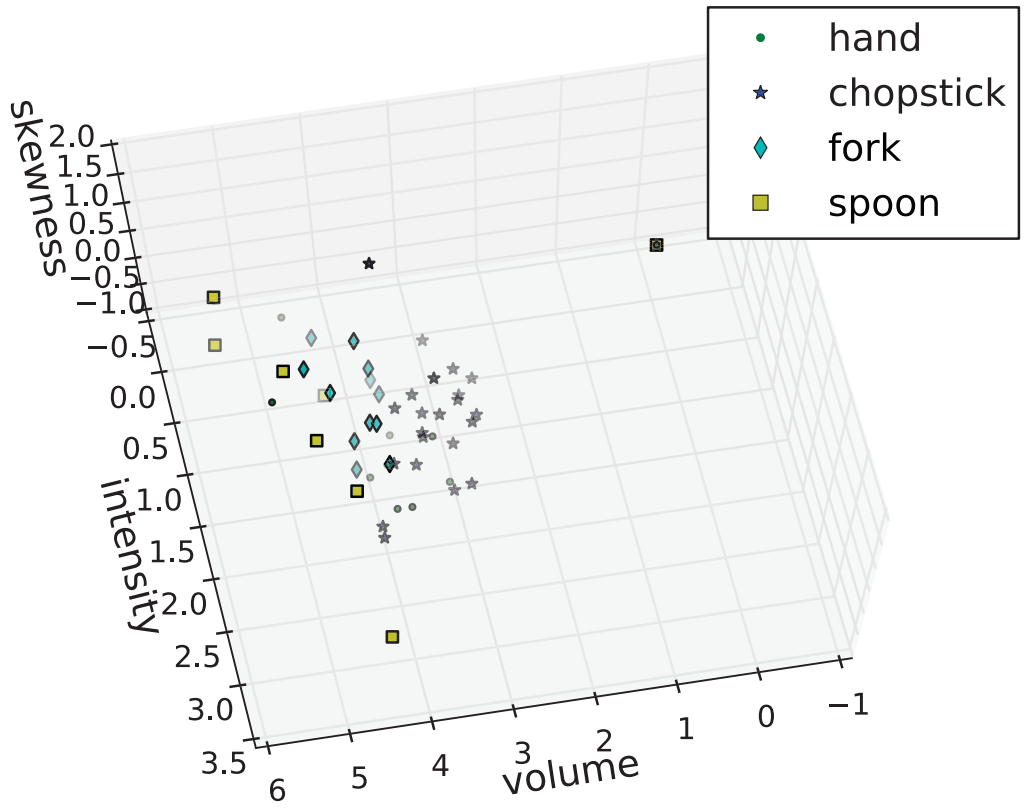


Figure 4.7: Utensil data points in three-dimension event-based feature space

effective to separate different kinds of foods. Similar to cuisine classification, the event-based features include volume, intensity and skewness, while segment-based features include variance of event-based features in volume, intensity, skewness.

To illustrate the discriminative capability of selected segment-based features, we plot the data samples over three dimension feature space as shown in Fig. 4.8. The variance of volume, intensity and skewness are selected as the major features. We can observe from the figure that data from eating congee and noodle are quite scattered, while data from fries, steak and rice combo are very close to the data of the same label. It is worth emphasizing that he data of different classes will be more separable when frame-based, event-based and segment-based features are combined. The detailed system performance will be shown in the evaluation section.

Domain	Features
Frame-based	variance, root mean square, average rectified value, std, mean absolute deviation
Event-based	volume, intensity, skewness
Segment-based	variance of event-based features in volume, intensity, skewness

Table 4.4: Effective features for food item classification

4.5 Implementation

iEat is implemented on both smartwatch (Moto 360) and smartphone (Nexus 5). The android version of the smartphone is 5.0.1. The smartphone keeps monitoring the user's behaviour context. Once the user is detected to walk and get closer to the dinning place, and then the data collection of smartwatch will be triggered. The built-in accelerometer sensor of smartwatch is sampled at 25 Hz.

The samples are buffered and segmented into frames with the duration of 2 seconds. According to the features extracted from the signals, iEat classifies the eating-related activities, including: non-eating and eating classification, cuisine classification and food item detection.

4.6 Evaluation

In this section, we evaluate the performance of iEat by conducting extensive experiments. Our primary results demonstrate that iEat is able to accurately monitor the user's eating behaviours in an energy-efficient manner.

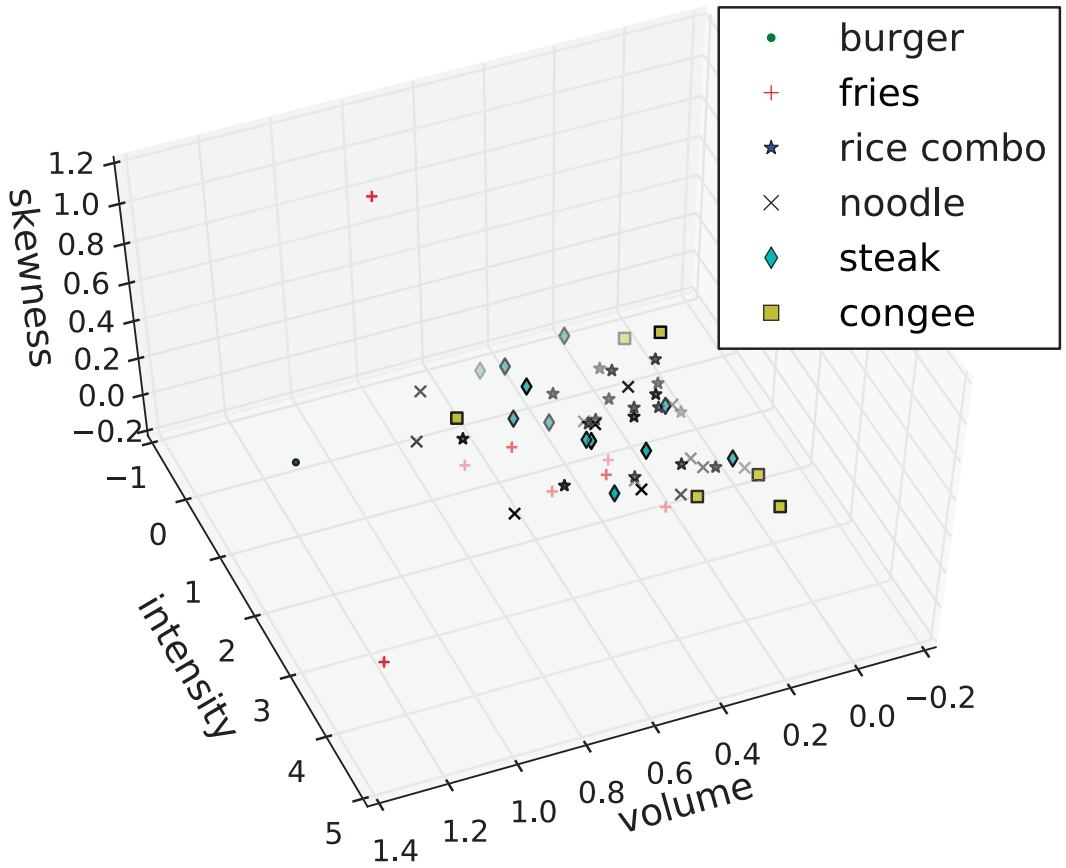


Figure 4.8: Food item data points in three-dimension segment-based feature space

4.6.1 Datasets

To evaluate the system performance, we have collected data from two subjects. All the experiment subjects come from the local university. During the data collection, each subject wears the Moto 360 smartwatch and the smartwatch is paired up with a Nexus 5 smartphone. To evaluate the difference between wearing smartwatch on the dominant and non-dominant hand, the subjects are asked to wear the smartwatch on both dominant hand and non-dominant hand.

In particular, we have collected the following datasets. First, we have collected the accelerometer data of smartwatch when individuals have different types of food in a natural setting for two weeks. Each subject will eat eastern food, western food and fast food. Specifically, eastern food includes: rice combo, noodle, congee, dim sum; while

western food includes steak and fast food includes french fries and hamburger. The ground truth annotations were made by the subject. In addition, to evaluate the performance of eating and non-eating classification, we also collect the accelerometer data of smartwatch when people perform non-eating activities. We focus on nine common non-eating activities: drinking, using computer, using phone, reading, writing, walking, downstair and upstair, etc.

Moreover, the experiment subjects are required to record the start time and end time of eating events, so that we can measure the effectiveness of the proposed data collection method. The data collection in this work is more like a software testing rather than human subject study. The dataset does not involve any privacy issues. Besides, all the subjects are volunteered to test the system.

4.6.2 Experimental Setting

To evaluate the performance of eating behaviour detection, we use leave-one-out cross-validation as the accuracy metric. Here, 50% of dataset will be used for training and the rest 50% will be used for test. The comparison among different classifiers is based on computation results from scikit-learn [sci15].

4.6.3 Performance evaluation

We consider three aspects in our evaluation: system accuracy, energy efficiency and micro-benchmarks. With regard to system accuracy, we focus on evaluating the accuracy of eating and non-eating classification, accuracy of cuisine classification, accuracy of food item classification as well as the generalization performance of classifiers. Energy efficiency evaluation consists of energy consumption of data collection in both smartwatch and smartphone, and the effectiveness of data collection method. Finally, we present a set of micro-benchmarks, which evaluate the impact of features, window

Classifier	Dominant hand	Non-dominant hand
Decision Tree	0.95	0.97
Naive Bayes	0.59	0.94
SVM(linear)	0.92	0.84
SVM(rbf)	0.95	0.78
Adaboost	0.87	0.64

Table 4.5: Comparison among different classifiers in eating and non-eating classification

length of a frame, sampling rate and the threshold for eating cycle detection towards system performance.

System Accuracy

Accuracy of eating and non-eating classification The objective of this experiment is to measure the accuracy of classifying eating and non-eating activities based on the wrist motion signal. In this experiment, the eating dataset includes the data collected from two subjects who tried out eight types of food: rice combo, noodle, congee, dim sum, steak, french fries and hamburger. The non-eating dataset consists of eight non-eating activities, including: drinking, using phone, using computer, writing, reading, walking, upstair, downstair, etc.

	Precision	Recall	F1-score
Eating (dominant hand)	0.95	0.95	0.95
Eating (non-dominant hand)	0.97	1	0.97

Table 4.6: Eating and non-eating classification performance

The results of this evaluation are shown in Table 4.5. We compare decision tree with other popular classifiers such as Naive Bayes, SVM (linear kernel), SVM (rbf kernel) and Adaboost. Despite its simplicity, decision tree achieves the highest accuracy among all the candidate classifiers. Specifically, eating and non-eating classification

can achieve 95% accuracy when the smartwatch is worn on the dominant hand and 97% accuracy when the smartwatch is worn on the non-dominant hand. The best baseline classifier is SVM (linear), which attains 92% precision and 84% recall.

Table 4.6 shows that the precision, recall and F1-score of eating and non-eating classification based on decision tree classifier. The result demonstrates that the eating and non-eating frames can be accurately distinguished. To our surprise, the classification accuracy can be over 95% even when the smartwatch is worn on the non-dominant hand. The result reveals that the wrist motion signal of non-dominant hand is very informative and can be exploited to derive eating activities.

	Precision	Recall	F1-score
Decision Tree	0.96	0.96	0.96
Naive Bayes	0.72	0.82	0.76
SVM (linear)	0.96	0.96	0.96
SVM (rbf)	0.94	0.94	0.94
Adaboost	0.90	0.90	0.90

Table 4.7: Comparison among different classifiers in utensil classification

Accuracy of cuisine classification We examine the performance of cuisine classification in this section. Since different cuisines usually associate with using different utensils, we transform the cuisine classification problem into utensil classification. We compare the performance of decision tree against other classifiers (see Table 4.7). The result shows that both decision tree and SVM (linear) achieve 96% accuracy. Figure 4.9 illustrates the classification performances of different utensils. Using hand and chopsticks can be recognized with 100% recall rate. Fork and spoon can be detected with a high precision rate but with a comparatively low recall rate.

Figure 4.10 presents the confusion matrix of utensil classification. From the confusion matrix we can find that the most challenging classification task is to distinguish chopstick from fork and spoon. The challenge is caused by the close similarity of the features for using these utensils.

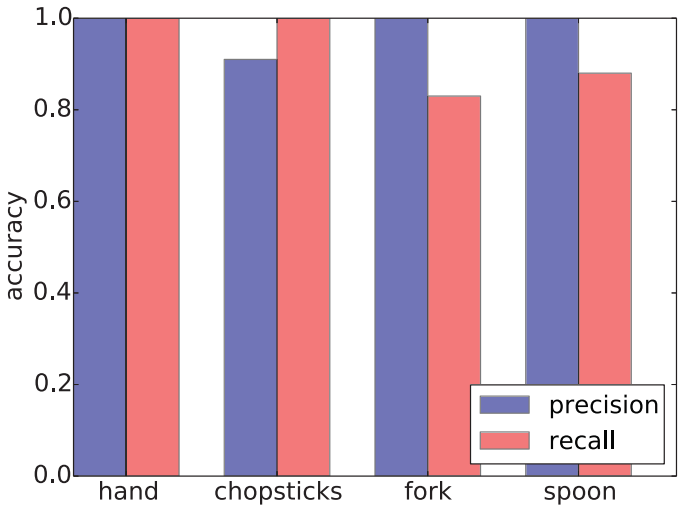


Figure 4.9: Classification performances of different utensils

	Precision	Recall	F1-score
Decision Tree	0.93	0.92	0.92
Naive Bayes	0.55	0.57	0.53
SVM (linear)	0.92	0.92	0.91
SVM (rbf)	0.92	0.92	0.91
Adaboost	0.56	0.47	0.43

Table 4.8: Food classification accuracy with different classifiers

Accuracy of food item classification This section presents the evaluation results of food item classification. The food classification accuracy together with a comparison against the solutions using other classifiers are illustrated in Table 4.8. The proposed solution based on decision tree classifier attains the highest accuracy with 93% precision and 92% recall. The best baselines are SVM (linear) and SVM (bf), which achieve slightly lower accuracy. As shown in Table 4.9, eating hamburger and fries can be recognised with very high precision and recall rate. However, rice combo, noodle and steak can be misclassified. Figure 4.11 shows the confusion matrix of food item classification. From the figure we can observe that eating dim sum, rice combo and noodle exhibit high similarity and misclassification among them occurs frequently. In particular, eating dim sum can be easily misclassified as eating noodle, and only 50% recall rate

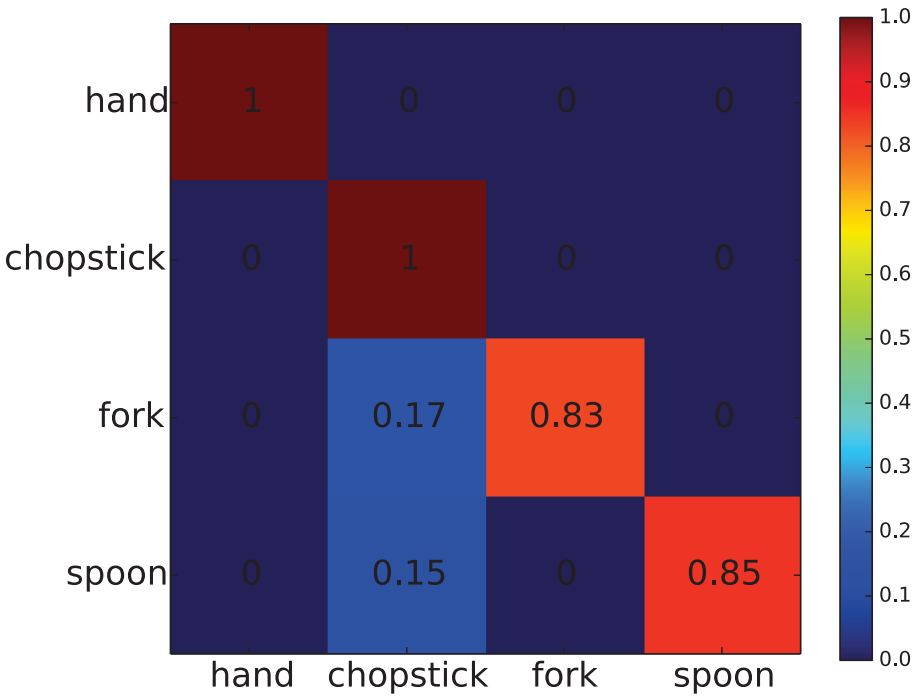


Figure 4.10: confusion matrix of utensil classification

is achieved.

	Precision	Recall	F1-score
Hamburger	1	1	1
fries	1	1	1
dim sum	1	0.50	0.67
rice combo	0.89	1	0.94
noodle	0.89	0.80	0.84
steak	1	0.83	0.91
congee	0.80	1	0.89

Table 4.9: Detailed classification accuracy of food classification

Generalization performance of classifiers We then demonstrate that the generalizabiliy of the proposed solution across different people. In this experiment, all the classifiers are first trained using the data from a particular user and then evaluated using the data from a new user. Table 4.10 shows the results of this evaluation. Based on the results, we can observe that the eating and non-eating classification is well generalized

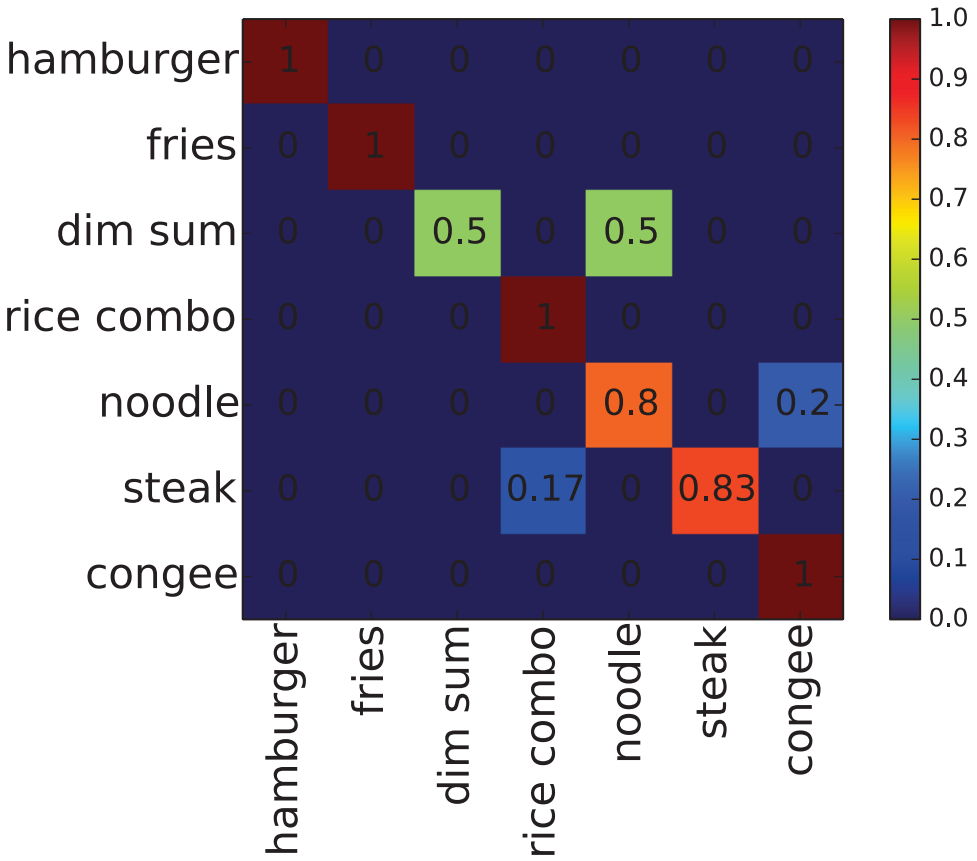


Figure 4.11: confusion matrix of food item classification

to new users, achieving 100% precision and 61% recall rate. However, the recall rate is low, which may not able to capture the start time and the end time of eating events accurately.

The results also reveal that the cuisine and food item classification achieve poor performances for new user. Specifically, cuisine classification achieves 81% precision and 43% recall rate, while food item classification achieves only 72% precision and 11% recall rate. The results demonstrate that people exhibits the similar wrist motion pattern for each eating cycle. The rationale behind this observation is that for most people, one eating cycle contains two actions: picking up the food towards the mouth and putting the hand down. However, different people might have different ways of eating certain food item. Thus, the misclassification rate is high when the system is used for new peo-

ple. But the good news is that it is a customized system and the system performance will improve by collecting more data from people.

	Precision	Recall	F1-score
eating and non-eating classification	1	0.61	0.76
cuisine classification	0.81	0.43	0.34
food item classification	0.72	0.11	0.14

Table 4.10: Generalization performance of the system

Energy Consumption

The objective of iEat is to enable long-term eating behaviour monitoring of an individual. Thus, energy consumption is a critical factor. Currently, iEat relies on a smartwatch and a smartphone. For smartwatch, the battery capacity is very limited. Here, we evaluate energy consumption of smartwatch based on the battery usage. Since the screen is turned off, accelerometer sampling and data transmission are the major sources of energy consumption of smartwatch.

Table 4.11 shows that energy consumption of accelerometer data collection and transimission in a smartwatch under different sampling frequency. We can find that continuously collecting accelerometer data for 30 minutes under 5 Hz can consume around 9% battery lifetime of smartwatch. When the sampling frequency comes to 25Hz, around 27% battery lifetime is depleted within half an hour. The above results show that continuously collecting accelerometer data from a smartwatch even under very low sampling frequency (5Hz) will deplete the energy very soon. Therefore, an energy-efficient method perform data collection is of great necessity.

In addition to smartwatch, iEat relies on smartphone for user context detection and data processing. Table 4.12 lists the sensors of smartphone that are used in our system. It shows that step detector only consumes around 2mW, which can used for long-term monitoring. For data processing, very low power is consumed due to the light-weight

decision-tree based classifier.

We evaluate the efficiency of the proposed context-aware data collection method. In this work, the dining places are limited to restaurants, since the experimental subjects do not eat at home. In the current system, GPS locations of all the restaurants within the campus area are collected manually. In the future, we will turn to the third party geographical information. Table 4.13 shows the comparison of the actual eating time and the sensor sampling time. On average, experiment subjects spend around 11 minutes on eating per meal. During each eating event, the accelerometer sampling time of smartwatch is around 25 minutes. The gap between actual eating time and data collection time is mainly caused by the fact that people waits for the food inside the restaurant. The GPS sampling time is 36 minutes between two consecutive meals. The data collection time is expected to increase when subjects eat at home, because being at home does not associate with eating activity only. The above results show that the proposed context-aware method can significantly conserve the energy consumption of system.

Time (min)	frequency (5Hz)	frequency (25Hz)
10	3%	9%
20	6%	18%
30	9%	27%

Table 4.11: Energy consumption of accelerometer sampling in smartwatch

Sensor	Power (mW)
step detector	2 mW
GPS	250 mW

Table 4.12: Energy consumption of sensors in smartphone

	Time (mins)
Actual eating time	11
Accelerometer sampling time	25
GPS sampling time	36

Table 4.13: Comparison of actual eating time and sensor sampling time

Micro-benchmarks

This section presents a set of micro-benchmarks that evaluate the performance of iEat under various feature sets, sliding window lengths, sampling rates and hands.

Impact of features This section presents the evaluation results of how the feature set impacts the classification accuracy. Figure 4.12 and 4.15 plot the eating and non-eating classification accuracy with respect to number of features when the smartwatch is worn on dominant hand and non-dominant hand respectively. From the figures we can observe that the combination of more feature does not necessarily improve the system accuracy. This is caused by the classification process of the decision tree classifier. For decision tree, the more discriminative features will be used in the decision rules earlier, while the less discriminative features will be discarded. Specifically, when the number of features is 10, the proposed method can achieve over 90% accuracy. The high classification accuracy given a few number of features demonstrates that the extracted features are distinctive. Moreover, the energy consumption of system can be reduced by extracting a few number of features.

Figure 4.18 and 4.19 plot the impact of feature sets towards cuisine and food item classification. Compared with segment-based features, the combination of frame-based features and segment-based features can significantly improve the system accuracy. The intuition is that segment-based features are able to capture the long-term motion pattern, while frame-based features capture the short-term motion pattern. Therefore, the combination of both features are better to characterize the eating behaviour patterns.

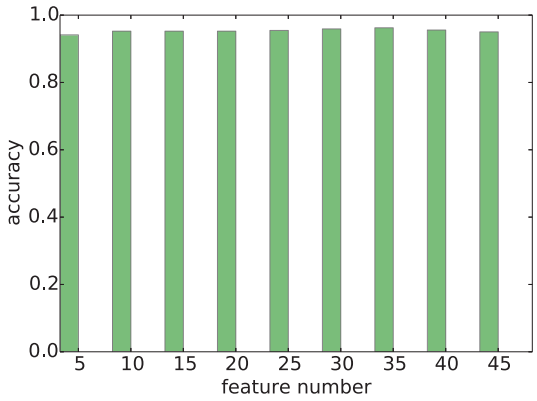


Figure 4.12: Impact of feature number on eating and non-eating classification

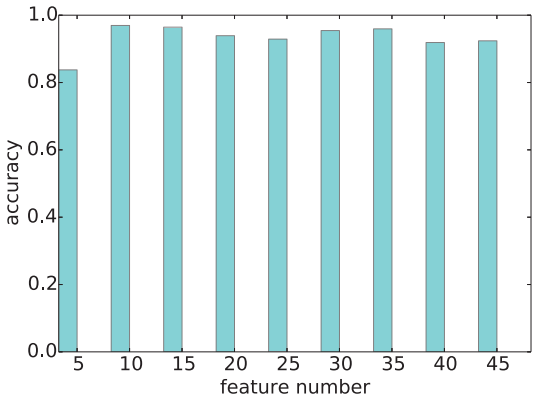


Figure 4.15: Impact of feature number on eating and non-eating classification

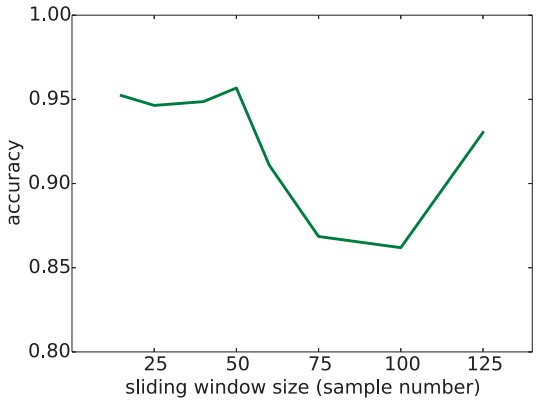


Figure 4.13: Impact of window size on eating and non-eating classification

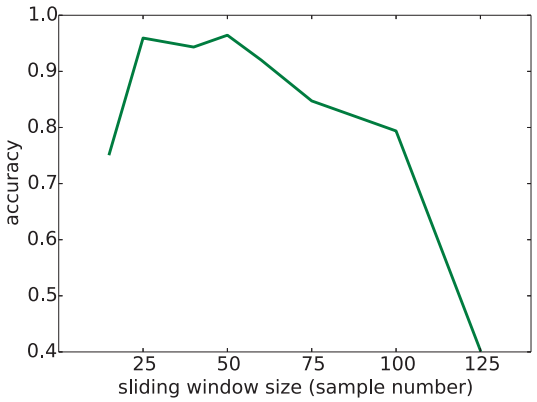


Figure 4.16: Impact of window size on eating and non-eating classification

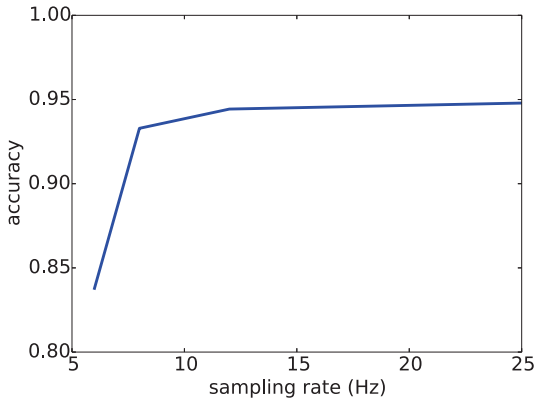


Figure 4.14: Impact of sampling rate on eating and non-eating classification (dominant hand)

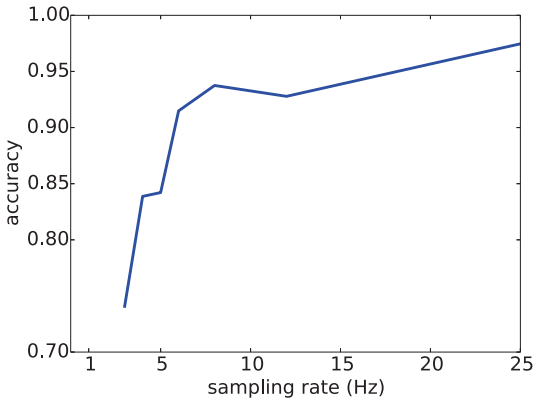


Figure 4.17: Impact of sampling rate on eating and non-eating classification (non-dominant hand)

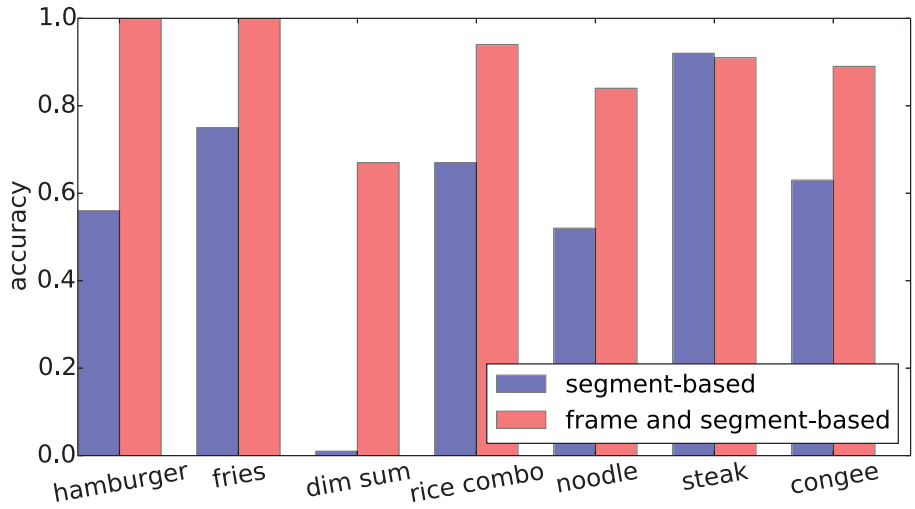


Figure 4.18: The impact of features towards cuisine classification

Impact of frame size This experiment aims to examine the impact of window size of a frame towards the system accuracy. The evaluation is conducted by setting different window sizes. Figure 4.13 and 4.16 show the result of system accuracy with respect to different frame sizes. In particular, the frame size is changed to 12, 25, 50, 75, 100 and 125. When the window size is 50, the eating and non-eating classification achieves the highest accuracy. This is due to two reasons. First, a frame with short window size contains insufficient information, and thus resulting in a degraded system accuracy. Second, the features extracted from a frame with large window size could become less distinctive. Given the sampling frequency of 25Hz, 50 data samples are equal to a 2-second signal.

Impact of sampling rate This experiment aims to examine the impact of sampling rate of the data collection towards the system accuracy. The evaluation is conducted by choosing different sampling rates. In particular, we set the sampling rate to be 5, 10, 15, 20 and 25 (Hz) respectively and evaluate the corresponding system accuracy.

Figure 4.14 and 4.17 show the result of eating and non-eating accuracy with respect to different sample rates. Figure 4.20 plots the accuracy of cuisine and food classification given different sampling rates. From above three figures, we can observe that increasing sampling rate leads to an improved system accuracy in general. In particular, the

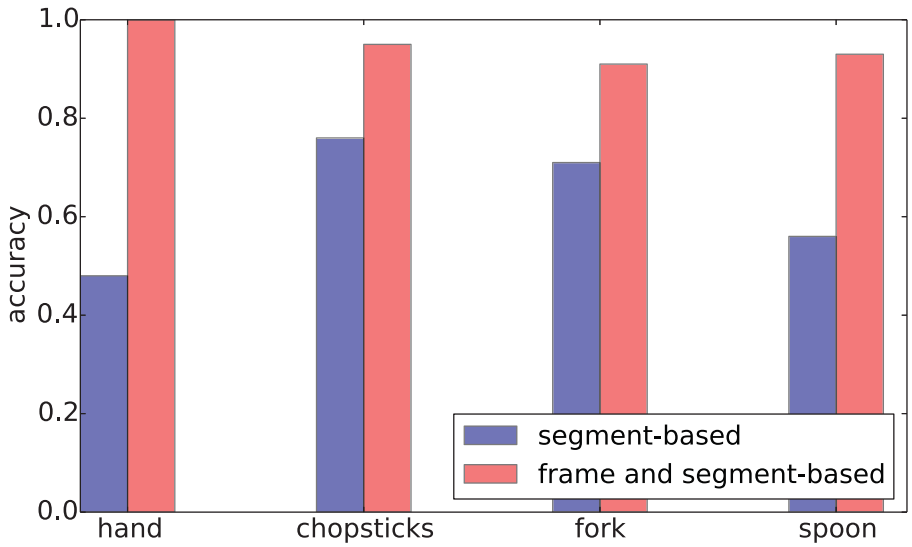


Figure 4.19: The impact of features towards food item classification

system performances under 15 Hz and 25 Hz sampling rate are very close. This is due to the fact that the wrist motion during eating is low-frequency and thus higher sampling rate does not necessarily increase the system accuracy. The results suggest that sampling rate of the accelerometer can be set as 15 Hz, which can achieve the satisfactory system accuracy and reserve energy.

Impact of threshold in eating cycle detection This experiment aims to examine the impact of change threshold in event-based feature extraction. In particular, we set the change threshold from 1 to 5 (m/s^2) and evaluate the corresponding system accuracy. In order to obtain the optimal threshold, we evaluate the system accuracy only based on event and segment-based features. Figure 4.21 shows the result of cuisine and food classification given different change thresholds. We can observe that when the change threshold is set as 2 m/s^2, both cuisine and food classification achieve the best performance. If the change threshold is too small, then the extracted features can not capture the motion patterns. However, when the change threshold is too large, a lot of distinctive information will be discarded.

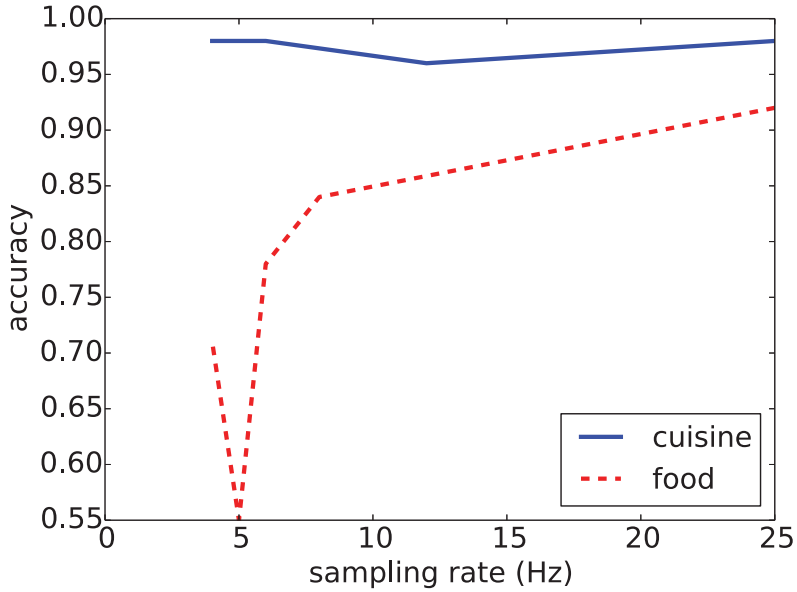


Figure 4.20: Impact of sampling rate towards cuisine and food classification

4.7 Discussion

The present work has introduced a practical system for eating behaviour monitoring based on smartwatch and smartphone. Evaluations based on the extensive experiments demonstrate that, our solution is able to achieve high accuracy of eating activity recognition such as eating and non-eating classification, cuisine and food item identification, while greatly reducing the energy consumption of the system. However, this work still has some limitations. In the following, we focus on discussing three main issues: dataset, specialty and generalisability along with integration with other devices.

Large-scale dataset Note that this work is based on a small scale of datasets. In particular, we focus on a few kinds of cuisine types: American cuisine, Chinese cuisine and fast food. Moreover, all the dataset is collected inside the restaurant, which means that some home-cooked food are not considered. In the future, we plan to collect a wider categories of datasets. Specifically, we will include the dataset when people is cooking. The intuition is that cooking different kinds of food will result in discriminative wrist motion patterns. Thus, the system accuracy is expected to increase by incorporating the motion signals from both cooking and eating. Furthermore, we will recruit

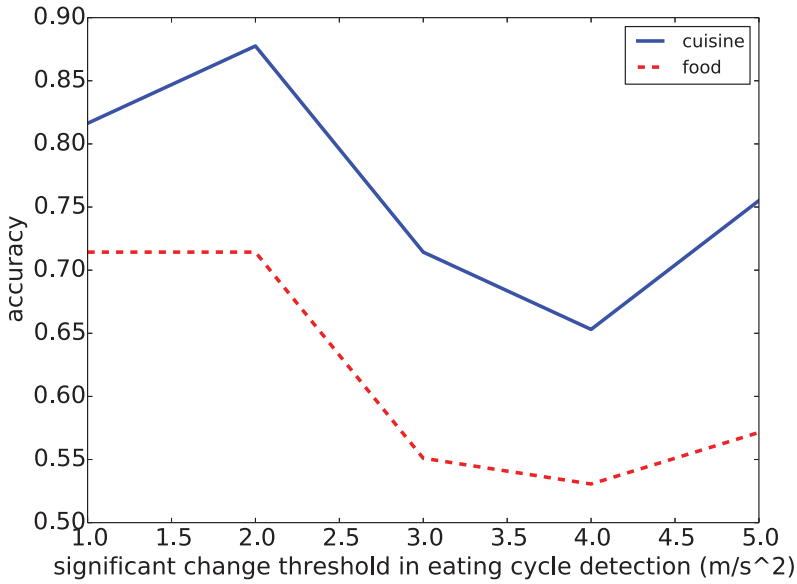


Figure 4.21: Impact of significant change threshold in eating cycle detection

more individuals for data collection.

Specialty and generalisability Due to the personal preference and culture difference, people performs eating activity in different ways. Thus, the wrist acceleration signals should be expected to be different when they are collected from a teenager and an elderly who eat the same food. In general, an eating behaviour detection model could be either general or specific to a particular user. However, it is a tradeoff between generalizability and accuracy. Usually higher generalizability results in lower system accuracy. But low generalisability compromises the usability of the system. We believe that to make the most of both worlds, group-specific classifiers can be designed, which are tailored to particular groups of individuals according to age, gender and nationality.

Integration with other smart devices The ultimate goal of this work is to offer fine-grained eating behaviour monitoring. However, this work is still on the early stage and only supports identifying a small set of food. Due to the limitation of accelerometer sensor, the current system cannot identify what kind of meat people is eating. To enable fine-grained eating behaviour monitoring, we might need to incorporate more information from other modalities. We plan to integrate google glass into the system.

Our preliminary study shows that there exists acceleration patterns of head movement when people are eating. Besides, we can make use of the camera of google glass to take pictures and conduct image processing to obtain more detailed information about the food.

4.8 Summary

This work presents iEat, a practical system to monitor an individual's eating behaviour using off-the-shelf smartwatch and smartphone. iEat exploits the accelerometer data collected from smartwatch to derive user's eating schedule, cuisine and food item. The primary contributions of our work include a context-aware data collection method to conserve energy, a novel set of accelerometer features that are able to capture key characteristics of eating motion patterns, and a light-weight eating activity detection algorithm. We evaluate our approach using real-world traces and the experimental results demonstrate iEat is able to monitor individual's eating behaviour in a non-invasive and energy-efficient manner.

By providing the detailed eating behaviour information, iEat could be a promising platform for the analysis of eating pattern and eating diagnosis. In particular, iEat enables the user to be aware of their eating habits, which has great potential to control weights and improve health conditions. Furthermore, iEat can provide eating behaviour data over a long period of time, which allows behaviour research to examine the relationship between eating behaviours and other behaviours.

Chapter 5

Accurate and Energy-Efficient Social Activity Recognition based on Smartphones

In this chapter, we study the problem of social activity recognition. This chapter is organised as follows. Section 5.1 presents the overview of this work. Section 5.2 and 5.3 describe the motivating scenarios and preliminaries respectively. Section 5.4 presents the system design. In section 5.5, we provide the evaluation details and results. We discuss the limitations and other issues in Section 5.6 and conclude this chapter in Section 5.7.

5.1 Overview

Social activities play a critical role in human's well beings. The imbalance of human's social activities may give rise to the problem of autism [Vit07] and social relationship [ELFB00]. A first step towards preventing this type of condition consists in being able to automatically recognize and record someone's social activities. For example, the

recognition of social activity enables the automatic logs of users' social activities, which can be used for the elderly to overcome cognitive decline [PBC+03] or delivering health intervention to motivate user behaviour change [KP11]. In addition, social activity recognition provides the support for data-driven social science research by automatically measuring the patterns of social activities. Moreover, mobile applications like phone interruption management [RDV11] can be augmented with the awareness of user's current social activity to provide seamless service.

A fair amount of research efforts have been devoted to activity recognition based on various sensors embedded in smartphone. Some studies utilize accelerometer to measure user's locomotion state and then derives his/her physical activities based on locomotion patterns [ZLC+08][MLEC07]. In addition to locomotion pattern, location pattern has been extracted to infer certain kinds of social activities. In papers [EP06][ZY11], authors use smartphone to obtain the location of users based on Wi-Fi/GPS, and then identify the social activities based on the context of location. However, the aforementioned approaches only consider the behaviour patterns of an individual user.

We observe that social activity is associated with a community, which inherently exhibit the behaviour patterns with respect to multiple users rather than individual users. As illustrated in Fig.1, we observe that an individual user performs certain social activity with a certain set of people, and different social activities usually involve distinctive sets of people. The observation inspires us that social proximity can be exploited as a discriminative feature for social activity recognition, which is anticipated to improve the accuracy of social activity recognition.

Based on the above observation, this work presents CircleSense, an accurate and energy-efficient Smartphone-based system to recognize social activities by exploiting the social proximity information. We focus on four categories of social activity, including: work, play, develop and connect (e.g. meeting, seminar, religion service, dating, etc.). In particular, we introduce a concept called social circle, which is defined as a set of users frequently gathering to conduct certain social activities. The main idea of the

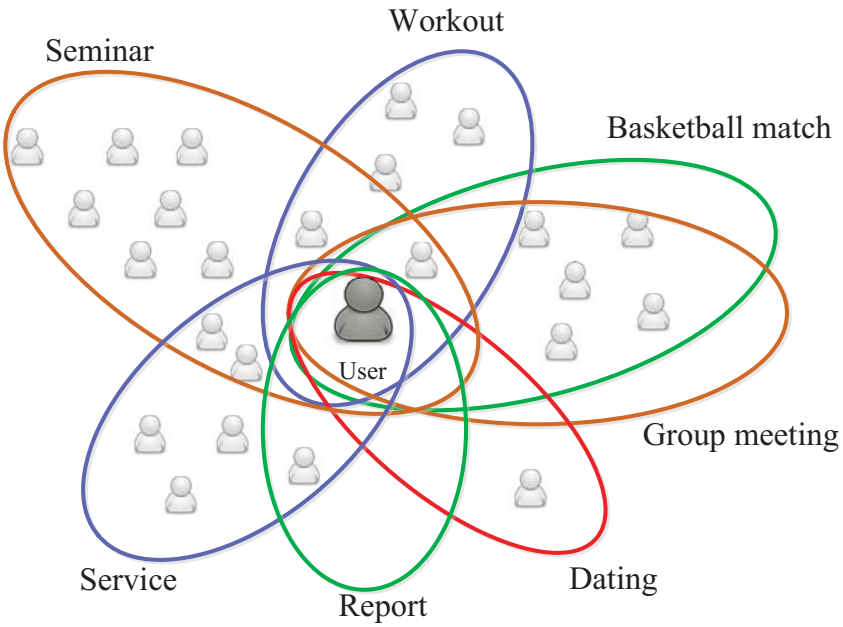


Figure 5.1: One people engaged in different social circles

system is that we first extract social circle information from the social proximity information which is captured by Bluetooth scan, and then we incorporate both social circle and time information to infer social activities. The inference of social activities is based on the social circle model which is constructed using machine learning techniques.

The design of CircleSense entails two major challenges. The first challenge is how to accurately extract social circle from proximity information. Social circle extraction is non-trivial due to two inherent properties of social circle: (1) members in certain social circle are dynamic; (2) different social circles overlap in members. The first property makes it difficult to extract members in certain social circle. The second property influences the accuracy of social circle classification, since overlapping in members increases the misclassification rate. To improve the accuracy of social circle extraction, we apply metric learning technique to develop a social circle classification model.

The second challenge is how to achieve low energy consumption. The system needs to use Bluetooth module embedded in smartphone to scan the Bluetooth-enabled devices to obtain the proximity information. However, frequently using Bluetooth to conduct de-

vice scanning consumes a significant amount of energy, which affects the applicability of the system. Therefore, we propose an energy-efficient algorithm which adaptively performs Bluetooth scanning based on the current social activity context.

In summary, our work has the following contributions:

1) We introduce the concept social circle to identify a distinctive behaviour pattern of social activities. Compare with other patterns extracted from individual user, social circle is able to characterize social activities more accurately.

2) Based on social circle, we develop a practical smartphone-based system called CircleSense for recognition of a generic categories of social activities.

3) We evaluate the system with a 16-participant dataset which is collected from the deployed android phones. The experimental results demonstrate that CircleSense outperforms the existing methods.

5.2 Motivating Scenarios

Social activity recognition opens up many opportunities for novel applications. In the following, we will discuss three motivating applications: automatic social activity log, real-time shared calendar and context-aware phone interruption.

Automatic Social Activity Log: Social activity recognition is able to automatically record users' social activities. Therefore, it allows users to have a reflection on how much time they have spent on different social activities, and thus being able to better control and manage their life towards a balanced lifestyle. For instance, the system can remind the user to spend more time with friends if the user spends most of time alone. Besides, automatic activity log can help the elderly to overcome cognitive decline. Furthermore, the social activity log provides the support for data-driven social science research. Compared with commercial software [Stu12], our work can automat-

ically and accurately record the activities without any human intervention.

Real Time Shared Calendar: As pointed out in [LOIP10], social activities in the shared calendar does not always occur due to different factors. Social activity recognition is able to record the social activity of a user in a real time manner. By combining the output of social activity recognition system and the original shared calendar, a real time shared calendar can be provided, which can potentially increase the effectiveness of collaboration among colleagues.

Socially-aware Applications: Social activity recognition can support many novel socially-aware applications such as context-aware phone interruption management [RDV11]. For example, a phone can automatically turn to the silent mode or choose the corresponding volume when identifying certain social activity is going on, so to avoid the embarrassing phone interruption.

5.3 Preliminaries

In this section, we first introduce several important concepts including: social activity, social circle and social proximity. Next, we formulate the problem of social activity recognition.

5.3.1 Social Activity

We give our formal definition of social activity and discuss some of the inherent properties as below.

Definition 1. *A social activity is an event triggered by a set of users gathering on purpose and interacting in a place.*

Based on our definition, some of the activities are excluded from social activities like

Table 5.1: Summary of terms and their definitions

Term	Definition
a_i	a social activity
A	a set of social activities
c_i	a social circle i
d_i	the time span of the activity i
D_s	scan duration
e_i	the mobile device of user i
f	the proximity information of a certain user
F	a set of proximity information
I_h	high scan interval
I_l	low scan interval
L_a	a set of historical duration of activity a
r_i	the reference fingerprint of social circle c_i
t	current time stamp
T	a set of time stamp
u_i	a certain user i
U	a set of users
U_i	a set of users who are involved in the activity i
V	variance threshold of activity duration

opportunistic chatting, since the focus of our work is on the long term repetitive social activities.

Specifically, we formulate a social activity a_i for an individual user as $a_i = (U_i, d_i)$, in which $U_i = \{u_1^i, ..., u_j^i\}$ is a set of users who is involved in the activity a_i, and $d_i =< t_s^i, t_e^i >$ denotes the time span of the activity starting at time t_s^i and ending at time t_e^i. In this work, we targets at a generic set of social activities $A = \{a_1, a_2, ...a_n\}$, including meeting, seminar, classes, sports, religion service, dating, family time, etc. The details of the experimental setting are presented in section 5.

5.3.2 Social Circle

As we discuss previously, an individual user performs certain social activity with a certain set of people, and different social activities usually involve distinctive sets of people. Here, we introduce the concept of social circle to characterise social activities. We give the formal definition of social circle as below:

Definition 2. *A social circle refers to a set of people who frequently participate in certain social activity together.*

For example, the colleagues in a research lab considered as a social circle, since they frequently engage in research meetings together. Particularly, we identify two inherent properties of social circle.

1) *Dynamic*. Members in certain social circle are dynamic over time. Take research lab colleagues as an example, new students will be employed and become the new members of this social circle. Likewise, some people might leave the lab and no longer belongs to the research lab circle.

2) *Overlapping*. Different social circles overlap in members. For example, some students might participate in the same research meeting, while playing basketball together. Thus, they are both the members of two social circles.

5.3.3 Social Proximity

In order to model social activity, we need to extract the social circle information. In this work, we take advantage of user's proximity information to extract social circles. Specifically, the proximity information is obtained by scanning the nearby Bluetooth modules using smartphones. The advantages of using Bluetooth discovery to obtain social proximity are three-fold. First, the discovery range of Bluetooth module usually is within a radius of 30 feet, which implies that those discovered Bluetooth module are in

the proximity of each other. Second, since Bluetooth module has a unique address and could be used to represent different user. Third, Bluetooth module is energy-efficient and available on off-the-shelf smartphone.

For a user s, the proximity information is the list of the MAC address of Bluetooth-module in smartphone in proximity of his/her smartphone. The proximity information is represented in the format of $f = [e_1, e_2, ..., e_m]$, where $e_k = 1$ indicates the device of u_k has been detected, while $e_k = 0$ means the device of u_k did not appear during the data collection.

5.3.4 Problem Formulation

Given a training dataset $D =< U, A, F, T >$ collected from multiple users, in which a data record indicates a user $u_i \in U$ is involved in a social activity $a_i \in A$ at time $t \in T$ and T means a set of time, while his/her proximity information is $f \in F$ and F is a set of proximity information. We assume users bring their phones, indicating that when user u_i's phone is detected, he/she is in the proximity. Our objective is to construct an accurate social activity recognition model through using the provided dataset, such that given a user's proximity data at certain time, we are able to recognize the social activities that he/she is engaged in. This problem is a classification problem and we will leverage supervised learning approach to tackle it.

5.4 System Design

In this section, we first describe the system overview of CircleSense. Second, we present the energy-efficient device discovery algorithm. Next, we describe how to construct social circle recognition model to identify social circle from proximity information. Finally, we discuss the details how to incorporate social circle and time information to

infer social activity.

5.4.1 System Overview

In order to underpin applications in real world, a social activity recognition system which is based on smartphone has to achieve high accuracy and low energy consumption, due to the limited battery life of smartphone. Thus, the objective of this work is to design an accurate and efficient smartphone-based system for social activity recognition. To meet the above-mentioned system requirements, we design two key system components: integration of social and time, energy-efficient data collection. In the following, we will convey the intuition of our ideas.

Integration of social circle and time. In order to accurately recognize social activities, the key lies in figuring out the informative feature that is able to characterise social activities. We observe that a social activity is closely associated with a certain set of people. Furthermore, due to the routine nature of human behaviour, social activity is also correlated with temporal information. Thus, based on the above observations, the social activity recognition model in this system incorporates social circle and time for social activity recognition under Bayesian framework.

Energy-efficient data collection. Due to the limited energy capability of smartphone, the system has to conserve energy. Collecting data especially conducting device discovery accounts for the majority of energy consumption, therefore, we focus on optimizing the strategy for Bluetooth device discovery. There is a key observation that the user engages in the same social activity for a certain amount of time and his/her social activity context remains the same during that period. Thus, it is a waste of energy to keep collecting data in a fixed scan interval during the social activities, since the user's state does not change at all. Our idea is that we first estimate the end time of current social activity, and then reduce the Bluetooth scan frequency before the end of the activity due to the fact that user will involve in the same activity at that period.

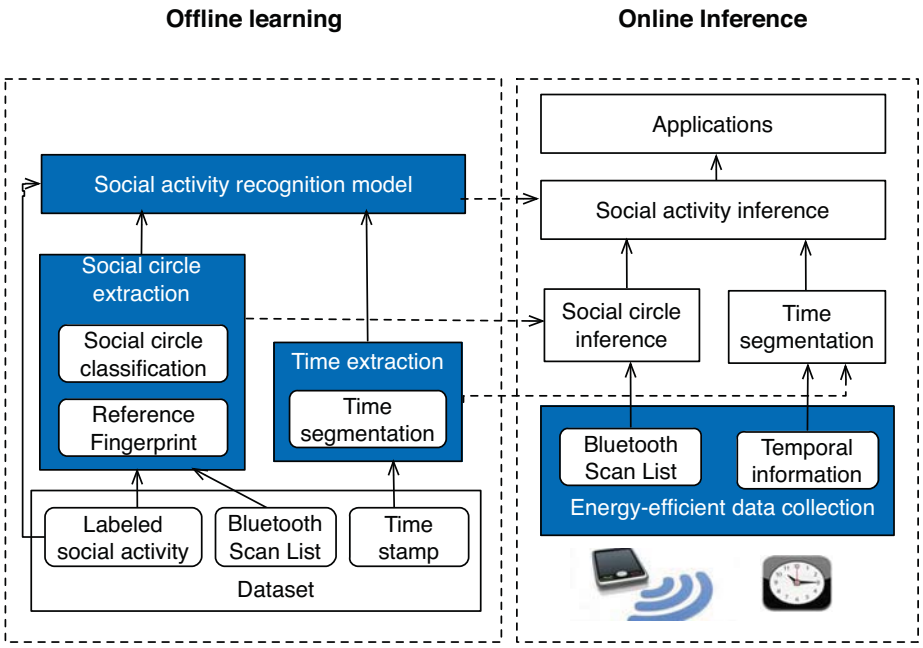


Figure 5.2: Social activity recognition architecture

As illustrated in Fig. 5.2, the system is composed of two modules: offline learning and online inference. In the offline learning module, social activity recognition model is constructed based on the collected dataset, while online inference module is responsible for deriving social activity based on the data collected in realtime from smartphone.

Offline learning module contains four major components: dataset, social circle extraction, time extraction and social activity recognition. The dataset includes bluetooth scan list, time and the corresponding social activity label. Social circle extraction is to classify different social circles based on the proximity information obtained from Bluetooth scan list. Time extraction focus on segmenting and transforming the time stamp into discrete time slots. Finally, social activity recognition model is developed with the output from social circle extraction and time extraction as well as the corresponding social activity label using supervised learning technique.

Online inference mainly includes four components: energy-efficient data collection, social circle inference, time segmentation and social activity inference. Energy-efficient data collection is collecting both Bluetooth scan list and the associated time stamp us-

ing Bluetooth module embedded in smartphone. After collecting the data, social circle inference will derive the corresponding social circle based on the proximity information obtained from the Bluetooth scan list, while time segmentation will transform the time stamp into a time slot. Finally, based on the social circle and time slot information, the social activity inference component will output the label of the associated social activity.

5.4.2 Energy-efficient Device Discovery

To capture social circle, the first step is to conduct device discovery. During the device discovery, the Bluetooth module sends out inquiry messages periodically and waits the response from the other scanning devices. Once the scanning device receive the inquiry message, it will sends back the response to the inquiring device.

The Bluetooth device discovery usually consists of an inquiry scan ($\tilde{1}$60 mW), page scan ($\tilde{2}$10 mW) and no-scan ($\tilde{1}$10mW). Although Bluetooth Low Energy (BLE) has been used a low power mechanism to for device discovery, the experiment [SLJ+14] shows that only 15 - 20 mW is saved with a Samsung Galaxy S4 running on Android 4.3, as BLE shares the same radio and antenna as regular Bluetooth. Therefore, frequently performing Bluetooth scan will consume a significant amount of energy. The simple strategy for device discovery is a periodic scheme with constant scan duration and interval. However, scan interval needs to be short enough to ensure the practicality of the system. Otherwise, long scan interval will give rise to low recall rate, since some of activities might occur within the scan interval. However, a periodic scheme with short scan interval causes a significant amount of energy consumption for a smartphone.

Thus, to reduce the energy consumption of smartphone and ensure system accuracy, we present an energy-efficient activity duration-aware device discovery algorithm which adaptively changes the Bluetooth inquiry interval. Our solution is based on a key observation that the user engages in the same social activity for a certain amount of time and his/her social activity context remains the same during that period. It indicates that

there is no need to perform device discovery during the activities. Our basic idea is that we first estimate the end time of current social activity, and then increase the Bluetooth scan interval before the end of the activity. In other cases, the device discovery is conducted with short scan interval. Therefore, this algorithm can strike a balance between high accuracy and energy efficiency.

The next question is how to accurately estimate the end time of a social activity. Our method leverages the historical data to estimate the duration of social activities. To guarantee the accuracy of estimating the end time of social activity, we only consider only those activities which is highly predictable in terms of duration. Specifically, only those activities with low duration variance on previous dataset are considered.

Algorithm 4: Energy-efficient activity duration-aware device discovery algorithm

```
if (a_t = a_{t-1}) or (var(L_{a_t}) > V) then
    scan nearby device with scan duration D_s and scan interval I_l ;
else
    from t to t + min(L_{a_t}):
        scan nearby device with scan duration D_s
        and scan interval I_h;
end
```

The energy-efficient device discovery algorithm is presented in Algorithm 4. There are two key parameters in Algorithm 4: scan interval and variance threshold V. Smaller variance threshold and shorter scan interval will lead to aggressive device discovery, and thus consuming more energy. We initialise scan duration D_s to be 10 seconds. Low scan interval is set to 10 minutes and high scan interval is set to 20 minutes. In Algorithm 1, a_t is denoted as the label of current social activity detected by the system at t, while L_{a_t} refers to a set of historical duration data of a_t activity.

After each round of social activity recognition, we adapt the values of scan interval based on the variance of historical duration data of detected activity a_t and the comparison with previous detection result. If the variance of the detected activity duration

L_a is larger than V or the detection result a_t is the same as the previous detection result a_{t-1}, we keep conducting device discovery with scan duration D_s and low scan interval D_l. If current detection result a_t is different from the previous result a_{t-1}, it indicates that time t is very close to the start time of activity a_t. And the condition that L_a is smaller than V is also satisfied, then we conduct device discovery with scan duration D_s and long scan interval D_h from time t to the time $t + \min(L_{a_t})$. Here, we use $\min(L_{a_t})$ as a conservative estimation of the duration of activity a_t.

5.4.3 Social Circle Extraction

This section presents the details of social circle extraction. The function of social circle extraction is to output the label of social circle given the social proximity information. We first extract reference fingerprints of different social circles from proximity information, and then adopt metric learning technique to construct accurate social circle classification.

Reference Fingerprint

In essence, reference fingerprint of a social circle is a set of users who can characterize this social circle. Reference fingerprints of different social circles will be used in the later social circle classification. Reference fingerprints are extracted based on the proximity information obtained from Bluetooth device discovery during different social activities.

To obtain reference fingerprint of a social circle, we need to extract the members of the social circle. In order to do that, we first define the degree of membership of a user with regard to a specific social circle. And then users with high membership to a social circle will be grouped together as the reference fingerprint.

In particular, the membership degree of user u_j to social circle c_i, denoted as m_{ji} is measured by frequency of attendance in activity a_i, which is computed as follows.

$$m_{ji} = \frac{c(u_j, a_i)}{c(a_i)}, \forall u_j \in U, \forall a_i \in A \tag{5.1}$$

where $c(u_j, a_i)$ is the number of records of activity a_i which involves user u_j , while $c(a_i)$ is the number of record of activity a_i.

Then, a user will be added to a social circle only if his/her membership degree to the associated activity is above a certain threshold. For a social circle c_i, its reference fingerprint r_i can be obtained by:

$$r_i = \{u_j | m_{ji} > \beta, \forall u_j \in U\} \tag{5.2}$$

The setting of threshold β will be discussed in the evaluation part. If the value β is set too high, a lot of members belonging to a social circle will be mistakenly filtered out. If it is set too low, then the noise will be included.

Metric Learning-based Social Circle Classification Problem Formulation

As we discuss previously, social circle exhibits dynamic and overlapping properties, which makes different social circles difficult to be distinguished based on social proximity information. Regarding these properties of social circle, we apply the metric learning method to construct social circle classification model in order to improve the accuracy.

The key idea of metric learning [NB11] is to learn a new distance measurement to make the data records of the same labels more similar while making the data records of different labels more distinguished. The rationale behind metric learning in our context is that discriminative features can be obtained based on the new distance measurement, so as to circumvent the dynamic and overlapping properties of social circles.

In order to classify social circle from proximity information, we first choose cosine similarity [NB11] as distance measurement between proximity data and the reference fin-

gerprint of a particular social circle. The cosine similarity is chosen to measure distance, because it is always within the range [-1, 1], leading to a simple and effective objective function for problem formulation. Here, both proximity and reference fingerprint of social circle are represented as a vector. Then the distance between proximity data sample x and reference fingerprint of social circle y based on cosine similarity with transformation matrix W is denoted as:

$$D(x, y, W) = 1 - \frac{(Wx)^T(Wy)}{||Wx|| \cdot ||Wy||} \tag{5.3}$$

The problem of metric learning in our context can be formulated as below: given a training dataset $D_f =< U, A, F >$. The proximity data records of social activity a_i are denoted as $F^i = \{f_{i1}, f_{i2}, ..., f_{iN}\}$ and the corresponding social circle is represented as c_i. $C = [c_1, c_2, ..., c_n]$ is the total set of social circles. W is the diagonal transformation matrix which we aim at learning. The objective is to maximize the classification accuracy by maximizing the margin [GBNT04]. Margin of a data record f_{ij} means the distance between f_{ij} and the nearest social circle reference fingerprint $r_j(j \neq i)$ minus the distance between f_{ij} and the corresponding social circle reference fingerprint r_i. The objective function is formulated as:

$$\begin{aligned} &\max \sum_{\forall f_{ij} \in F^i, F^i \in D_f} (D(r_{ne}, f_{ij}, W) - D(r_i, f_{ij}, W)) \\ &s.t. \forall_{i \neq j}, W_{ij} = 0 \end{aligned} \tag{5.4}$$

where $D(r_{ne}, f_{ij}, W) = \min_{k \neq i} D(r_k, f_{ij}, W)$ means the Euclidean distance of f_{ij} to the nearest social circle reference fingerprint of different social activity and $D(r_i, f_{ij}, W)$ denotes the distance to corresponding social circle.

Social Circle Classification Algorithm Design

In this section, we need to compute the transformation matrix W to optimize the objective function of the problem given the training dataset. Since cosine similarity is used

as the distance measurement, we can use a fast and simple gradient-based algorithm to obtain the optimal value of W.

Recall that the objective function is represented as:

$$f(W) = \sum_{\forall f_{ij} \in F^i, F^i \in D_f} (D(r_{ne}, f_{ij}, W) - D(r_i, f_{ij}, W)) \tag{5.5}$$

And the gradient of objective function can be computed as

$$\frac{\partial f(W)}{W} = \sum_{\forall f_{ij} \in F^i, F^i \in D_f} (\frac{\partial D(r_{ne}, f_{ij}, W)}{\partial W} - \frac{\partial D(r_i, f_{ij}, W)}{\partial W}) \tag{5.6}$$

In order to compute the gradient of the objective function, without loss of generality, we redefine

$$D(x, y, W) = \frac{u(W)}{v(W)} \tag{5.7}$$

Then we have

$$\frac{\partial D(x, y, W)}{\partial W} = \frac{1}{v(W)} \frac{\partial u(W)}{\partial W} - \frac{u(W)}{v(W)^2} \frac{\partial v(W)}{\partial W} \tag{5.8}$$

We further obtain $\frac{\partial u(W)}{W}$ and $\frac{\partial v(W)}{W}$ based on the following equations:

$$\frac{\partial u(W)}{\partial W} = W(xy^T + yx^T) \tag{5.9}$$

$$\frac{\partial v(W)}{\partial W} = \frac{\sqrt{y^T W^T W y}}{\sqrt{x^T W^T W x}} W x x^T + \frac{\sqrt{x^T W^T W x}}{\sqrt{y^T W^T W y}} W y y^T \tag{5.10}$$

As shown in Algorithm 5, we use gradient ascent method to obtain the optimal value of W such that the objective function is maximized.

Algorithm 5: A gradient-based algorithm for learning parameter W

```
δ denotes a small positive number close to 0;
while h > δ do
    Compute h = ∂f(W)/∂W with dataset D_tr;
    W → W + h;
end
Return W;
```

The complexity of the computing gradient of the objective function is $O(s \cdot m \cdot m)$, where s is the total number of data record in D and and *m* denotes the total number of users. Therefore, the complexity of the solution for the optimization algorithm is $O(r \cdot s \cdot m \cdot m)$, where r is the number of iteration.

Social Circle Classification

The classification of social circle is very similar to nearest neighbours algorithm. First, we measure the distance between the proximity data and the reference fingerprint of each social circle. Then, the social circle label of the proximity data will be assigned as the label of nearest neighbour.

Given the proximity information f and a set of social circle reference fingerprints, then the label of f will assigned as c_i only if f has the minimum distance with r_i among all the reference fingerprints. The decision rule for social circle recognition is described as below:

$$c_i = \begin{cases} \arg\min_i D(c_i, f, W) & \text{if} \quad D(c_i, f, W) < \varepsilon \\ Unknown & \text{oterhwise} \end{cases} \tag{5.11}$$

where ε denotes the distance threshold for decision. When the minimum distance is above the threshold, it would be treated as noise and discarded to reduce the recognition error. We will analyze the impact of setting ε in the evaluation session.

5.4.4 Time Information Extraction

In this section, we will incorporate temporal information into social activity recognition to improve the system accuracy. Although social circle has been extracted in previous section, it might not be adequate to identify some social activities due to the fact that people in a social circle can engage in more than one social activities.

Take religion circle as an example, they can have worship, scripture sharing or other social activities together at different time (seen in Fig. 3). Another motivating example would be on Wednesday from 9 a.m. to 10 a.m., a user engages in a team meeting with colleagues and go to play basketball with colleagues on Friday afternoon. Therefore, we want to include the temporal information to better recognize users's social activities. The underlying rationale of exploiting temporal information is that social activities exhibit a temporal pattern.

Time Segmentation

To associate time with social activities, we first segment the time of a week into slots. The critical thing is to select the appropriate length for each time slot. If the length of the time slot is too large or two small, then the correlation between time and social activities might not be obvious. For instance, if we set the length of a time slot to be two hours, then several activities might happen during the same time slot, making the temporal information less discriminative than it should be.

In particular, we set the length of each time as 30 minutes. The first 30 minutes of Sunday is referred as time slot 1 and the last 30 minutes of Saturday is referred as time slot 336. The time segmentation $seg(t)$ is defined as below.

$$seg(t) = weekday(t) \cdot 48 + hour(t) \cdot 2 + [minute(t)/30] + 1 \tag{5.12}$$

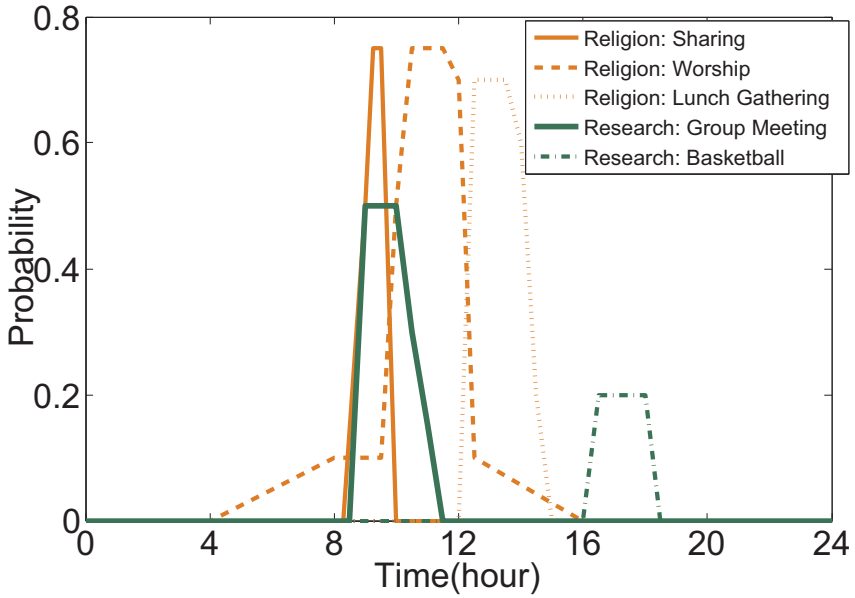


Figure 5.3: Temporal patterns of different social activities

Given time t, the corresponding segmented slot t_k can be computed by the following equation.

$$t_k = seg(t) \tag{5.13}$$

5.4.5 Social Circle and Time-based Activity Recognition

After the extraction of both corresponding social circle and time slot, we adopt the naive bayesian framework to combine both social circle and time slot to derive social activity. First, we need to obtain the conditional probability of social circle or time slot given particular social activity.

The conditional probability of the occurrence of social circle c_k under the social activity a_i is computed as below.

$$P(c_k|a_i) = \frac{c(c_k, a_i)}{c(a_i)} \tag{5.14}$$

where $c(c_k, a_i)$ is denoted as the number of data record of activity a_i that is conducted by people in social circle c_k, and $c(a_i)$ denotes the total number of data record of activity

a_i.

The conditional probability of the time slot t_k under the social activity a_i is computed as below.

$$P(t_k|a_i) = \frac{c(t_k, a_i)}{c(a_i)} \tag{5.15}$$

where $c(t_k, a_i)$ represents the number of data record of activity a_i at time slot t_k.

However, the probability $P(t_k|a_i)$ might be 0 if a social activity occurs in a time slot different from before. As a consequence, it cancels out the information of social circle and thus increasing the false negative rate. To avoid this scenario and better make use of temporal information, we apply the additive smoothing technologies to refine the equation.[Wik12]

$$P(t_k|a_i) = \frac{c(t_k, a_i) + 1}{c(a_i) + \mu} \tag{5.16}$$

In above equation, μ will be set as $1/10$ of $c(a_i)$. Adding μ means we consider all the possible time slots in the calculation. With the refinement, we will hold $P(t_k|a_i) \neq 0$ for all t_k.

Then the conditional probability of social activity given both social circle and time can be computed as below.

$$P(a_i|c_k, t_k) = P(c_k, t_k|a_i)P(a_i) \tag{5.17}$$

Since we adopt naive baysian framework, which assumes social circle and time are conditionally independent with respect to social activity. Then we have:

$$P(c_k, t_k|a_i)P(a_i) = P(c_k|a_i)P(t_k|a_i)P(a_i) \tag{5.18}$$

Finally, based on maximum a posteriori decision rule, the activity will be derived as

$$a = \arg\max_{a_i}\{P(c_k|a_i)P(t_k|a_i)P(a_i)\} \tag{5.19}$$

In summary, the social activity recognition can be conducted based on the algorithm 6.

Algorithm 6: social activity recognition algorithm

Input : the proximity information f and time t

Output: the label of social activity as a

1 Step 1: social circle recognition from f

2

$$c_k = \begin{cases} \arg\min_k D(r_k, f, W) & \text{if} \quad D(r_k, f, W) < \varepsilon \\ Unknown & \text{oterhwise} \end{cases}$$

3 Step 2: time segmentation from t

$$t_k = seg(t)$$

4 Step 3: social activity recognition result

5

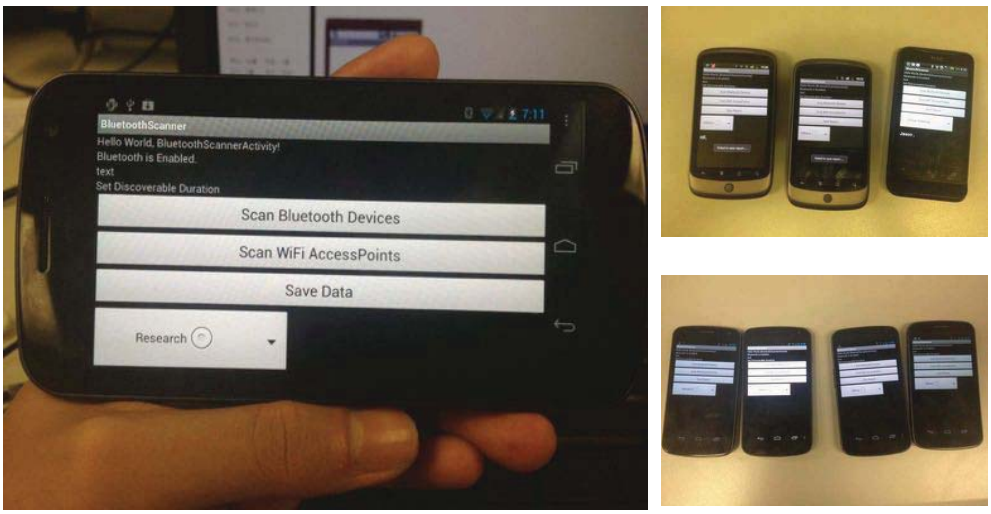
$$a = \arg\max_{a_i}\{P(c_k|a_i)P(t_k|a_i)P(a_i)\}$$

5.5 Evaluation

In this section, we present the evaluation methodologies and results of our proposed methods. We start by discussing the experiment setup including dataset, benchmarks and evaluation methods. Then, we present a detailed evaluation result of the system.

5.5.1 Implementation

To collect users' social activities, we develop a prototype application (see Fig. 5.4) that can run on Android-based smartphones. Both Android 2.3 and 4.0 platforms are



(a) Screenshot of Prototype Software (b) Some Deployed Android Phones

Figure 5.4: Experiment deployment

supported.The application is composed of two main components: the first component is responsible for obtaining the list of nearby Bluetooth-enabled devices and WiFi access points data. The reason why we collect WiFi access points data is to conduct the evaluation of the baseline location-based method. The second module provides a user interface for users to manually label the social activity.

We initialise Bluetooth device scan duration D_s to be 10 seconds. Low scan interval is set to 10 minutes and high scan interval is set to 20 minutes.

5.5.2 Trace Collection

To the best of our knowledge, there are no standard datasets available for evaluating social activity recognition. Henceforth, we collect the annotated users' social activities trace by ourselves.

To have a reasonable and representative dataset, we select four categories of social activity (as shown in Table 5.2), including: work, play, develop and connect. Those activities are selected based on the survey from local students and cover most of the social activities that one campus student might conduct. We recruit subjects from a

local university to collect the dataset. Each day, the subject will carrie the cell phone that is deployed with the prototype software and participate different social activities. After that, data trace will be collected and each subject need to label the data with the related social activity.

Category	Specific activities
Work	Meeting: research meeting. Seminar: Networking seminar.
Play	Sports: frisbee, basketball.
Develop	Classes: mobile computing. Religion service: worship, prayer
Connect	family time Dating

Table 5.2: Target social activities

In total, we have collected two datasets. The first dataset is contributed by seven people from May to June, 2012. Three of them are undergraduates, three are graduate students and one is a company employee. The data record format is illustrated in Table 5.3. The data is recorded in a format of $<$time, location, social activity, proximity $>$. In some cases, there is not WiFi access points or the nearby users does not open their Bluetooth-modules, the experimental subjects will record the nearby users' names and the location manually.

The second dataset is collected from four research students in Computing Department of PolyU in a 10-minute period for two weeks. Each of these students are distributed one deployed Android-based smartphone and the rest five lab colleagues are equipped with smartphones that open the embedded Bluetooth Modules periodically. The data format is the same as the one in the first dataset.

Scan Time	Location	Proximity	Social Activity
09:00 22/03/2012	AP_1, AP_2	u_1, u_3, u_4, u_6	Meeting
11:00 22/03/2012	PQ703	u_1, u_2, u_3u_5	Seminar
19:00 24/03/2012	AP_3	u_4	Dating

Table 5.3: Collected data format

5.5.3 Methodology

We examine the proposed methods based on the collected dataset. In order to evaluate the recognition performance, we use Leave-one-out validation technique. We first spilt the collected dataset into two subsets in a time order. The first subset serves as the training set and the second subset is used as the test set. To evaluate the impact of training dataset towards the system accuracy, we extract training datasets with different data amount. Specifically, the first 30%, 50% and 70% of the overall dataset will be used as three training datasets respectively.

In order to measure the performance, we use three widely used evaluation metrics: *precision*, *recall* and *F-measure*. Let us consider a data sample of activity A_1 in the test dataset. If the predicted result is A_1, it will be counted as a true positive (TP). Otherwise, assume the predicted result is A_2, then it would be counted as a false positive (FP). In addition, activity A_1 will be counted as a false negative (FN) since it is missing in the prediction. F-measure reflects the overall effect of both precision and recall. The metrics can be computed based on the following equations.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.20}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.21}$$

$$F = 2\frac{precision \times recall}{precision + recall} \tag{5.22}$$

5.5.4 Benchmarks

In order to demonstrate the effectiveness of the proposed methods, we compare the performance of proposed methods against a number of benchmarks. We have two kinds of benchmarks with respect to social circle classification and social activity recognition respectively. The benchmarks of social circle classification include the popular classification approaches widely adopted in activity recognition such as Naive Bayes, kNN and Support Vector Machine. Those classification approaches are implemented in the open source machine learning software Weka 3.6 [**?**]. The benchmarks of social activity recognition contain the existing time-based, location-based, location and time-based models.

5.5.5 Results

Social Circle Classifier Performance

The objective of the first experiment is to evaluate the social circle classification performance of the proposed metric learning method with respect to different training datasets. As illustrated in Figure 5.5 and 5.6, given 30% of overall data for training, the proposed metric learning is able to classify different social circles with 90% in precision and 97% in recall. In general, when the amount of training dataset increases, the social circle classification accuracy improves. When using 70% of overall dataset to train the model, the proposed metric learning can achieve 100% accuracy in both precision and recall. The main reason is that training dataset covers the test dataset, and thus the model that is learned based on the training dataset shows a good generalisability on the test dataset.

Next, we compare the proposed metric learning method with the typical classification techniques given different amount of training dataset. Here, the typical classification techniques includes kNN, naive bayes and support vector machine. On average, the

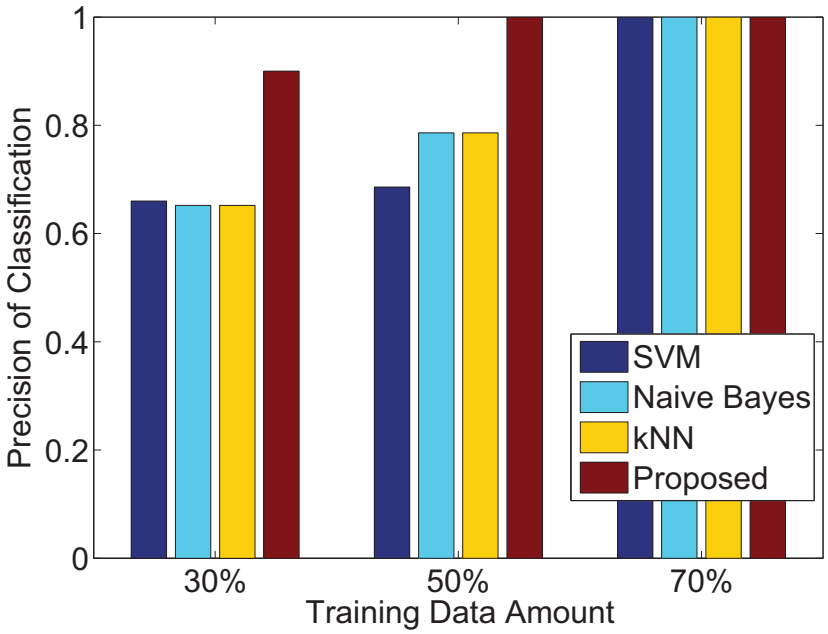


Figure 5.5: Social circle classification precision comparison given different amount of training dataset

proposed method achieve 17% higher precision rate and 5% higher recall rate over the baseline classification techniques. When the amount of training dataset increases, the gap between the metric learning method and other baseline methods in terms of performance reduces. Given a comparatively small amount (30%) of training data, the metric learning approach is able to achieve a quite high precision rate (90%), while the baseline methods only achieve 62% precision rate on average. In particular, it shows that SVM performs slightly better than other two benchmarks given 30% of overall data for training. The above experimental results demonstrate that the proposed method that leverages metric learning technique is able to classify the overlapped and dynamic social circles accurately.

Parameter Analysis

This session aims at analyzing the effects of two main parameters β and ε to the system performance. The first experiment tries to figure out the effect of parameter β on the

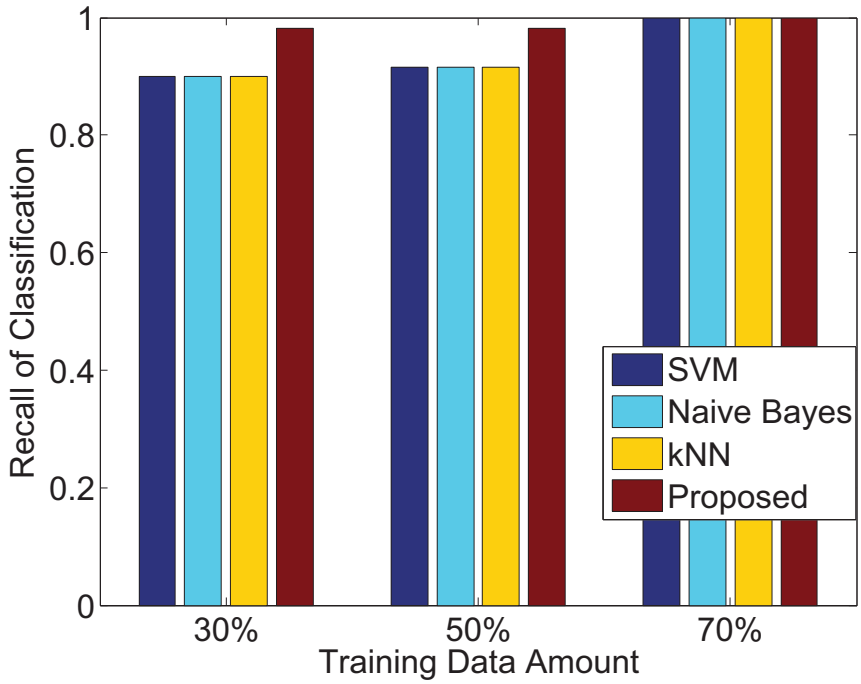


Figure 5.6: Social circle classification recall comparison given different amount of training dataset

accuracy of the social circle classification. Parameter β is the threshold of membership degree to determine whether a user should be considered as a member of a social circle. As seen in Figure 5.7, when the 30% of overall dataset is used as training data, the optimal value of β is 0.5. If the value of β is set above 0.5, the accuracy of social circle classification will decrease. The explanation for this scenario is that when the value of β is set too high, some members of the social circle will be excluded. As a consequence, the extracted members in each social circle are inadequate to be the reference of the actual social circles, and thus degrading the classification accuracy.

Interestingly, when we use 50% of overall dataset as training data, we find that the value of β does not affect the classification accuracy. The reason for that is when the amount of training data is considerable, the members in each social circle with high discriminative ability are included regardless of the value of β. Although some noisy information might be included when the value of β is set too low, the metric learning approach is able to filter out the noise by adjusting the transformation matrix.

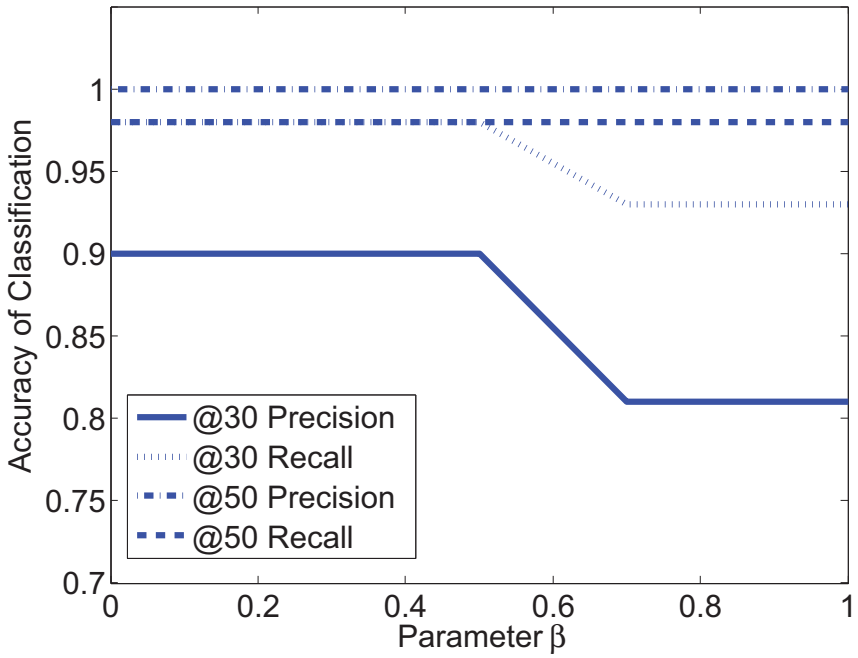


Figure 5.7: The Impact of Parameter β in social circle classification

The second experiment focuses on analyzing the effect of parameter ε. Parameter ε is the threshold of distance measurement for decision making in social circle classification. Figure 5.8 shows that given 30% amount of overall dataset as training data, the optimal value of ε is 0.63. The optimal value ε should be set to 0.65 when 50% amount of overall dataset is used for training. The Figure 5.8 reveals that when the value of ε increases, the precision rate and recall rate of social circle classification will decrease and increase respectively. It is mainly due to two reasons. First, when the value of ε is large, many scenarios other than social activities will be mistakenly recognized as some social activities. Consequently, it leads to a high false positive rate and then degrades the precision rate. Second, when the parameter ε is set with a small value, some social activities would be treated as unknown events and falsely discarded, resulting in a high false negative rate and then a low recall rate.

It is worth emphasizing that there is a tradeoff between false positive and false negative rate. Tuning the value of parameter ε affects false positive rate and false negative rate in social circle classification. Obviously, the the optimal value of parameter ε varies by different datasets.

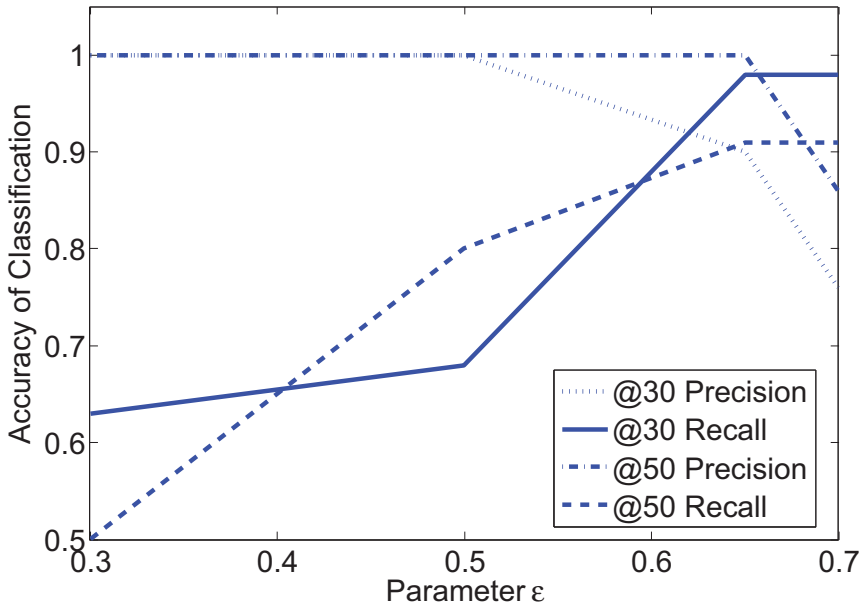


Figure 5.8: The Impact of Parameter ϵ in social circle classification

Social Activity Recognition Performance

The objective of this experiment is to evaluate the social activity recognition performance of the proposed social circle-based method. In this experiment, The first 50% of the overall dataset will be used as training dataset, and the rest of the 50% dataset will be used for test. Table 5.4 shows that the proposed social circle-based model is able to classify the target social activities with 100% in accuracy, 99% in recall and 99% in F1 score. Compared with other models, social circle-based model is able to improve the performance against the best baseline method by 15% in precision, 20% in recall and 18% in F-measure. Among the three baseline models, $location + time$ model achieves the best performance with 85% in precision and 79% in recall. Surprisingly, the model that merely based on temporal information is able to obtain 59% accuracy, which implies that users' social activities exhibit temporal patterns due to routine lifestyle.

Figure 5.9 and 5.10 show the detailed performance of different models in recognizing different social activities. Figure 5.9 shows that social circle-based model outperforms

Methodologies	Precision	Recall	F-measure
Time	0.59	0.57	0.57
Location	0.64	0.58	0.60
Location + Time	0.85	0.79	0.81
Social Circle + Time	1	0.99	0.99

Table 5.4: Social activity recognition performance summary

other baseline models in identifying different social activities. In particular, the second best model which is based on location and time performs poorly in recognizing dating activities. The result is reasonable, since the dating activities are quite dynamic in terms of location and can occur in new places that training dataset does not cover. Location and time model achieves a comparatively low accuracy in identifying sports activities, because different kinds of sport activities can be conducted in the same sports centre so that location is not a good feature to distinguish different sport activities.

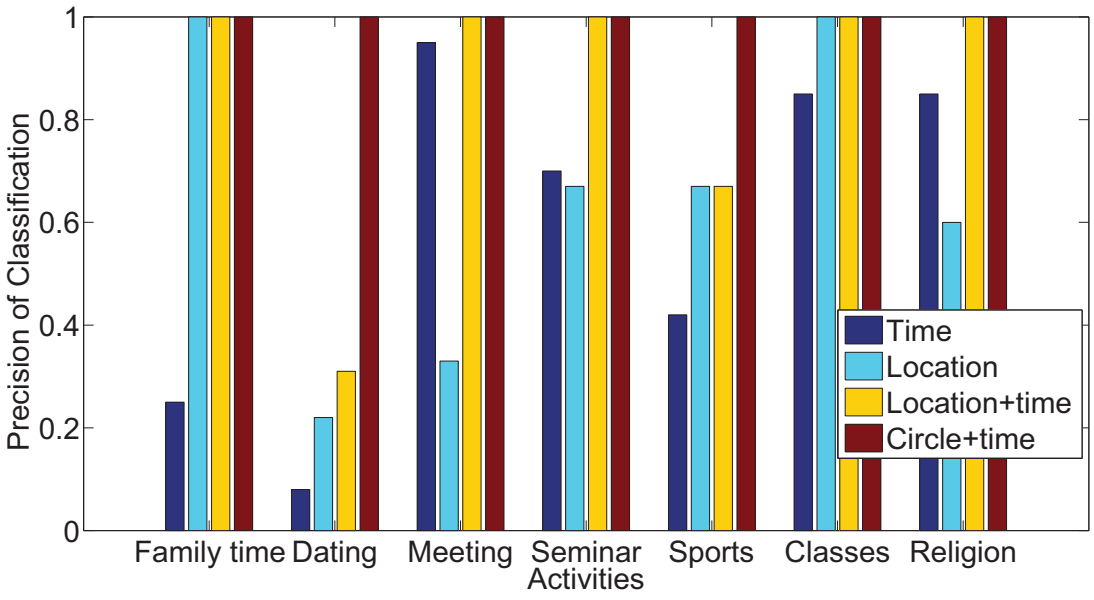


Figure 5.9: Precision rate of different social activities

Figure 5.10 shows the recall rate of different models. Location and time based model only achieves around 72% recall rate in recognizing the class activity. The poor performance is attributed to the occurence of make-up classes. Make-up classes are usually

rescheduled in different places and different time, so that the model built upon the historical location and time does not work.

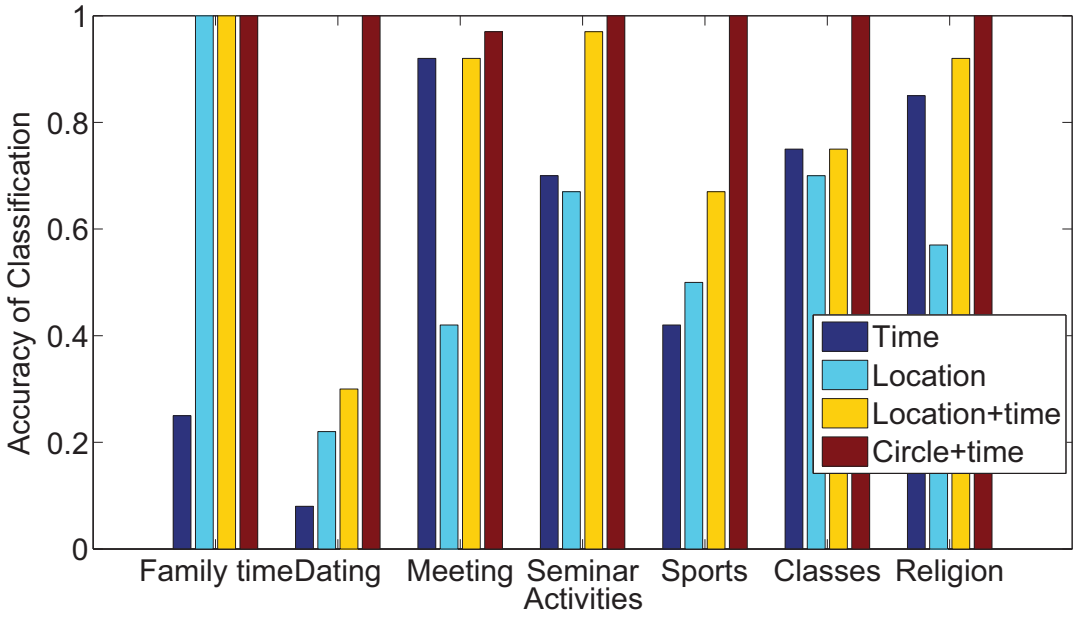


Figure 5.10: Recall rate of different social activities

In summary, the above results reveals some insights. First, the proposed social circle-based model outperforms the baseline methods, particularly in identifying the activities with dynamic spatial and temporal patterns. Second, a considerable amount of social activities such as meeting and seminar show a repetitive pattern in location and time. Third, most of the social activities exhibit a strong social circle pattern, which enables the proposed social circle-based model to achieve high accuracy.

Robustness of the Algorithm

We evaluate the robustness of our social activity recognition algorithm by introducing the noise into the dataset. Specifically, we add noise into the discovered device list. To generate noise, we insert a random number ranging from 1 to 9 of new devices into the scanned device list of both training and testing dataset. When the noise is added into the training dataset, the noise will be eliminated in the reference fingerprint extraction

due to the low membership degree to the associated activity. Therefore, it does not affect the online inference.

Recall that the proximity information is represented in the format of $f = [e_1, e_2, ..., e_m]$, where $e_k = 1$ indicates device e_k has been detected, while $e_k = 0$ means user e_k did not appear during the data collection. Note that the device set $e_1, e_2, ..., e_m$is based on training dataset and fixed. Hence, introducing noisy devices into the testing dataset will not affect the the representation of the original proximity information, since the noisy devices do not belong to the device set. Thus, the proposed algorithm is robust and able to tolerate random noise in the dataset.

Impact of the Penetration Rate

The objective of this experiment is to evaluate the impact of penetration rate of the active Bluetooth devices towards the system accuracy. The penetration rate is measured by the percentage of active Bluetooth devices among all the Bluetooth devices in our dataset. Figure.5.11 depicts how the accuracy of social activity recognition varies with respect to different penetration rates. Generally, higher penetration rate will lead to higher system accuracy. In particular, the accuracy of social activity recognition will still be 100% even when the penetration rate drops to 75%. When the penetration rate is 50%, the system accuracy will be 78%. In this case, still seven out of nine categories of social activity can be accurately recognized.

Given a comparatively low penetration rate, the system can still achieve high accuracy. This scenario occurs due to two reasons. First, different social circles overlap in members. Hence, even those overlapping members are missing, the system accuracy will not be influenced due to their low contributions to differentiating different social activities. Second, there are some social circles that do not share any members with rest of other social circles. For those social circles, there exist redundant members to distinguish with other social circles. Thus, the system is resilient to the fluctuated penetration

rate of active Bluetooth devices.

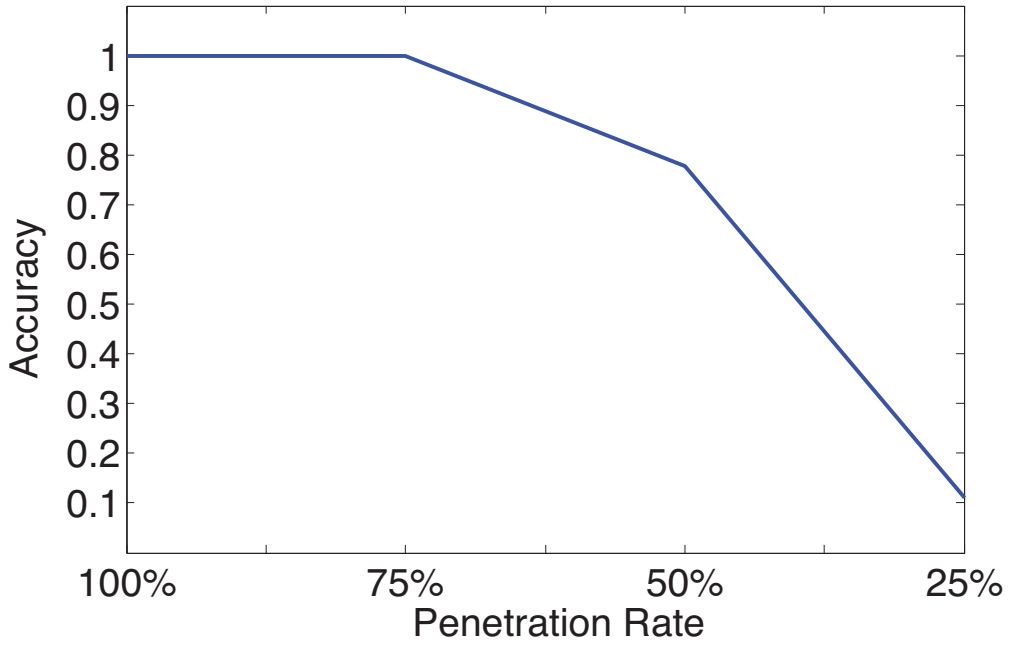


Figure 5.11: System accuracy with respect to different active bluetooth device penetration rate

System Overhead

In this section, we conduct experiment to evaluate the effectiveness of the energy-efficient algorithm for data collection. The proposed algorithm is able to reduce the scanning frequency only when the duration of the on-going social activities is predictable. Thus, only those social activities with small variance in duration are considered. In this work, the value of threshold V is 20 (mins). After the statistical analysis, social activities with duration variance under given threshold V include: meeting, seminar, class and religion. Therefore, if the users conduct one of those social activities, then our algorithm will be able to conserve energy consumption caused by Bluetooth device discovery.

Based on the above the analysis, the effectiveness of energy conservation algorithm

varies and depends on users' social lifestyles. Therefore, we compare the proposed energy-saving data collection algorithm with the baseline periodic data collection method with respect to users in different social statues (undergraduate student, graduate student). Here, we believe different social statues implies different social lifestyles. To estimate the energy consumption, we measure how many times that smartphone conducts Bluetooth device discovery, since the energy consumption caused by one device discovery is quite fixed. The Bluetooth device discovery usually consists of an inquiry scan ($\tilde{1}$60 mW), page scan ($\tilde{2}$10 mW) and no-scan ($\tilde{1}$10mW). In particular, we compare the proposed data collection method with the baseline data collection method in terms of device discovery count within a weekday (9am - 9pm). The baseline data collection method will periodically conduct inquiry scan with an interval of 10 minutes.

Table 5.5: Energy consumption comparison between the baseline and the proposed data collection method

User	Device discovery count (periodic)	Device discovery count (proposed)	Energy saving
Undergrad	84	69	18%
Graduate	84	75	10%

Table 5.5 shows the comparison between the proposed data collection method and the periodic data collection method based on the social activity traces of undergraduate and graduate student respectively. It reveals that the proposed data collection method can save up to 18% of the counts to conduct device discovery than the basline one given the social activity trace of undergraduate student. 10% of the counts to conduct device discovery can be saved given the social activity trace of graduate student. The explanation for the scenario is that compared with graduate student, undergraduate students have a more active social life and spend a considerable amount of time in taking classes and seminars. Therefore, our energy-efficient data collection algorithm works better for those users with active and routine social activities.

There is a tradeoff between energy efficiency and system accuracy. To guarantee the

system accuracy, we adopt a mild energy conservative scheme. Note that, the system keeps collecting data with a low frequency even the user is anticipated to involve in the same social activities for a significant amount of time. In fact, a more aggressive energy efficient scheme could be stopping data collection when user is anticipated to involve in the same social activities until the end of the activities. However, the more aggressive the energy efficient scheme is, the higher the error rate will become, since activities might end earlier than expected and different activities occur during that period. So our energy efficient scheme is able to strike a balance between energy efficiency and system accuracy.

5.6 Discussion

The evaluation result demonstrates that the CircleSense is effective to perform social activity recognition. However, some of the issues still remain further investigation. In this section, we present the limitation of our work and future research direction.

5.6.1 Limitations

The major limitation of our work is that the algorithm is only evaluated on a small-scale dataset collected by ten volunteers in campus. Therefore, the generalizability of the system is not thoroughly validated. We are planning to collect a large-scale dataset with the help of the advanced indoor localisation technology. Besides, our work assumes that the members in social circles are fixed within a significant amount of time and does not tackle the issues brought by the dynamic nature of social circle. Social circle by nature is dynamic in the sense that some members might leave and some new members join in. To enable long-term monitoring, it is essential to adaptively update the members of social circles.

5.6.2 Finer Activity Granularity

Although a set of social activities can be accurately recognized by CircleSense, they are still not fine-grained enough with regard to some applications. Dating, for example, could be interpreted as many different finer-grained activities like watching a movie, shopping or just walking in the park. Apparently, those finer-grained activities can be differentiated by the locations (theatres, shopping malls and park). Therefore, we plan to extract the finer-grained social activities by leveraging the opportunistic location information from WiFi access point or GPS when users are using wireless service, which is energy-efficient.

Social circle does not indicate the activity always. For example, Bob and Alice are lovers. Bob is playing a table tennis with Alice now. Would you detect it as "Dating" or "Sports"? The currently proposed scheme may detect as "Dating" but real benefit of activity detection comes from a more detailed description of the activity such as "playing table tennis". Social circle may be one of strong indicators for potential activities, but cannot replace many other sensory information to detect a more detailed activity.

5.6.3 Feasibility

Our system requires a considerable amount of users to use the proposed prototype software, since this work is based on the Bluetooth device discovery to extract social circle information. One of the major obstacles to the adoption of this software is the energy consumption. Thanks to the low energy consumption of the Bluetooth module and the proposed energy-efficient data collection algorithm, the proposed system is energy efficient, yet achieving high accuracy.

To bootstrap the system, a suitable incentive mechanism needs to be designed to enable the collective sensing, which is considered in our future research plan. Besides, the system also needs to tackle the issue when only a small portion of users turn on

bluetooth module of their smartphones.

It is worth emphasising that the main idea of this work is to extract the social circle pattern to identify user's social activities. Bluetooth device discovery is just one way to obtain user's social circle information. With the advance of the sensing technologies, this work can be generalised into the new platform that are based on speaker recognition, indoor localisation, etc.

5.6.4 Privacy

The system needs to collect the social proximity information of users, which by nature is sensitive. In particular, the proposed system requires the user to allow Bluetooth module to be discovered, which might comprise user privacy as the Bluetooth mac address is unique and can be associated with the user. To avoid being scanned by the malicious party, one remedy is to leverage the pair-up mechanism of Bluetooth such that only the pair-up devices are visible to each other, but invisible to other devices. In addition, since all the data is collected locally from the smartphone and the social activity recognition is performed on the smartphone, which can protect the user privacy.

5.7 Summary

This work presents the study of using mobile phone to obtain social proximity information for social activity recognition. We first introduce the concept of social circle, which is defined as a set of users frequently gathering to conduct certain social activities. Based on the social circle and temporal information, we develop a system called CircleSense for recognizing a generic set of social activities. In particular, the system leverages the metric learning technique to train a classification model for social circle recognition. The result demonstrates that CircleSense achieves better accuracy than

other baseline methods in recognizing a variety of social activities.

As an initial study, we demonstrate the effectiveness of exploiting social circle and temporal information to derive different social activities. Although the result is promising, the design and functionalities of the system are premature. In the future, we plan to work on the following directions. First, we will leverage more sensor modalities such as accelerometer and microphone to infer more fine-grained social activities. Second, we will conduct a large-scale dataset to evaluate of the effectiveness of social circle information to distinguish different social activities. Third, we will investigate the mechanism that could adaptively update the members in a social circle, which enables the long term monitoring.

Chapter 6

Social Context-based Human Mobility Prediction based on Large-scale Wi-Fi Traces

In this chapter, we study the problem of human dwell time prediction. The rest of this chapter is organised as follows. Section 6.1 presents the overview of this chapter. In section 6.2, we introduce the Wi-Fi probe request mechanism, Wi-Fi probe collection system and the statistical details of dataset. Section 6.3 provides an overview of our StayPredictor system and the problem formulation. In section 6.4, we present the details of system design, including the pre-processing, feature extraction, correlation analytics and predictive model. We evaluate our solution in section 6.5. Section 6.6 concludes this chapter.

6.1 Overview

Understanding and predicting human mobility underpins a variety of applications. By exploiting the information of user movement, systems such as cloud computing, location-

based recommendations [ZZXM09] and content-based delivery networks [Lei09] can further improve its performance. In particular, knowing the dwell time of users in a place gives marketing and network administration a big advantage [MSCN13]. For example, marketing agencies can conduct promotional campaigns through delivering customised coupons to attract rushed shoppers, or they can also offer entertainment packages for those who stay longer. Network administrators could also prioritize bandwidth for users with short dwell time to maximize the networking service utility by meeting the network access needs of the maximum number of users.

Recently, we have witnessed the explosive growth in Wi-Fi-enabled mobile devices such as smartphones or tablets. To offload high data demands, those devices would connect to Wi-Fi networks whenever possible. To search for available networks, these mobile devices would periodically scan the Wi-Fi band for access points by transmitting probe messages. Note that each probe message contains a unique device identifier (MAC address) that could be associated with an individual. Therefore, through collecting and analysing those Wi-Fi probe traces, human mobility patterns could be further revealed.

The goal of our work is to understand and predict the dwell time of users at a location based on large-scale Wi-Fi probe traces. In particular, we seek to identify the fundamental factors that influence the user dwell time and quantify the influences. We then aim to construct a predictive model for user dwell time by incorporating the influence factors. Note that predicting the dwell time of users based on some contextual information is not novel (e.g., arrival time, past dwell time). Works [MSCN13][VDN11] [DGP12], have all studied such cases. However, none of the previous works have tried to quantify the influence of fundamental factors upon user dwell time. It is worth emphasising that quantifying the influences of factors towards user dwell time is significant. On one hand, it advances the understanding of user movement behaviour. On the other hand, it provides guidelines for the design of dwell time prediction model. Moreover, social context which does impact user mobility behaviour, has not been considered into the

previous dwell time prediction models. To achieve our goal, we need to answer the following research questions.

First, how to identity and quantify fundamental factors of user dwell time? Although human movement has a certain degree of variation, they also exhibit structural patterns due to routine life style and social influence [CML11]. Therefore, we identify three factors that may impact user dwell time: arrival time context, social context and historical context (e.g., previous dwell time). To quantify the influence of those factors, a number of techniques have been adopted. First, we apply Pearson correlation coefficient [ZTS03] and Spearman's rank correlation coefficient [Ken48] to measure the linear and monotonic relationship between social context and dwell time. Besides, mutual information is used to assess the dependency between time context and dwell time. We also adopt standard deviation as a metric to quantify the routine degree of user dwell time and leave time.

Second, how to develop an accurate user dwell time prediction model that effectively incorporates influence factors? Instead of exploiting all the influence factors in a single model, we propose an ensemble model which combines multiple single models which are built on different factors. The rationale of using ensemble model is that it improves the prediction accuracy via reducing the variance. In particular, we first build up three single prediction models: time and social context based model, historical dwell time model and historical leave time model, and then integrate them under a linear framework.

We analyze the Wi-Fi trace dataset collected from 111K devices in a large shopping mall for 10 days. We find that: (i) Repeat visitors exhibit a high degree of repetition in dwell time and leave time. (ii) Dwell time of visitors is positively correlated with the size of social group during the visit. (iii) Surprisingly, dwell time of repeat visitor is negatively correlated with the amount of co-located users.

The primary contribution of this work is as follows:

- We design StayPredictor, a system to predict user dwell time at a shopping mall based on Wi-Fi traces. Our solution is evaluated on 10-day WiFi traces collected from 111K devices, and being able to achieve 32.6% relative error.

- We propose a correlation analytics framework to quantify the impact of factors towards user dwell time. Specifically, this framework can measure the linear and non-linear dependency relationship between dwell time and other factors.

- To our knowledge, this is the first work to identify and quantify the impact of social context towards user dwell time. We believe that the findings could provide new insights for human mobility modelling.

- We present an ensemble method for dwell time prediction. By incorporating multiple models that are built from different context information, the proposed ensemble method is able to achieve higher prediction accuracy over single models.

6.2 Preliminary

6.2.1 Wi-Fi Probe Request Mechanism

To detect known and unknown Wi-Fi networks, client devices usually scan the Wi-Fi band by actively sending probe request message according to the 802.11 standard [Soc12]. After receiving a probe request, the access point that associates with the probe message will reply with a probe response, allowing the client devices such as smartphones to initiate a connection. Sending probe request message is an efficient way to provide a transparent connection for the known Wi-Fi networks. Additionally, it enables the connection to the networks whose access points do not broadcast Beacons that contain the SSID.

6.2.2 Wi-Fi Probe Collection

In order to collect real-world Wi-Fi probe messages, we deploy a Wi-Fi probe collection system in the region of interest. As illustrated in Figure 6.1, the Wi-Fi probe collection system consists of several monitors that are based on off-the-shelf routers for data acquisition, and a central server for data storage. As the users with mobile devices pass along the region of interest, the probe request message will be snipped and collected by the deployed monitors, and then sent to the central server. Technically, whenever the Wi-Fi interface of a device is on, the Wi-Fi probe collection system is able to track the device.

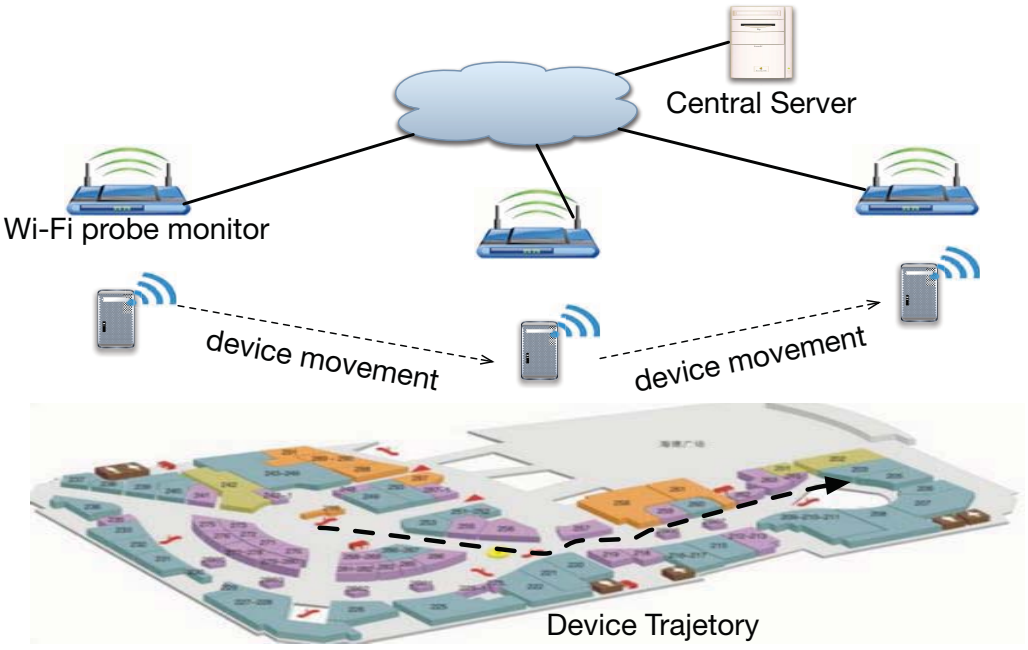


Figure 6.1: Overview of the Wi-Fi probe collection system

Device Mac Address	AP ID	Start Time	End time	Dwell Time
ff:ff:ff:ff:ff:ff	1	1388569044	1388569844	800

Table 6.1: Example of a single data record

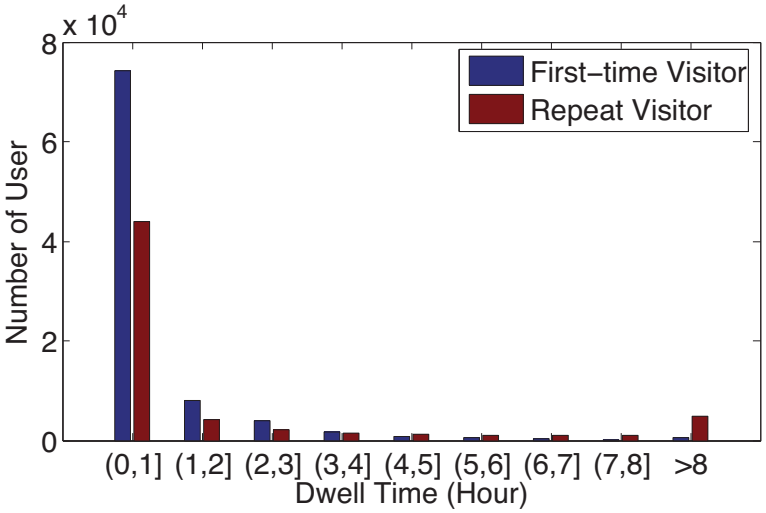


Figure 6.2: Dwell time distribution in the collected dataset

6.2.3 Details of Dataset

The dataset used in the work is collected in one of the biggest shopping mall in Shenzhen, China. The data is collected from 1 Jan, 2014 to 10 Jan, 2014. During that period, many families and friends will hang out to celebrate the new year, indicating that the dataset is quite informative and thus enabling us to study the impact of factors such as social context towards human mobility. As shown in Table 6.1, a single data record contains five attributes, including: the mac address of the device, the associated access point ID, start time and end time of being detected, and the dwell time. Specifically, the Unix time is used to record time stamp of the data.

In total, the dataset has 396772 records, and contains 111411 devices with unique mac address. Among all the monitored devices, 21260 devices are detected for at least two days, whose user is called repeat visitor, while 90151 devices are recorded in only one day, whose user is called first-time visitor. In this work, we assume each device is associated with a user. Therefore, we use device and user interchangeably in the work. Figure 6.2 shows the dwell time distribution of all the users. Particularly, most of the user dwell time is less than one hour. Among repeat visitors, around 10 percent of them stay for more than 8 hours. One possible explanation could be that those repeat

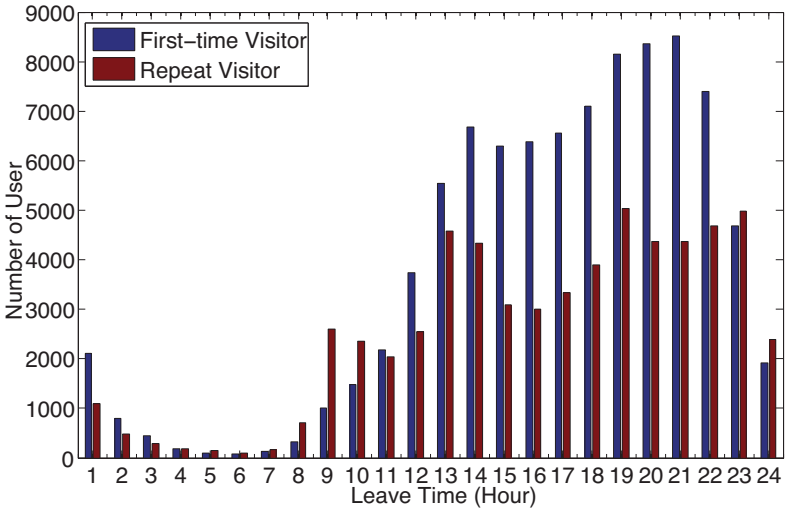


Figure 6.3: Leave time distribution in the collected dataset

visitors work in that place. The leave time distribution of users is presented in Figure 6.3. We observe that most of users will leave the place starting from 9 am to 12 pm.

6.3 System Overview

StayPredictor systems is based on Wi-Fi traces collected from mobile devices and able to predict dwell time of a given user in a place. Wi-Fi traces include the Wi-Fi probe request message and Wi-Fi connection information. As shown in Figure 6.4, StayPredictor system has four modules.

1. **Data pre-processing**: since the collection of Wi-Fi probe or connection data is not continuous in time domain, the function of this module is to conduct data aggregation to combine the discrete data records into a continuous one.

2. **Feature extraction**: this module is responsible for extracting a set of candidate features (factors) that may impact the dwell time of users. Because dwell time prediction task is a learning problem, we use the term feature to represent factor that associates with the dwell time. This work include three categories of features: time context, social context, dwell time and leave time history.

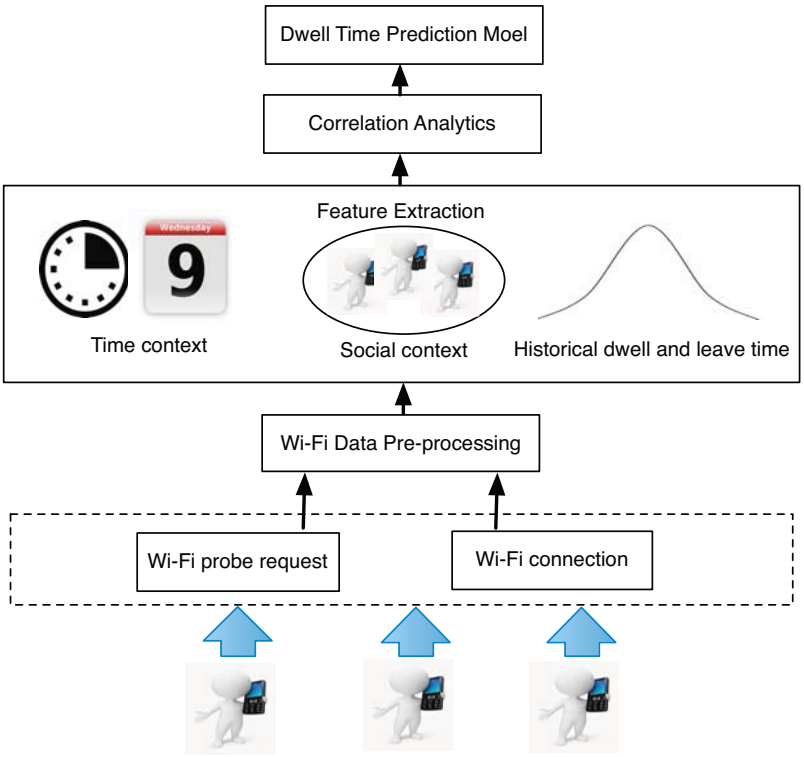


Figure 6.4: System architecture

3. **Correlation analytics**: after feature extraction, this module will quantify the dependency between features and user dwell time. Specifically, we adopt a variety of metrics such as Pearson correlation coefficient, Spearman's rank correlation coefficient, mutual information and standard deviation to measure the correlation.

4. **Dwell time prediction model**: this module is to output the predicted result of dwell time. In particular, the proposed dwell time prediction model is an ensemble model that integrates three individual models: time and social context based model, historical dwell time model and historical leave time model.

The problem of dwell time prediction can be formulated as finding a mapping function $f : x \rightarrow y$, where x is the feature space and y is the dwell time. In our case, feature space covers that factors that influence the dwell time. The rationale of identifying and quantifying the influence factors of dwell time is that it helps to improve the generality of the mapping function, as it filters out the non-related features.

6.4 System Design

6.4.1 Pre-processing

Data preprocessing focuses on data aggregation, which aims to combine the segmented data records of a give user. The reason for data aggregation is that although human movement is continuous, it may not be reflected by the collected Wi-Fi traces. First, the collection of Wi-Fi probe message is executed in a periodical manner. Second, the Wi-Fi interface of devices may not be on all the time. As a result, even though a user keeps staying in the same place, the collected data records are segmented.

To aggregate the segmented data records, we design a data aggregation algorithm. The algorithm is based on a simple assumption that: for a specific device at a certain spot, if the difference between the end time of a given record and the start time of the next record is less than a threshold, we consider the device keep staying in the spot from the start time of the given record to the end time of the next record. Here, the threshold is set as 0.5 hour. The details of the aggregation algorithm are presented in algorithm 7.

6.4.2 Feature Extraction

The objective of feature extraction is to extract a number of features that may associate with the user dwell time. Since human mobility is largely determined by daily routine and social relationship [CML11], we extract three feature categories including: arrival time context, social context, historical dwell time and historical leave time.

Algorithm 7: Data-aggregation algorithm

input:
Wi-Fi probe dataset collected at a specific hotspot
$D =< d_1 = (u_1, ts_1, te_1), ..., d_n = (u_n, ts_n, te_n) >$, where u_i denotes user ID, ts_i and te_i denote start time and end time respectively.
The total amount of record in D is N.
output:
Aggregated dataset AD.
Initialization:
Aggregation threshold δ;
Aggregated dataset $AD = \emptyset$;
Sort dataset D based on item u_i;
for $t = 1, ..., N-1$ **do**
    if $u_i == u_{i+1} \&\& ts_{i+1} - te_i < \delta$ **then**
        $d_{i+1} = (u_{i+1}, ts_i, te_{i+1})$
    end
    $AD = AD \cup \{d_{i+1}\}$.
end

Arrival time context

In fact, when human arrives at a certain place might correlate with the stay duration. The intuition behind this statement is that different arrival time may indicate different activities and different activities could lead to different lengths of stay. For example, if a user arrive a place at 9am in the weekday, he/she probably goes to the working place. Thus, he/she will stay until the end of working hours. In another case, if a user arrive at 12am, he/she may go to have lunch and the stay duration could be around one hour.

Three arrival time contexts are identified: hour of the day; day of the week; weekday/weekend indicator. First, we need to transform the time contexts into numeric variables, which can be incorporated into the prediction model. Hour of the day is represented by using a slot number ranging from 1 to 24. Slot number 1 means the time

interval [00:00, 01:00), while 24 means the time interval [23:00-24:00). Day of the week is represented with a number ranging from 1 to 7, where number 1 means Monday and number 7 means Sunday. Weekday/weekend indicator is a binary variable to differentiate weekday or weekend. In this work, 0 is used to represent workday and 1 is used to represent weekend.

Social context

According to sociology research [Pre12], human behaviour is influenced by social circle. Specifically, we focus on the two types of social circle, including: co-located users and social group. At the very beginning, we segment the data collection period into consecutive time intervals with the same length. In our experiment, the length is set as 100 seconds. The start time of first time interval uses the earliest start time among all the data traces.

For a given user, the co-located users refer to those who are being detected by the same monitor as him/her within the same time interval. Since the dataset does not have the ground truth for the explicit social group relationship, we need to make some assumptions. Here, a social group is defined as a set of users that exhibit two characteristics: first, they arrive at the place within the same time interval; second, the pairwise similarity of their dwell times is large than a threshold. The details of the algorithm are shown in algorithm 8. In particular, we are interested in extracting the amount of co-located user and the size of social group as the features for dwell time prediction model.

Historical dwell time

In addition to time and social context, human behaviour could be governed by the routine life. Therefore, we consider historical dwell time as a feature, as it may correlate

Algorithm 8: Social context extraction algorithm

input:

Wi-Fi probe dataset collected at a specific hotspot

$D =< d_1 = (u_1, ts_1, te_1, l_1), ..., d_n = (u_n, ts_n, te_n, l_n) >$, where u_i denotes user ID, ts_i and te_i denote start time and end time. l_i denotes the dwell time.

The total amount of record in D is N.

A user u_i who arrives at time ts_i.

output:

The amount of co-located user c_i of user u_i.

The size of social group g_i of user u_i.

Initialization:

Set the length of time interval as L.

Set the social group threshold as tH.

$Te = \max(\forall_i te_i)$.

$Ts = \min(\forall_i ts_i)$.

$M = (Te - Ts)/L + 1$.

Initiate M empty sets $S_1, ..., S_M$.

Initiate M empty sets $T_1, ..., T_M$.

for $t = 1, ..., N$ **do**

    $a = (ts_t - Ts)/L$

    $S_a = S_a \cup u_t$

    $T_a = T_a \cup l_t$

end

forall the $l_k \in T_b$ **do**

    if $|l_k - l_b| / \max(l_k, l_b) < tH$ **then**

        $G_a = G_a \cup u_k$

    end

end

$b = (ts_i - Ts)/L$

$c_i = |S_b|$

$g_i = |G_a|$

with the dwell time the future. For example, some users will go to a place for lunch regularly during the lunch time, and they will stay there for 1.5 hour. When they arrive next time, the potential dwell time will be considered as 1.5 hour.

Historical leave time

We can also predict the dwell time by estimating the leave time as the arrival time is known. The rationale of considering leave time is that compared with the dwell time, the leave time may not vary too much. For instance, a user normally leave a place at 10pm; if this time he arrives at 1pm, then the potential dwell time is 9 hours.

6.4.3 Features and Dwell time Correlation Analytics

We apply different techniques to measure the influence of features towards the dwell time. Since both social context and dwell time are continuous variables, Pearson correlation coefficient [ZTS03] and Spearman's rank correlation coefficient [Ken48] are adopted to quantify the linear and monotonic correlation relationship receptively. In addition, mutual information [Pre07] is used to measure the correlation among time context and dwell time, because time context belongs to categorial variable. Finally, we use standard deviation to assess the routine degree of dwell time and leave time. In the following, we will discuss the details of used techniques.

Pearson correlation coefficient

It is a measure of the degree of linear dependence between two continuous variables X and Y, where 1 indicates total positive correlation, 0 means no correlation, and -1 means total negative correlation.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{6.1}$$

where σ_X denotes the mean of X and μ_X refers to the standard deviation of X.

Spearman's rank correlation coefficient

It is a metric to assess how well two variables can be described using a monotonic function, where +1 or -1 indicates there is a perfect monotone function to incorporate both two variables. For a dataset of size n, the raw data X_i and Y_i are first transformed into ranks x_i and y_i. Spearman's rank correlation coefficient ρ is computed based on the following equation:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \tag{6.2}$$

where $d_i = x_i - y_i$ denotes the difference between ranks.

Mutual information

The mutual information of two variables is a measure of the mutual dependence among variables. For variable X and Y, mutual information could be calculated as follows:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{6.3}$$

where $p(x,y)$ refers to the joint probability of x and y; $p(x)$ and $p(y)$ denote the probability of x and y respectively.

Note that dwell time is a continuous variable, it needs to be transformed as a categorial variable to fit the mutual information measurement. Therefore, we cluster dwell time by using k-mean clustering. Given the predefined number of clusters and the data,

k-mean algorithm will automatically assign the data samples into different clusters. In the evaluation section, different number of clusters are chosen.

Standard deviation

This is a metric to measure the degree of variation from the average. A high standard deviation indicates a low degree of repetition. We apply standard deviation to measure the routine degree of dwell time and leave time of users. Before the measuring standard deviation, we will normalize the dwell time data based on feature scaling, to make this metric equal for users with long dwell time and short dwell time.

6.4.4 Dwell Time Prediction Model

Time and social context based model

Time and social context based model is a single prediction model for user dwell time, based on the features extracted from arrival time context and social context. In particular, among social contexts, only co-located users is considered in this model. The main reason is that social group may not be captured when a user arrives, since it requires the complete visit traces of users.

This predictive model is based on a linear ridge regression model. Different from traditional linear regression model, linear ridge model imposes a penalty on the size of coefficients, which is able to mitigate the effect of linearity among factors. Within the linear ridge regression model, only the weight vector $w = [w_1, w_2, ..., w_i]$ of different features needs to be learned from the training dataset, Basically, the parameter w can be determined by minimising the linear least squares between the predicted result Xw and the ground truth dwell time y with l2 regularization. The objective function of learning the parameter w is shown as below:

$$\min_{w} ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{6.4}$$

where α is a complexity parameter to control the amount of shrinkage. In this work, α is set to 5. The learning algorithm has a cost of O(np^2), given X a matrix of size (n, p) and assuming n $\geq$ p.

Since the regression model may give the negative value, which contradicts with the reality. Therefore, we proposed a contrained time and social context based prediction algorithm, which is shown in algorithm 9. The basic idea is that if the result of linear ridge regression model is positive, then the dwell time is estimated as $d_1 = Xw$, where X is the features extracted from the arrival time context and social context. Otherwise, the dwell time is estimated using the average dwell time in the past.

Algorithm 9: Constrained time and social context based prediction algorithm

input:
Features extracted from time and social context: X.
Pre-trained parameter w.
Average dwell time in the past m.
output:
Predicted dwell time dt.
if $Xw > 0$ **then**
    $dt = Xw$
else
    $dt = m$
end

Historical dwell time model

The main idea of this model is to exploit the dwell time data in the past to predict their dwell time in a place in the future. Specifically, we use a normal distribution for

modelling the conditional distribution of dwell time y under context x_k as follows:

$$P(y|x_k) = \frac{1}{\sqrt{2\pi{\sigma_x}^2}} \exp(-\frac{(y-\mu_x)^2}{2\sigma_x^2}) \tag{6.5}$$

where μ_x and σ_x are the mean and standard deviation of the historical dwell time data of a particular user. Then when a user arrive, dwell time will estimated with a value that has the highest likelihood under the context x_k, that is $d_2 = \max_y P(y|x_k)$.

Historical leave time model

To predict the dwell time of a user, we could estimate his/her leave time indirectly. Historical leave time model is to estimate the leave time in the future based on the historical leave time distribution. Similar to historical dwell time, we adopt normal distribution to model the conditional distribution of dwell time y given context x_j as follows:

$$P(k|x_j) = \frac{1}{\sqrt{2\pi{\sigma_j}^2}} \exp(-\frac{(k-\mu_j)^2}{2\sigma_j^2}) \tag{6.6}$$

where μ_j and σ_j are the mean and standard deviation of the historical dwell time data of a particular user. Suppose a user arrive at time t_j, leave time will estimated with a value that has the highest likelihood of the historical leave time distribution under the context x_j and then the dwell time can be calculated with leave time minus arrival time: $d_3 = \max_k P(k|x_j) - t_j$.

Ensemble model

The key idea of ensemble model is to incorporate several individual models, in order to improve the predictive performance. Here, we combine three single prediction models: time and social context based model, historical dwell time model and historical leave time model, to build up an ensemble model. The ensemble model is developed via

integrating three single models linearly with a weight vector w as follows.

$$D = \sum_{i=1,2,3} w_i d_i \tag{6.7}$$

The strategy of weight estimation is to assign more weight to models that are more accurate. In practise, the weight w can be learned based on the training dataset. In this work, we simply assign the equal weight to each model. The details of the model performance are presented in the evaluation section.

6.5 Experimental Evaluation

In the following section, we are interested in evaluating the performance of our proposed prediction model for user dwell time. Specifically, we will present the details of the the evaluation metrics, the baseline methods and the experimental results.

6.5.1 Evaluation Metrics

To compare the effectiveness of different models, we propose a measurement metric: relative error. The relative error is measured by the ratio of the absolute difference between the prediction result and ground truth to maximum value among the prediction result and ground truth. For prediction result list X and ground truth list Y, the relative error [DGP12] can be calculated based on the following equation.

$$error(X,Y) = \Big(\sum_{\forall x_i \in X, y_i \in Y} \frac{|x_i - y_i|}{\max(x_i, y_i)} \Big) \Big/ |X| \tag{6.8}$$

Compare to the absolute error, the relative error is able to exhibit equality to performance evaluation in visitors with long dwell time and short dwell time.

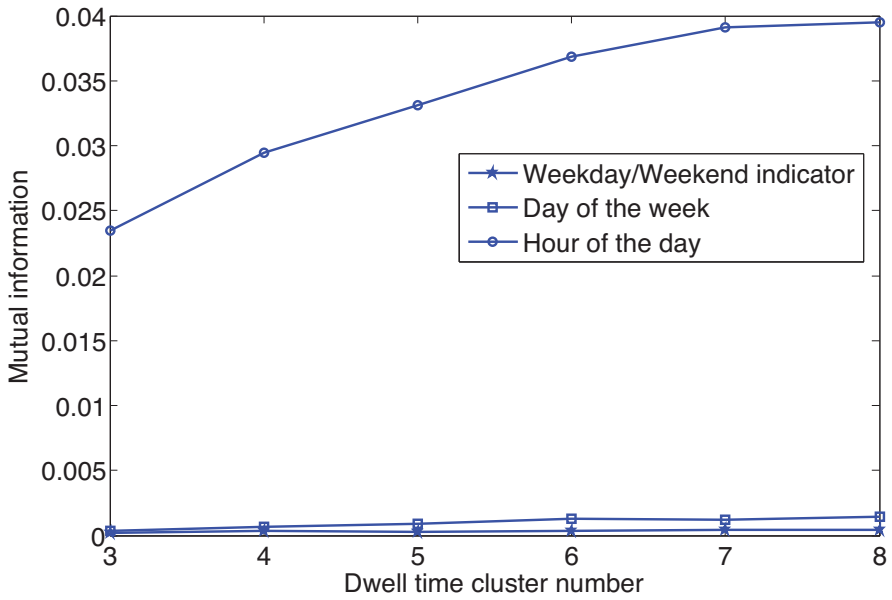


Figure 6.5: Mutual information among time context and dwell time for first-time visitor

6.5.2 Baselines

We consider two non-trivial baseline methods for comparison. The first baseline is time and social context model, which is based on linear regression model, which inputs the time and social context and outputs the predicted dwell time.

The second baseline is the historical dwell time model. The main idea of this model is to leverage the average dwell time of a certain user in the past to predict his/her future dwell time. Despite its simplicity, this model is a very strong baseline. Assume a user's dwell time follows Gaussian distribution, then the average dwell time of a user in the past lies very close to the peak of the distribution. It indicates that the output of historical dwell time model presents a higher probability to get close to the ground truth and it is a preferable model without any other prior information.

6.5.3 Understanding the Impact of Features towards Dwell Time

This section presents the quantitative measurement for the correlation among the proposed features and dwell time.

Time context vs dwell time. We begin by investigating the effect of time context on human dwell time at a given place. One way to quantify the dependency between the time context and the dwell time is to measure their mutual information which stems from information theory. The lower the mutual information score, the less they are correlated. The score 1 indicates perfect correlation, while score 0 means no correlation. Specifically, Figure 5 plots the mutual information between time context and first-time visitor dwell time given different cluster number for dwell time clustering. Specifically, the dwell time is clustered based on different cluster numbers ranging from 3 to 8. For repeat visitor, the mutual information between time context and dwell time is illustrated in Table II. The results show that for both first-time visitor and repeat visitor, hour of the day feature has much more mutual information with dwell time compared with day of the week and weekday/weekend indicator. One explanation could be different hour slot means different activities in a place. For example, if a user arrives at 12 am, which means he will probably go for a lunch and will stay for a short time.

Cluster Number	**Weekday/Weekend Indicator**	**Day of the week**	**Hour**
3	0.000048	0.000954	0.097421
4	0.000058	0.001030	0.111517
5	0.000247	0.001294	0.121165
6	0.000237	0.001359	0.126000
7	0.000223	0.001423	0.132192
8	0.000233	0.001514	0.132962

Table 6.2: Mutual information among time context and dwell time for repeat visitor

Social context vs dwell time. The objective of this experiment is to evaluate the impact of social context upon the dwell time of first-time visitor and repeat visitor. In order to quantify the impact, we use Pearson correlation coefficient and Spearman's rank correlation coefficient as the metrics, which can measure degree of the linear and monotonic dependency respectively. The correlation coefficient 1 or -1 indicates perfect correlation, while score 0 means no correlation. To evaluate the correlation between social

context and dwell time, we set the segmented length of time $L = 100$, and the social group threshold tH to be 0.2, 0.4 and 0.6.

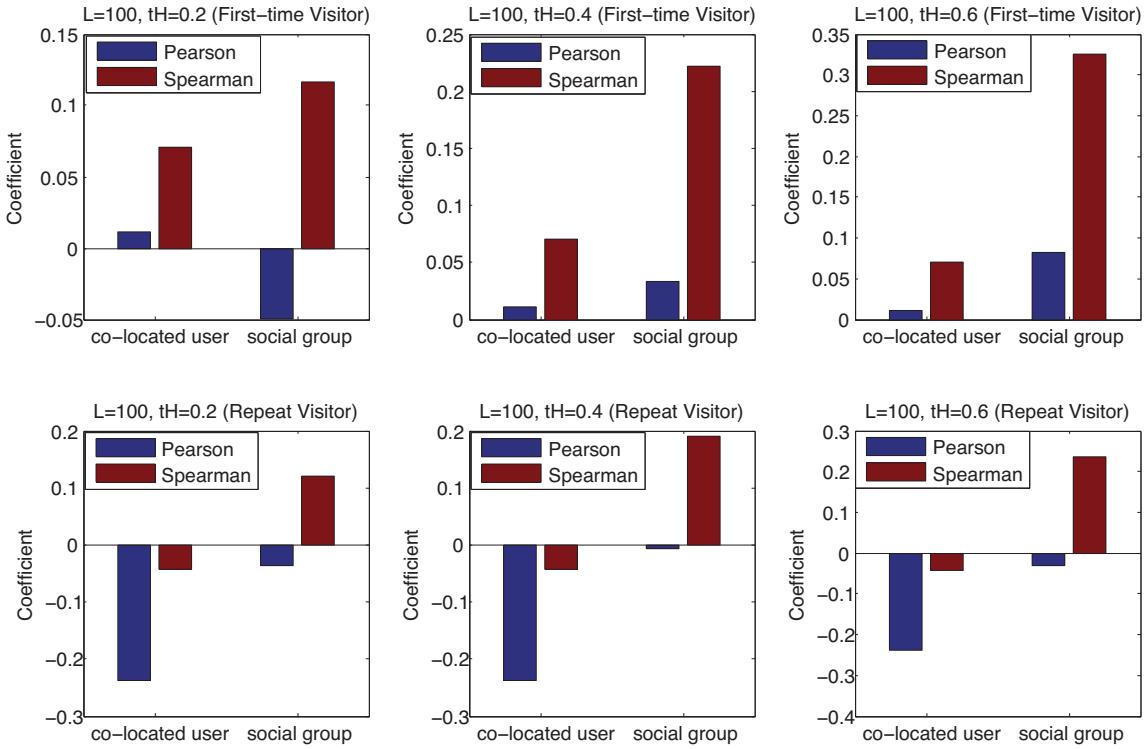


Figure 6.6: Correlation between social context and dwell time

From Figure 6, we observe that generally the dwell time of a user is positively correlated with the size of social group during the visit. This means that if a user go for a visit with more people, the longer he/she will stay in that place. Interestingly, the dwell time of first-visitor is more correlated with the size of the social group, whereas the dwell time of repeat-visitor is more correlated with the amount of co-located users. In particular, the dwell time of repeat visitor is negatively correlated with the amount of co-located user at the arrival time. Above observations reveal that social context does influence human movement behaviour.

Routine degree of dwell time. We now turn our attention to study the routine factors that affect the dwell time of repeat visitors. We are interested to understand the routine degree of dwell time of different repeat visitors with respect to visit frequency. In this case, we use standard deviation (SD) to measure the routine degree. A low stan-

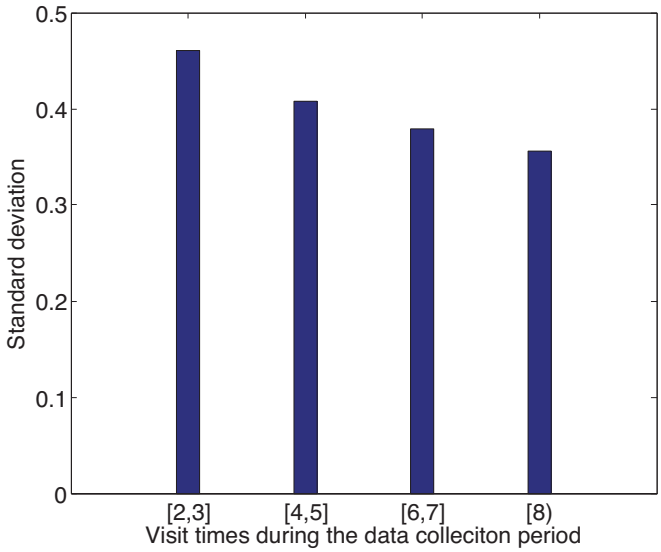


Figure 6.7: Routine degree measurement of dwell time

dard deviation indicates that the dwell time exhibit a high degree of repetition, while a high standard deviation means that the dwell time varies and spreads out over a large range of values. As illustrated in Figure 7, we first classify the repeat visitor into four categories according to visit times during the data collection period, including: [2,3], [4,5], [6,7], [8,). The result reveals that the more frequent a user visit a place, his/her dwell time will exhibits a higher routine degree. This result conforms to our common sense because users with a high visiting frequency probably work in that place or regularly come for dining. Due to the routine working hour, their dwell times present a high degree of repetition.

Routine degree of leave time. Given the arrival time, the dwell time of user can be predicted if the leave time could be approximated. If the leave time of user exhibit a high routine degree, we can leverage the past leave time and the current arrival time to estimate the user dwell time. This experiment is to measure the routine degree of user leave time. Similar to the evaluation on the routine degree of user dwell time, we focus on exploring the routine degree of leave time of different repeat visitors. We first classify the repeat visitor into four categories according to visit times during the data collection period, including: [2,3], [4,5], [6,7], [8,). The evaluation result shown in Figure 8 shows

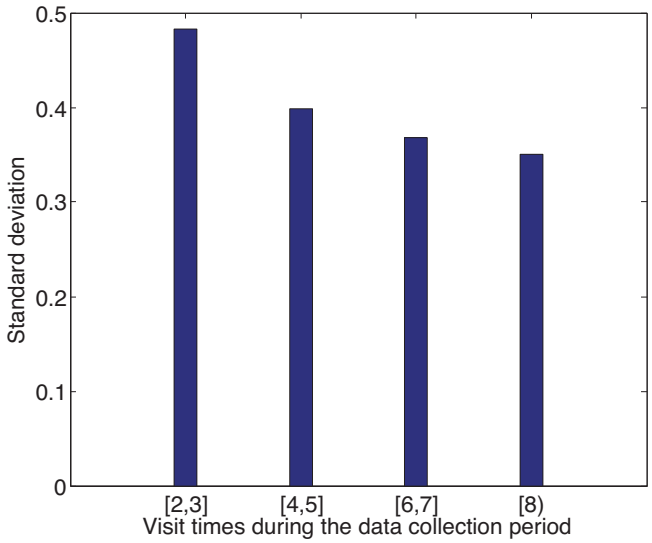


Figure 6.8: Routine degree measurement of leave time

that with a higher visit frequency, the dwell time of a user will present a lower standard deviation and thus indicating a higher routine degree.

6.5.4 Dwell Time Prediction Results

Dwell time prediction model	**Error**
Time and social context based model (linear regression)	0.414
Time and social context based model (linear lasso)	0.406
Time and social context based model (linear ridge)	0.390
Historical dwell time model	0.392
Historical leave time model	0.360
Ensemble model	0.346

Table 6.3: Dwell time prediction performance of different models

Prediction accuracy comparison on different models. First, we compare the performance of the proposed model with the performance of the baselines. In the following, in order to make the regression model converge, we only select the data of repeat visitors with at least 6 visit times during the data collection period. By default, 50% of the dataset will be used for training and the rest 50% is used for test. Table III shows

the prediction error of five models. The first baseline time and social context model based on linear regression achieves 0.406 error and the second baseline historical dwell time model achieves a better performance with 0.393 error. Our proposed time and social context model based on linear ridge regression gives 0.356 error. Among all the individual model, the historical leave time model achieves the best performance, giving 0.334 error. Compared to all the other individual model, the ensemble model which combines linearly three best individual models, exhibits the best performance with 0.326 prediction error.

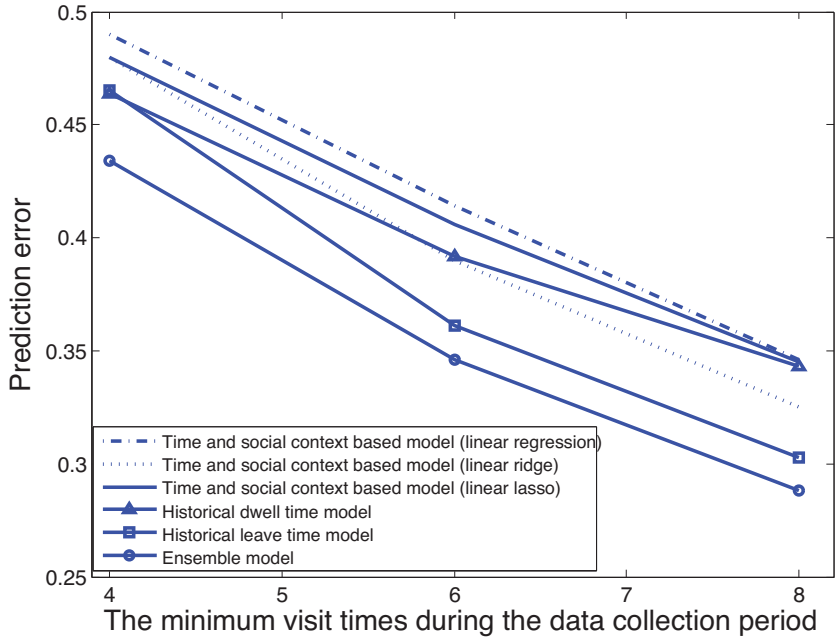


Figure 6.9: Dwell time prediction model performance in the different user datasets

Besides, we study the average error for different repeat visitors with different visit times within the data collection period. The repeat visitor dataset will be filtered based on different minimum visit times: including: 4, 6 and 8. For example, by setting minimum visit times to be 4, only those users appearing in the place for at least 4 times will be included in the dataset. The prediction algorithm will be evaluated on above three datasets. Figure 9 shows that for all the models, the prediction error for users with high visit frequency is lower than those with low visit frequency. The results provides an insight that users with high visit frequency is easier to be predicted. This conclusion

also matches the observation in routine degree measurement of dwell time and leave time, that is, the dwell time and leave time of a user with a higher visit frequency present a lower standard deviation.

Impact of training dataset on prediction accuracy. Finally, we evaluate how the percentage of training dataset impacts the prediction performance. From the result shown in Figure 10, we can see that the performance of prediction model gets better with more training set. Specifically, when the 70% of data is used for training, the prediction error of linear ridge model is 0.364, historical dwell time model and historical leave time model achieves 0.372 and 0.327 error respectively. The ensemble model gives the lowest error 0.310.

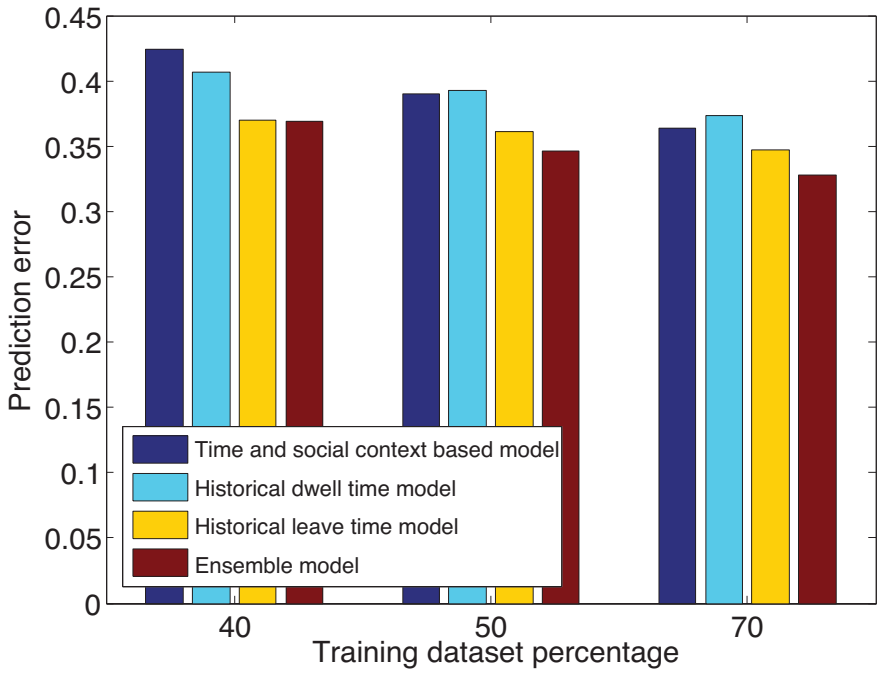


Figure 6.10: Dwell time prediction model performances with different amount of training data

6.6 Summary

In this work, we aim to identify the fundamental factors that influence user dwell time in a place and then quantify the influences. Quantifying the influences of factors towards dwell time not only advances our understandings of user behaviour, but also offers guidelines for accurate dwell time prediction. Based on correlation analytics of 10-day Wi-Fi traces collected from 111K devices in a large shopping mall, we found that the visitor dwell time was positively correlated with the size of social group during the visit. Interestingly, the dwell time of repeat visitors was negatively associated with the size of co-located user. Furthermore, repeat visitors exhibited a high degree of repetition in dwell time and leave time. We believe those findings could provide new insights for human mobility analytics.

To our knowledge, this is first work to identify and evaluate the impact of social context towards user dwell time, and incorporate social context into the dwell time prediction model. Evaluation results demonstrate that the proposed ensemble model is able to achieve higher accuracy of predicting the dwell time of users over single models. In the future, we will extend the proposed framework to predict future user trace inside a shopping mall.

Chapter 7

Non-intrusive Stress Detection Based on Seating Pressure

In this chapter, we study the problem of stress detection based on seating pressure patterns. This chapter is organised as follows. In Section 7.1, the overview of the thesis is presented. Next, we provide the detailed information about the experiment design in Section 7.2. In Section 7.3, we describe the stress detection framework including: feature extraction, correlation analysis and stress classification. The experiment results are presented in Section 7.4. We discuss the limitations of this work in Section 7.5 and we conclude the thesis in Section 7.6.

7.1 Overview

Stress is one of the major problems in modern society. Studies have found that stress can cause many health problems such as depression, anxiety and cardiovascular diseases [CKG97]. Stress detection can help people gain more awareness of their stress levels and the associated factors, and thus being able to reduce stress. For instance, if a person experienced more stress when s/he excised less, which can motivate user

to conduct more physical exercise to relieve stress. Furthermore, stress detection enables computer to be more user friendly in the context of human-computer interaction. When a user is detected to be stressed, for instance, then the computer could perform some soothing interventions such as playing music.

Several technologies have been developed to measure stress, including surveys, physiological signal measurement (blood pressure [VvDdG00], heart rate variability [DNG+00], skin conductance [HMP11] [SAS+10], cortisol [DK04] [vEBNS96]). However, these methods are intrusive, which require the cognitive attention of the users. We argue that an ideal stress detection system for long-term monitoring should be unobtrusive, without posing additional stress upon people. A less intrusive approach to measure stress is analysing the user behaviours such as typing pattern [HPRC14] and mobile phone usage [LLLZ13], which correlate with stress.

In this work, we explore the possibility of detecting stress based on the seating pressure distribution on a chair. Our idea is motivated by the embodied theory of cognition [Cla97], which indicates that affective states of people are manifested in their posture channels. Specifically, this work seeks to answer two research questions. First, what seating pressure features differ between stressed and relaxed states? Second, can stressed and relaxed states be classified using machine learning methods?

In particular, we conduct a laboratory study to collect seating pressure data from 15 participants using a seat cushion which is deployed with 20 pressure sensors. Through correlation analysis, we identify a number of seating pressure features that are associated with stress, including: average seating pressure, pressure imbalance, etc. Moreover, we build up a stress detection framework based on the associated features. The evaluation result shows that the proposed stress detection framework can achieve 86% accuracy.

The primary contributions of this work is as follows.

- We present a set of effective seating pressure features in both temporal and

spatial domain, which are able to distinguish user's stressed and relaxed states.

- We find that stressed subjects reveal higher average seating pressure, pressure deviation and seating imbalance.

- Based on the selected seating pressure features, the proposed stress detection framework can achieve 86% accuracy using kNN classifier,

7.2 Experimental Procedure and Data Collection

The goal of this work is to study the feasibility of using pressure sensor-based cushion to capture the manifestation of stress. Thus, we design several tasks that are conducted by participants under stressed and relaxed situations. In the following, we present the details of data collection, experiment tasks and procedure.

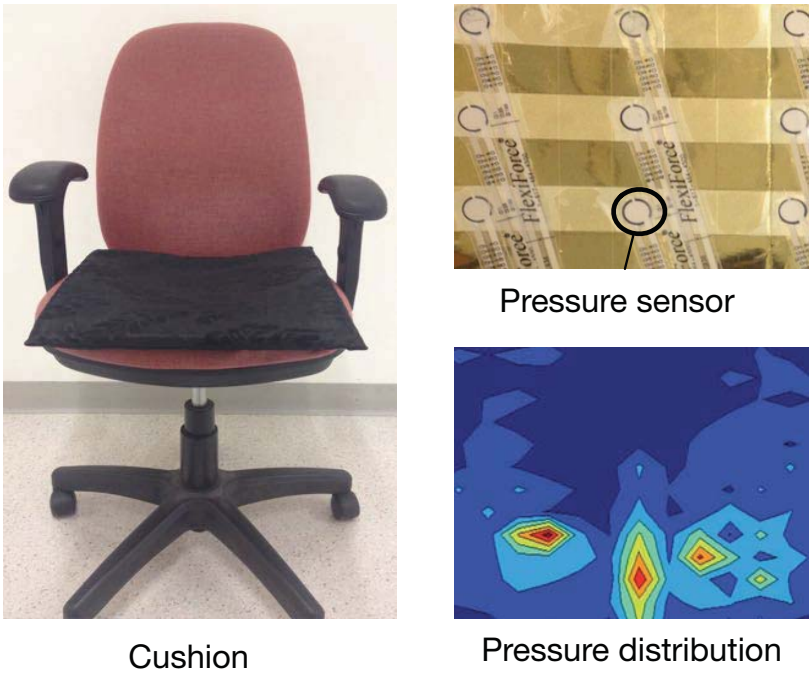


Figure 7.1: Pressure sensor-based cushion

7.2.1 Participants

Fifteen participants (15 males) are recruited in this study. The average age of the participants is 26 with a standard deviation of 3.17. The minimum age is 23 and the maximum age is 35. The weight (kg) of the participants ranges from 50 to 87, with the average weight of 65 and a standard deviation of 10.84. The highest education levels for the participants include: bachelor degree (9), master degree (3) and doctor degree (3).

7.2.2 Seating pressure data collection

To enable long-term stress monitoring, stress measurement needs to be unobtrusive. In this study, we aim to identify stress of users from their seating pressure information, which can be captured by a pressure sensor-based cushion mounted on a chair.

The pressure sensor-based cushion used in this work (see Figure 7.1) is a pressure sensor array, consisting of 4-by-5 array of pressure sensors distributed over an area of 40*40 centimeters. In particular, FlexiForce A201 [Tek] is chosen as our pressure sensor, because it's very thin (0.208 mm thick) and very stable (less than 3% linearity error), which can provide non-intrusive user experience. The pressure sensor reading is encoded as a 16-bit digital value and the sampling rate is 15Hz. The sensors are equally divided into four groups. At each sampling period, we will sample one sensor in each group simultaneously.

7.2.3 Experimental Tasks

We design some experiment tasks which induces both stressful and relaxed states, so that we can examine whether people exhibit different seating pressure patterns under different affective states. Since cognitive load is the most frequent stressor during daily

activity, we include two stressors that induce cognitive stress including: Stroop color-word test and mental arithmetic task.

Stroop Color-word Test: The Stroop test [Gol78] has been widely used as a cognitive stressor. In this test, a set of words describing colour are first shown on the screen, then the participants are required to type in the first letter of font colour of the words. During the test, participants will experience Stroop effect which has been demonstrated to elicit mental stress. Particularly, a ticking clock sound is played during the task. Furthermore, the amount of remaining time is notified every ten seconds. The duration of the test is sixty seconds.

Mental Arithmetic Task: This task [KMY11] introduces cognitive stress via performing a series of arithmetic operation. In this study, the participants are requested to carry out arithmetic subtraction. The arithmetic subtraction contains a series of subtraction from 1029 by using number 7. The correct answers will be informed to the subjects once the miscalculation occurs. After being informed the miscalculation, the subject will continue the task until the task time is due. During the mental arithmetic task, a number of stressors are further introduced to mimic highly stressful environment. First, participants are required to finish the task as soon as possible. Second, remaining amount of time is notified every ten seconds during the task. Third, a loud traffic noise is played throughout the task. The duration of the task is sixty seconds.

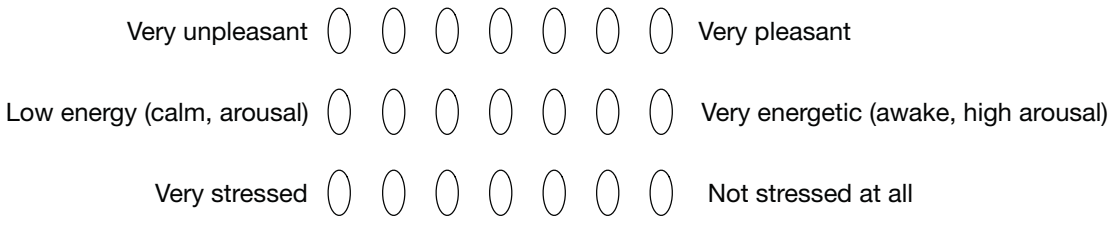


Figure 7.2: Stress, valence and arousal level survey after task

Neutral Task: In addition to capture the stressed states, we collect data when participants are relaxed. In this task, participants will listen to the soothing music for sixty seconds.

7.2.4 Stress Measurment

Although several experimental tasks are designed for participants to perform, it is still unknown whether those tasks elicit the intended emotions or not. Thus, after the completion of each task, participants are required to report their valence, arousal and stress levels on a 7-point Likert scale (see Figure 7.2).

7.2.5 Protocol

In order to examine the seating pressure differences between stressed and relaxed conditions, we perform a laboratory study. The procedure is shown in Figure 7.3. First, a briefing session will be held to introduce the participants the purpose and procedures of the experiments. After providing the written consent, participants are requested to fill out some demographic information. Next, participants go through a training session, so as to reduce the novelty effects of experimental conditions and procedures. During the training session, they have a short tutorial on how to conduct Stroop color-word test and mental arithmetic task.

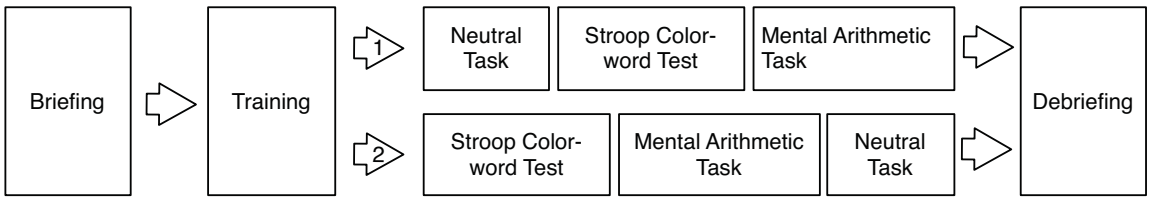


Figure 7.3: Experimental Procedure for Participants

After the training session, participants perform three tasks: two stressed tasks and one neutral task. The participants are seated on a height-adjusted spinning chair. In particular, the ordering of stressed tasks and neutral task are counterbalanced. Half of the participants will start with the stressed tasks, and continue with the neutral tasks. The other half start with the neutral tasks and then continue with the stressed tasks. After the completion of each task, participants are required to report their valence, arousal and stress levels. Finally, a debriefing session is conducted. Participants will

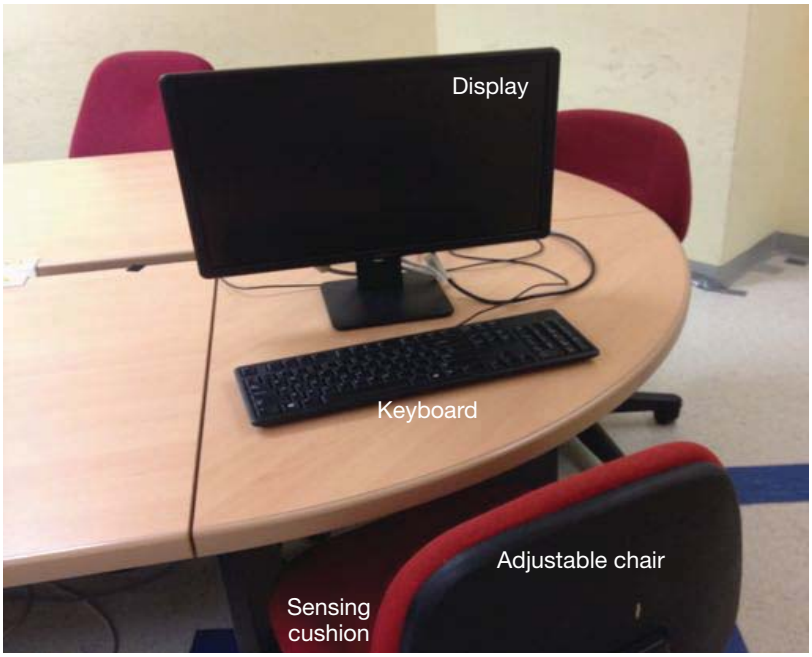


Figure 7.4: Experimental setup

be informed about the purpose of this experiment, and provide feedback and comments about the experiment.

Figure 7.4 shows the setup of the experiment. The same pressure sensor-based cushion will be used to collect sitting pressure data of all the participants. The lighting conditions are also the same for all the participants.

7.3 Stress Detection Framework

7.3.1 Feature Extraction

In order to identify the association between stress and seating pressure, we need to extract a number of seating pressure features. Feature extraction is necessary since the original seating pressure signal is a sequence of 20-dimensional data points, while the stress measurement is only conducted after the completion of the tasks. As a consequence, it does not work when directly applying the correlation analysis techniques

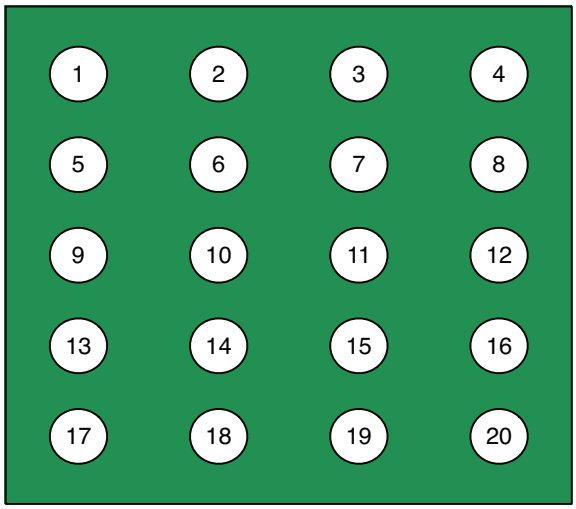


Figure 7.5: Pressure sensor placement in the cushion

to analyse seating pressure and stress data. In the following, we use $d_{i,t}$ to represent the ith sensor data in time t. The ordering of sensors are illustrated in Figure 7.5. A sampling interval is denoted as T.

Since human movement while sitting is very low-frequency, the information that can be obtained from frequency domain is fewer than time or spatial domain. Thus, we extract seating pressure features in both time domain and spatial domain, rather than frequency-domain features. In our work, time-domain features include pressure variance and spatial-domain features include pressure std, pressure imbalance, etc.

Average pressure. It refers to the mean value of all the pressure sensor data. During the task section, a sequence of average pressure data can be extracted. In order to enable the correlation analysis, we compute the mean value of the sequence of data, which reduce a time series to a single data point. Average pressure can be computed based on the following equation, where nT refers to the time length of data.

$$f_1 = \frac{\sum\limits_{t=T,\ldots,nT} ((\sum\limits_{i=1}^{20} d_{i,t})/20)}{n} \tag{7.1}$$

Pressure std. It represents the standard deviation of all the pressure sensor data.

Similar to the process of average pressure feature, pressure std can be calculated using below equation.

$$f_2 = \frac{\sum\limits_{t=T,\ldots,nT} (\sqrt{\sum\limits_{i=1}^{20} (\overline{d}_{i,t} - d_{i,t})^2 / 20})}{n} \tag{7.2}$$

Max pressure. It means the maximum value of sensor data. This feature can be computed using the equation as follows.

$$f_3 = \frac{\sum\limits_{t=T,\ldots,nT} (\max(\underset{i=1-20}{\forall} d_{i,t}))}{n} \tag{7.3}$$

Pressure variance. It measures the sum of the standard deviation of each sensor data within a time interval. Pressure variance can be computed as follows.

$$f_4 = \frac{\sum\limits_{i=1}^{20} (\sqrt{\sum\limits_{t=T}^{nT} (\overline{d}_{i,t} - d_{i,t})^2 / n})}{20} \tag{7.4}$$

Pressure imbalance. It measures the difference between the sum of sensor data in the left block and the one in the right block. It can be calculated with the following equation.

$$f_5 = \frac{\sum\limits_{t=T,\ldots,nT} (\max(\sum\limits_{i \in L} d_{i,j} / \sum\limits_{i \in R} d_{i,j}, \sum\limits_{i \in R} d_{i,j} / \sum\limits_{i \in L} d_{i,j}))}{n} \tag{7.5}$$

Where $L = [1, 2, 5, 6, 9, 10, 13, 14, 17, 18]$ represents the index set of sensors in the left block, while $R = [3, 4, 7, 8, 11, 12, 15, 16, 19, 20]$ denotes the index set of sensors in the right block.

Top-k percentage. It means the ratio of the sum of top-k (k=1-10) sensor data to the sum of all the sensor data.

$$f_6 = \frac{\sum\limits_{t=T,\ldots,nT} (\sum\limits_{i\in K} d_{i,j} / \sum\limits_{i=1}^{20} d_{i,j})}{n} \tag{7.6}$$

where K represents the index set of top-k (k=1-20) sensors.

7.3.2 Correlation Analytics

To measure the correlation between seating pressure features and stress, Pearson correlation coefficient [ZTS03] and Spearman's rank correlation coefficient [Ken48] are adopted to quantify the linear and monotonic correlation relationship receptively.

Pearson correlation coefficient is a measure of the degree of linear dependence between two continuous variables X and Y, where 1 indicates total positive correlation, 0 means no correlation, and -1 means total negative correlation.

Spearman's rank correlation coefficient is a metric to assess how well two variables can be described using a monotonic function, where +1 or -1 indicates there is a perfect monotone function to incorporate both two variables.

7.3.3 Classification

After feature extraction and correlation analytics, a number of highly correlated features can be obtained. In this section, we need to design a classification model that can differentiate whether participants are stressed or relaxed based on seating pressure features. For simplicity, the states of stressed group and control group are labeled as stressed and relaxed respectively.

In this study, the classification model adopts the one-size-fits-all paradigm, where a single stress detection classifier is trained for all the participants. The advantage of training a classifier for all the participants is that it can improve the generality of the

model. However, the drawback is that it may not achieve the best performance for a particular participants. Specifically, two classifiers K-nearest neighbors (kNN) and support vector machine (SVM) are adopted in this work.

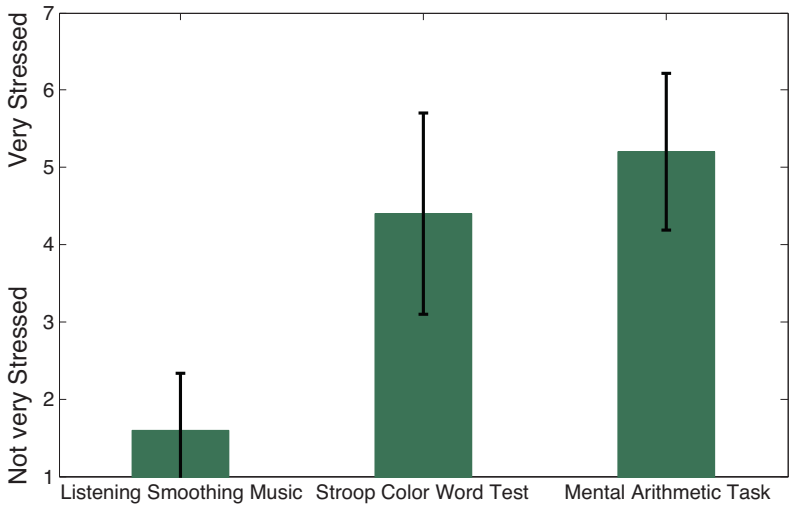


Figure 7.6: Average and standard error of self-reported stress

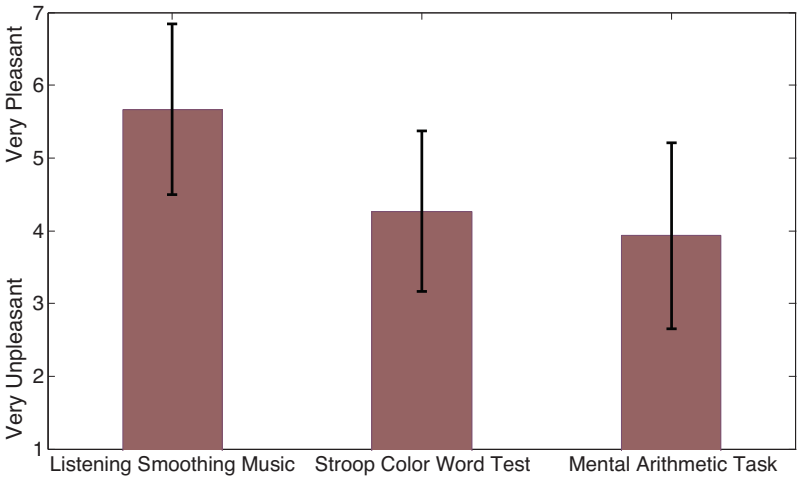


Figure 7.7: Average and standard error of self-reported valence

7.4 Results

In this section, we first evaluate the effectiveness of the tasks to induce the intended states. Second, we analyse the correlation between different seating pressure features

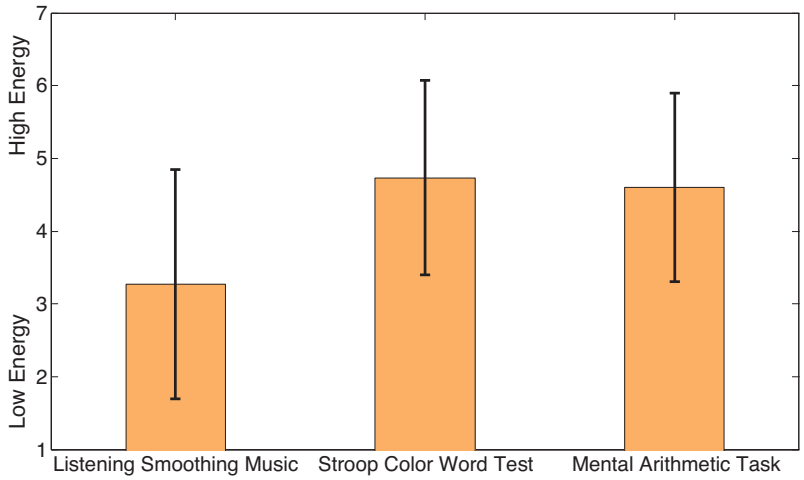


Figure 7.8: Average and standard error of self-reported arousal

and stress. Finally, we present the performance of stress detection based on the sitting pressure features.

Wilcoxon Rank Sum test is used to test whether significant difference exhibits between two distributions. The advantage of Wilcoxon Rank Sum test is that it is applicable to both normal and non-normal distributions. Two distributions are considered to be significantly different if $p < 0.05$.

7.4.1 The effectiveness of the tasks

The purpose of the three tasks is expected to elicit specific emotions, including relaxed and stressed states. To check whether the tasks have induced the expected emotions or not, we measure the stress of participants via self-reports.

The average mean and standard error of the self-reported stress is shown in Figure 7.6. Clearly, the self-reported stress is significantly higher for Stroop color-word test and mental arithmetic task than neutral task. This result indicates that the participants react to the stressors induced in the experiment. Particularly, mental arithmetic task induces more stress to participants than Stroop color-word test. As expected, Figure 7.7 shows that participants show significantly higher valence during the relaxed conditions, and

significantly higher arousal during the stressed conditions (see Figure 7.8).

7.4.2 Correlation among seating pressure features and stress

Table 7.1: Correlation among features and stress

Feature	Pearson correlation coefficient	Rank correlation coefficient
Average pressure	0.18	0.10
Pressure variance	0.01	-0.08
Max pressure	-0.08	-0.04
Pressure std	0.20	0.01
Pressure imbalance	0.27	0.14
Top-1 percentage	0.22	0.08
Top-2 percentage	0.15	-0.01
Top-3 percentage	0.13	-0.06

Table 7.1 shows that correlation coefficient result of different features. Particularly, average pressure, pressure variance, pressure imbalance and top-k (k=1,2,3) are positively correlated with stress. One possible interpretation is that stress might influence the muscle activities. As a result, the changes in the muscle activities could be revealed in the seating pressure.

We further conduct Wilcoxon Rank Sum Test to evaluate whether significant difference exhibits between the stressed and relaxed conditions in terms of those correlated features. Participants show higher average pressure in the stressed conditions than the relaxed conditions, although it is not significant ($Z = 1.56$, $p = 0.120$). No significant differences are found between the stressed and relaxed conditions in terms of pressure variance, pressure imbalance and top-k (k=1,2,3).

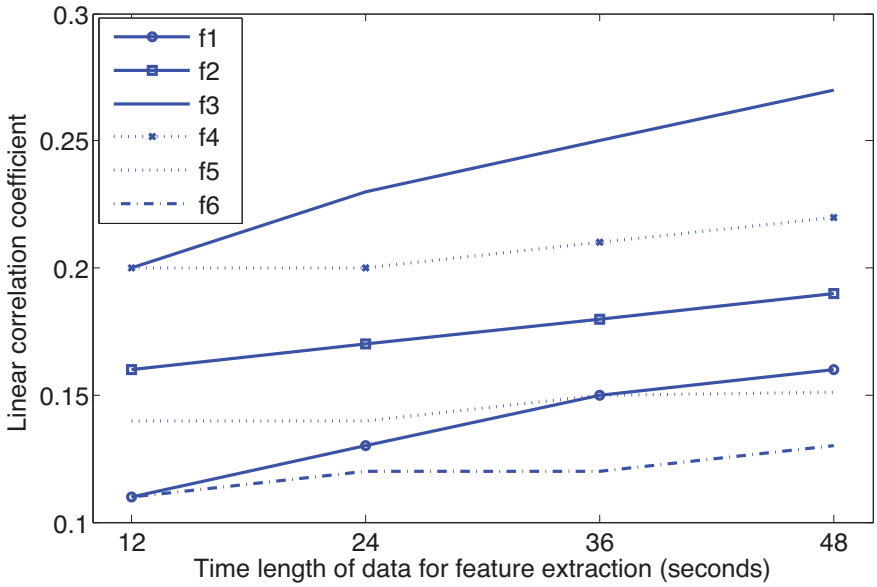


Figure 7.9: Impact of time length of data towards feature extraction (f1= Average pressure, f2 = Pressure variance, f3 = Pressure imbalance, f4 = Top-1 percentage, f5 = Top-2 percentage, f6 = Top-3 percentage)

To study the impact of time length of data towards feature, we select different time length nT to be 12s, 24s, 32s and 48s. Figure 7.9 plots the linear correlation coefficients of different features under different time length. Note that the correlation coefficient of seating pressure feature generally increases when the time length increases. This result shows that the features become more discriminate for the stressed and relaxed conditions with a larger time length.

7.4.3 Stress classification performance

The main objective of the this experiment is to evaluate the performance of stress classification. First, we segment the original data according to a fixed interval (3 seconds). Then the feature extraction is conducted on each data segment. Note four features with high correlation coefficient are included in the classification: average pressure, pressure std, pressure imbalance and top-1 percentage. In this study, 50% data will be used for training and the rest 50% data is used for test.

The confusion matrix of the classification is shown in Table 7.2. The true positives (stressed) are 133, and the true negatives (relaxed) are 60. False positives are 9 and false negatives are 23. There are more stressed data samples, since we collect data from two stressed tasks and one neutral task. Interestingly, although the individual feature does not show significance, a combination of features can still have high discriminative ability because some of features are complimentary to each other.

Table 7.3 summarises the classification result of three classifiers. kNN classifier (k=2) achieves the best performance with 86% test accuracy, while 69% test accuracy is obtained by SVM (linear kernel) classifier.

Table 7.2: Stress classification confusion matrix

	stressed (predicted)	relaxed (predicted)
stressed (actual)	133	23
relaxed (actual)	9	60

Table 7.3: Stress classification accuracy (training dataset amount = 50%)

Classifier	Training accuracy	Test accuracy
SVM (linear kernel)	0.64	0.69
SVM (rbf kernel)	0.83	0.83
kNN	0.96	0.86

We will further evaluate the impact of feature number to the stress classification accuracy. Figure 7.10 plots the recognition accuracy with respect to number of features. Note that the recognition accuracy generally improves when the feature number increases. The high recognition accuracy with a few number of features demonstrates that the extracted features are distinctive.

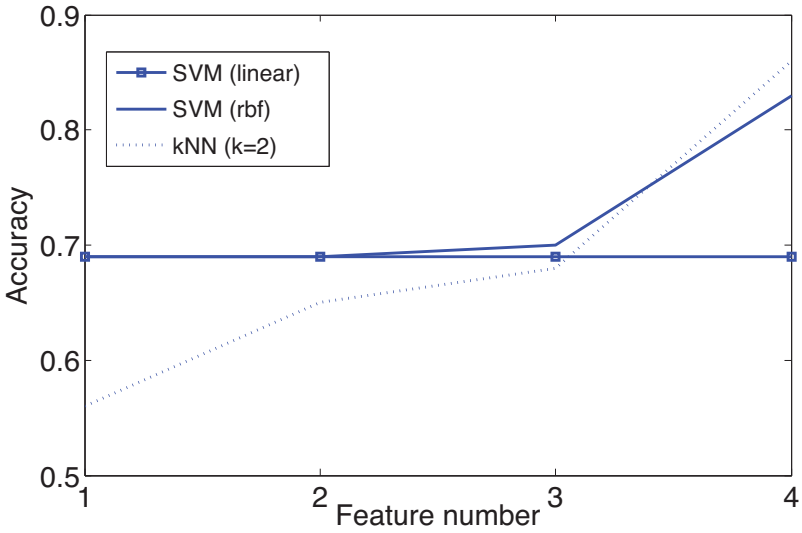


Figure 7.10: Impact of feature number towards classification performance

Finally, we evaluate how the percentage of training dataset impacts the classification performance. From the result shown in Figure 7.11, we can see that the performance of classification model gets better with more training set. Specifically, when the 50% of data is used for training, the classification error of kNN classifier is 0.79, SVM (linear kernel) and SVM (rbf kernel) achieves 0.68 and 0.75 error respectively.

7.5 Discussion

Although some interesting findings have been revealed, we need to consider the limitations of this work.

First, the stress is measured via self-reports in this work. The definition and measurement of stress remains an open challenge. Some of participants say it is difficult to accurately report the stress using 7-point Likert scale. Therefore, the reported stress level may not linearly reflect the real stress intensity. Note that some physiological sensors have been used to quantify the stress level based on skin temperature, electrodermal activities, etc. In order to obtain more accurate stress measurement, our future work will include different physiological sensors.

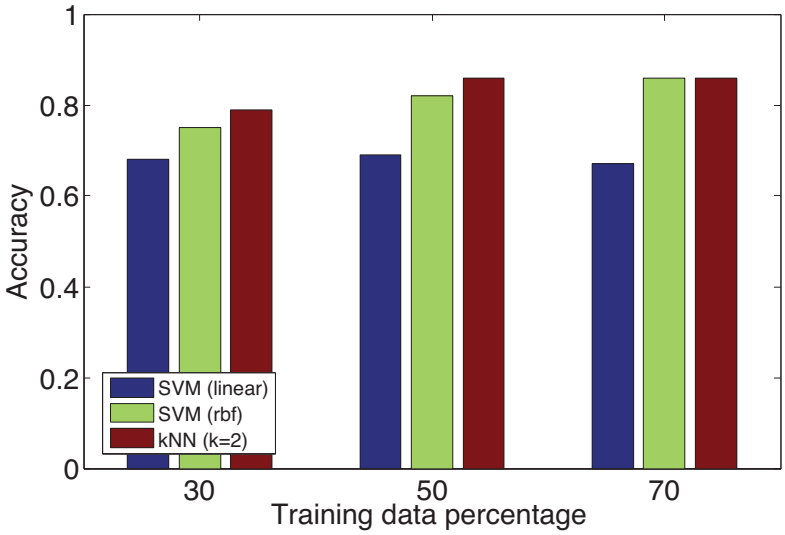


Figure 7.11: Impact of training dataset towards classification performance

The second limitation of this work is only cognitive stress have been induced during the experiment. In addition to cognitive stress, however, people experience other types of stress elicited by physical or psychosocial stressors. Considering other types of stressors can give us more understandings on the association between seating pressure and stress. In the future, we plan to collect more data from participants in real-life scenarios.

After correlation analysis, we find that participants show increased average seating pressure, larger variance among sensor data and increased seating imbalance under the stressful conditions. However, those seating pressure features are not significantly different between stressed and relaxed states. Besides, it is important to understand why those seating pressure features relate to the stress. Therefore, future research will incorporate additional wearable and physiological sensors to monitor both the physical and muscle activities, in an attempt to provide more insights about how stress influences the seating pressure.

Even under the same condition, the stress response of different participants varies. It suggests that the stress detection model should be user-specific. Note that all the participants are male in this work. Thus, our further study will involve female participants

to conduct the experiment.

7.6 Summary

This work presents a stress detection system based on seating pressure signals that are collected from non-intrusive pressure sensors in a seat cushion. The result reveals that during the stressful conditions, participants show increased average seating pressure, larger variance among sensor data and increased seating imbalance. Besides, the proposed stress detection system is able to achieve 86% accuracy using four seating pressure features.

Although the results are preliminary, which are based on limited number of participants and stressors, it reveals that seating pressure distribution is related to stress level. In the future, we will collect larger datasets and introduce some physiological sensors to quantify the stress level. Furthermore, we will incorporate other sensing modalities to improve the stress detection accuracy. We believe that the findings of this work will create new opportunities for designing a less unobtrusive stress monitoring system.

Chapter 8

Conclusions and Future Research

In this chapter, we summarize this thesis in section 8.1 and point out the future research direction in section 8.2.

8.1 Conclusions

Understanding user behaviour opens up a lot of novel opportunities for healthcare, business intelligence and internet of things applications, which is drawing increasing attention from both academy and industry. Specifically, user behaviour understanding encompasses the modeling, recognition and analytics of user behaviour.

This thesis focuses on two research directions: user behaviour modeling and recognition, correlation analytics. The first part focuses on how to accurately model and recognize certain categories of user behaviours based on ambient sensor data. In particular, we study three kinds of behaviours which play critical roles in both physical and psychological heath, including: sitting posture, eating habit and social interaction. Second part aims to identify and exploit the correlation between user behaviours and other user states such as emotion. Specifically, we conduct correlation analytics be-

tween mobility and social circle, stress and sitting posture, and exploit the correlation relationship to build up new recognition model.

First, we focus on the modelling and recognition of three important user behaviours, including: sitting posture, eating habit and social interaction. 1) Current solutions for siting posture recognition, however, are impractical due to intrusiveness, high cost or low generalization accuracy. In this work, we design Postureware, an accurate, low-cost and non-intrusive sitting posture recognition system. In particular, Postureware incorporates very thin pressure sensors to offer non-intrusive experience, an effective sensor placement solution to reduce cost, a set of user-invariant features and an ensemble learning classifier to improve generalization ability. The results show that Postureware can achieve 99.6% ten-fold cross validation accuracy and 84.7% generalization accuracy only with 10 sensors. In addition, we further evaluate the system utility by developing three applications, including unhealthy sitting posture monitoring, sitting posture-based game playing and wheelchair control.

2) We study the problem of monitoring an individual's eating behaviour using off-the-shelf smartwatch and smartphone. However, very few works have been developed for long-term eating behaviour monitoring by means of a noninvasive platform. In particular, we exploit the accelerometer of smartwatch to derive user's eating behaviour, including: eating schedule, food cuisine and food item. Besides, we leverage the collaboration between smartwatch and smartphone to reduce the energy consumption of smartwatch, and thus enabling long-term monitoring. The primary contributions of our work include a context-aware data collection method to conserve energy, a novel set of accelerometer features that are able to capture key characteristics of eating behaviour patterns, and a light-weight decision tree-based classification algorithm. We evaluate our approach using real-world traces and the experimental results demonstrate our work is able to monitor individual's eating behaviour in a non-invasive and energy-efficient manner.

3) Most of the existing work in social activity recognition are based on the patterns of

individual user such as location pattern, vocal pattern, etc. However, we observe that social activity is associated with a community, which inherently exhibits the patterns with respect to multiple users. Thus, by exploiting the information from multiple users, we are able to improve the accuracy of social activity recognition. In this paper, we introduce the concept of social circle, which reveals the behaviour pattern associated with multiple users in social activities. Here, a social circle refers to a set of users frequently gathering to conduct certain social activities. Based on the social circle concept, we present CircleSense, an accurate and efficient smartphone-based system for social activity recognition. The main idea is to derive social activity by integrating both social circle and time information. In particular, social circle is extracted from the social proximity information obtained by Bluetooth device discovery. To further improve the system accuracy, we apply metric learning technique to extract social circle from social proximity information. To evaluate the system performance, we conduct extensive experiment based on the dataset collected in real world from 16 subjects. The experiment result shows that CircleSense outperforms the existing methods in terms of the recognition accuracy.

Second, we study the problem of correlation analytics among behaviours. In particular, we find that there exists correlation between human mobility and social circle, as well as stress and sitting behaviour. Furthermore, by leveraging the correlation relationships, we improve the accuracy of human mobility prediction and stress measurement. To identify the correlation between human mobility and social circle, we conduct correlation analytics on 10-day Wi-Fi traces collected from 111K devices in a large shopping mall. We found that dwell time of repeat visitor exhibits a low degree of variation. Interestingly, visitor dwell time is positively correlated with the size of social group during the visit. By exploiting the above findings, this work presents an accurate user dwell time prediction model that incorporates time and social context, dwell time and leave time history. Evaluation results show that the proposed model is able to provide high accuracy of predicting user dwell time and outperform the baseline methods.

Next, We are interested to explore the possibility of detecting stress by analysing the seating pressure distribution on the chair. In particular, we collect seating pressure data from 15 participants using a seat cushion which is deployed with 20 pressure sensors. Through correlation analysis, we identify a number of seating pressure features that are associated with stress, including: average seating pressure, pressure imbalance, etc. Based on the associated features, we build up a stress detection framework to classify whether participants are stressed or not. The result show that the stress detection framework can achieve 86% accuracy using kNN classifier.

8.2 Future Research

In this section, we highlight some future directions for our research works.

First, we study sitting posture recognition problem based on pressure sensors in Chapter 3. In future work, we plan to extend the sitting posture recognition system to perform inter-disciplinary researches. In particular, we are interested in three directions: affection recognition, rehabilitation and self-improvement. First, we will investigate the correlation relationship among sitting behaviours with human emotion. By leveraging the correlation relationship, we can develop a sitting posture-based prediction model to recognize user emotion. Second direction is rehabilitation, which can help people with chronic spinal cord or traumatic brain injuries, to recover the sitting balance via game-based exercise. The third direction is self-improvement, which can provide feedback for dancers or meditators to maintain a good posture.

Second, we develop an eating behaviour detection system based on smartwatch and smartphone in Chapter 4. Note that this work is based on a small scale of datasets. In particular, we focus on a few kinds of cuisine types: American cuisine, Chinese cuisine and fast food. Moreover, all the dataset is collected inside the restaurant, which means that some home-cooked food are not considered. In the future, we plan to collect a

wider categories of datasets. Specifically, we will include the dataset when people is cooking. The intuition is that cooking different kinds of food will result in discriminative wrist motion patterns. Thus, the system accuracy is expected to increase by incorporating the motion signals from both cooking and eating. Furthermore, we will recruit more individuals for data collection. Besides, due to the limitation of accelerometer sensor, the current system cannot identify what kind of meat people is eating. To enable fine-grained eating behaviour monitoring, we might need to incorporate more information from other modalities. One of the future directions is to integrate google glass into the system. Our preliminary study shows that there exists acceleration patterns of head movement when people are eating. Besides, we can make use of the camera of google glass to take pictures and conduct image processing to obtain more detailed information about the food.

In Chapter 5, we study social activity recognition problem based on smartphone. As an initial study, we demonstrate the effectiveness of exploiting social circle and temporal information to derive different social activities. Although the result is promising, the design and functionalities of the system are premature. In the future, we plan to work on the following directions. First, we will leverage more sensor modalities such as accelerometer and microphone to infer more fine-grained social activities. Second, we will conduct a large-scale dataset to evaluate of the effectiveness of social circle information to distinguish different social activities. Third, we will investigate the mechanism that could adaptively update the members in a social circle, which enables the long term monitoring.

In Chapter 6, we identify the correlation relationship between social group and dwell time and construct a user dwell time prediction model based on social context. To our knowledge, this is first work to identify and evaluate the impact of social context towards user dwell time, and incorporate social context into the dwell time prediction model. In the future, we will extend the proposed framework to predict future user trace inside a shopping mall.

Finally, in Chapter 7, we study the association between seating behaviour and stress, and then build up a stress measure model based on seating pressure features. Although the results are preliminary, which are based on limited number of participants and stressors, it reveals that seating pressure distribution is related to stress level. In the future, we will collect larger datasets and introduce some physiological sensors to quantify the stress level. Furthermore, we will incorporate other sensing modalities to improve the stress detection accuracy. We believe that the findings of this work will create new opportunities for designing a less unobtrusive stress monitoring system.

References

[AKT07] Oliver Amft, Martin Kusserow, and Gerhard Tröster. Probabilistic parsing of dietary activity events. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pages 242–247. Springer, 2007.

[ASLM+10] B. Arnrich, C. Setz, R. La Marca, G. Troster, and U. Ehlert. What does your chair know about your stress level? *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):207–214, March 2010.

[B+06] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[BBT05] Bernard Boulay, François Brémond, and Monique Thonnat. Posture recognition with a 3d human model. In *Imaging for Crime Detection and Prevention, 2005. ICDP 2005. The IEE International Symposium on*, pages 135–138. IET, 2005.

[BEM+13] Marco V. Barbera, Alessandro Epasto, Alessandro Mei, Vasile C. Perta, and Julinda Stefa. Signals from the Crowd: Uncovering Social Relationships Through Smartphone Probes. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 265–276, New York, NY, USA, 2013. ACM.

[BGA+08] Aaron Beach, Mike Gartrell, Sirisha Akkala, Jack Elston, John Kelley, Keisuke Nishimoto, Baishakhi Ray, Sergei Razgulin, Karthik Sundaresan, Bonnie Surendar, et al. Whozthat? evolving an ecosystem for context-aware mobile social networks. *Network, IEEE*, 22(4):50–55, 2008.

[Bit14] Bitbite. https://www.indiegogo.com/projects/bitbite-lose-weight-improve-your-eating-habits, December 2014.

[BWJR87] Phillip J Brantley, Craig D Waggoner, Glenn N Jones, and Neil B Rappaport. A daily stress inventory: Development, reliability, and validity. *Journal of behavioral medicine*, 10(1):61–73, 1987.

[Cam11] Camera program to monitor schoolchildrens' eating habits. http://www.foxnews.com/us/2011/05/11/texas-schools-pictures-worth-1000-calories/, May 2011.

[CK00] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical report, Hanover, NH, USA, 2000.

[CKG97] Sheldon Cohen, Ronald C Kessler, and Lynn Underwood Gordon. *Measuring stress: A guide for health and social scientists*. Oxford University Press, 1997.

[CKM83] Sheldon Cohen, Tom Kamarck, and Robin Mermelstein. A global measure of perceived stress. *Journal of health and social behavior*, pages 385–396, 1983.

[Cla97] Andy Clark. *Being there: Putting brain, body, and world together again*. MIT press, 1997.

[CMC+12] Ningning Cheng, P. Mohapatra, M. Cunche, M.A. Kaafar, R. Boreli, and S. Krishnamurthy. Inferring user relationship from hidden information in

wlans. In *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, pages 1–6, Oct 2012.

[CML11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[DCG07] Sidney DâĂŹMello, Patrick Chipman, and AC Graesser. Posture as a predictor of learnerâĂŹs affective engagement. In *Proceedings of the 29th annual cognitive science society. Cognitive Science Society, Austin, TX.*, 2007.

[DGP12] Trinh Minh Tri Do and Daniel Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 163–172. ACM, 2012.

[DHM09] Yujie Dong, Adam Hoover, and Eric Muth. A device for detecting and counting bites of food taken by a person during eating. In *Bioinformatics and Biomedicine, 2009. BIBM'09. IEEE International Conference on*, pages 265–268. IEEE, 2009.

[DK04] Sally S Dickerson and Margaret E Kemeny. Acute stressors and cortisol responses: a theoretical integration and synthesis of laboratory research. *Psychological bulletin*, 130(3):355, 2004.

[DNG+00] Rod K Dishman, Yoshio Nakamura, Melissa E Garcia, Ray W Thompson, Andrea L Dunn, and Steven N Blair. Heart rate variability, trait anxiety, and perceived stress among physically fit men and women. *International Journal of Psychophysiology*, 37(2):121–133, 2000.

[DSW+14] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. Detecting periods of eating during free-living by tracking wrist motion. *Biomedical and Health Informatics, IEEE Journal of*, 18(4):1253–1260, 2014.

[ELFB00] K.M. Everard, H.W. Lach, E.B. Fisher, and M.C. Baum. Relationship of activity and social support to the functional health of older adults. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 55(4):S208–S212, 2000.

[EP06] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[Euc14] Euclid. Measure. analyze. improve offline experience. `http://euclidanalytics.com/solutions/`, 2014.

[fit14] fitbit. http://www.fitbit.com/, December 2014.

[FMGK12] N. Foubert, A.M. McKee, R.A. Goubran, and F. Knoefel. Lying and sitting posture recognition and transition detection using a pressure sensor array. In *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium on*, pages 1–6, 2012.

[for14] Hapifork. https://www.hapi.com/product/hapifork, December 2014.

[FS96] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm, 1996.

[GBNT04] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection-theory and algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 43. ACM, 2004.

[GGAAB+13] Marta Garaulet, Purificación Gómez-Abellán, Juan J Alburquerque-Béjar, Yu-Chi Lee, Jose M Ordovás, and Frank AJL Scheer. Timing

of food intake predicts weight loss effectiveness. *International journal of obesity*, 37(4):604–611, 2013.

[Gol78] CJ Golden. Stroop colour and word test. *age*, 15:90, 1978.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[GWC+11] Tao Gu, Liang Wang, Hanhua Chen, Xianping Tao, and Jian Lu. Recognizing multiuser activities using wireless body sensor networks. *Mobile Computing, IEEE Transactions on*, 10(11):1618–1631, Nov 2011.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[HMP11] Javier Hernandez, Rob R Morris, and Rosalind W Picard. Call center stress recognition with person-specific models. In *Affective Computing and Intelligent Interaction*, pages 125–134. Springer, 2011.

[HPRC14] Javier Hernandez, Pablo Paredes, Asta Roseway, and Mary Czerwinski. Under pressure: Sensing stress of computer users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 51–60. ACM, 2014.

[KAS14] Haik Kalantarian, Nabil Alshurafa, and Majid Sarrafzadeh. A wearable nutrition monitoring system. In *Wearable and Implantable Body Sensor Networks (BSN), 2014 11th International Conference on*, pages 75–80. IEEE, 2014.

[KB11] John Krumm and AJ Bernheim Brush. Learning time-based presence probabilities. In *Pervasive Computing*, pages 79–96. Springer, 2011.

[KBA+07] James F Knight, Huw W Bristow, Stamatina Anastopoulou, Chris Baber, Anthony Schwirtz, and Theodoros N Arvanitis. Uses of accelerometer data collected from a wearable system. *Personal and Ubiquitous Computing*, 11(2):117–132, 2007.

[KBP07] Ashish Kapoor, Winslow Burleson, and Rosalind W Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, 2007.

[Ken48] Maurice George Kendall. Rank correlation methods. 1948.

[KKNT08] Kazuhiro Kamiya, Mineichi Kudo, Hidetoshi Nonaka, and Jun Toyama. Sitting posture analysis by pressure sensors. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[KLT+14] Azusa Kadomura, Cheng-Yuan Li, Koji Tsukada, Hao-Hua Chu, and Itiro Siio. Persuasive technology to improve eating behavior using a sensor-embedded fork. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 319–329, New York, NY, USA, 2014. ACM.

[KMP01] Ashish Kapoor, Selene Mota, and Rosalind W Picard. Towards a learning companion that recognizes affect. *AAAI Fall symposium*, 2001.

[KMY11] P Karthikeyan, M Murugappan, and S Yaacob. A review on stress inducement stimuli for assessing human stress using physiological signals. In *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*, pages 420–425. IEEE, 2011.

[KP11] P. Klasnja and W. Pratt. Healthcare in the pocket: Mapping the space of mobile-phone health interventions. *Journal of Biomedical Informatics*, 2011.

[KWM11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, March 2011.

[KY13] Yoshiyuki Kawano and Keiji Yanai. Real-time mobile food recognition system. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 1–7. IEEE, 2013.

[LA06] Yue Li and R. Aissaoui. Smart sensor, smart chair, can it predicts your sitting posture? In *Industrial Electronics, 2006 IEEE International Symposium on*, volume 4, pages 2754–2759, 2006.

[LC13] Chengwen Luo and Mun Choon Chan. Socialweaver: Collaborative inference of human conversation networks using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 20:1–20:14, New York, NY, USA, 2013. ACM.

[LCLH14] Guanqing Liang, Jiannong Cao, Xuefeng Liu, and Xu Han. Cushionware: A practical sitting posture-based interaction system. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 591–594. ACM, 2014.

[Lei09] Tom Leighton. Improving performance on the internet. *Communications of the ACM*, 52(2):44–51, 2009.

[LFR+12] Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360. ACM, 2012.

[LLLZ13] Robert LiKamWa, Yunxin Liu, Nicholas D Lane, and Lin Zhong. Moodscope: building a mood sensor from smartphone usage patterns. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 389–402. ACM, 2013.

[LMH+13] Youngki Lee, Chulhong Min, Chanyou Hwang, Jaeung Lee, Inseok Hwang, Younghyun Ju, Chungkuk Yoo, Miri Moon, Uichin Lee, and Junehwa Song. Sociophone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 375–388. ACM, 2013.

[LOIP10] Tom Lovett, Eamonn O'Neill, James Irwin, and David Pollington. The calendar as a sensor: analysis and improvement using data fusion with social networks and location. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 3–12, New York, NY, USA, 2010. ACM.

[Lue94] Rani Lueder. *Hard facts about soft machines: the ergonomics of seating*. CRC, 1994.

[LXL+11] N.D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A.T. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). *UbicompÂąÂŕ11*, pages 355–364, 2011.

[MAST10] Jan Meyer, Bert Arnrich, Johannes Schumm, and G Troster. Design and modeling of a textile pressure sensor for sitting posture classification. *Sensors Journal, IEEE*, 10(8):1391–1398, 2010.

[May09] Mayfieldclinic. Posture for a healthy back. `http://www.mayfieldclinic.com/PDF/PE-Posture.pdf`, 2009.

[MBN02] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 306–313. IEEE, 2002.

[MCY+14] Chao Ma, Jiannong Cao, Lei Yang, Jun Ma, and Yanxiang He. Effective social relationship measurement based on user trajectory analysis. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):39–50, 2014.

[ME12] ABM Musa and Jakob Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 281–294. ACM, 2012.

[MGW+12] Rania A Mekary, Edward Giovannucci, Walter C Willett, Rob M van Dam, and Frank B Hu. Eating patterns and type 2 diabetes risk in men: breakfast omission, eating frequency, and snacking. *The American journal of clinical nutrition*, 95(5):1182–1189, 2012.

[MKF+07] Bilge Mutlu, Andreas Krause, Jodi Forlizzi, Carlos Guestrin, and Jessica Hodgins. Robust, low-cost, non-intrusive sensing and recognition of seated postures. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 149–158. ACM, 2007.

[MLEC07] E. Miluzzo, N. Lane, S. Eisenman, and A. Campbell. Cenceme–injecting sensing presence into social networking applications. *Smart Sensing and Context*, pages 1–28, 2007.

[MLF+08] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, and A.T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme

application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM, 2008.

[MP03] Selene Mota and Rosalind W Picard. Automated posture analysis for detecting learner's interest level. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 5, pages 49–49. IEEE, 2003.

[MSCN13] Justin Manweiler, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. Predicting length of stay at wifi hotspots. In *INFOCOM, 2013 Proceedings IEEE*, pages 3102–3110. IEEE, 2013.

[NB11] H. Nguyen and L. Bai. Cosine similarity metric learning for face verification. *Computer Vision–ACCV 2010*, pages 709–720, 2011.

[PBC+03] M.E. Pollack, L. Brown, D. Colbry, C.E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3):273–282, 2003.

[PGK+09] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity identification using body-mounted sensorsâĂŤa review of classification techniques. *Physiological measurement*, 30(4):R1, 2009.

[Pre07] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[Pre12] Christina Prell. *Social network analysis: History, theory and methodology*. Sage, 2012.

[PWF12] Sebastian Päßler, Matthias Wolff, and Wolf-Joachim Fischer. Food intake monitoring: an acoustical approach to automated food intake ac-

tivity detection and classification of consumed food. *Physiological measurement*, 33(6):1073, 2012.

[QZLS13] Weijun Qin, Jiadi Zhang, Bo Li, and Limin Sun. Discovering human presence activities with smartphones using nonintrusive wi-fi sniffer sensors: The big data prospective. *IJDSN*, 2013, 2013.

[RDV11] S. Rosenthal, A. Dey, and M. Veloso. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. *Pervasive Computing*, pages 170–187, 2011.

[RKL+12] Ioanna Roussaki, Nikos Kalatzis, Nicolas Liampotis, Pavlos Kosmides, Miltiades Anagnostou, Kevin Doolin, Edel Jennings, Yiorgos Bouloudis, and Stavros Xynogalas. Context-awareness in wireless and mobile computing revisited to embrace social networking. *Communications Magazine, IEEE*, 50(6):74–81, 2012.

[SAS+10] Cornelia Setz, Bert Arnrich, Johannes Schumm, Roberto La Marca, G Troster, and Ulrike Ehlert. Discriminating stress from cognitive load using a wearable eda device. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):410–417, 2010.

[sci15] scikit-learn. http://scikit-learn.org/stable/index.html, April 2015.

[SK12] T. Shibata and Y. Kijima. Emotion recognition modeling of sitting postures by using pressure sensors and accelerometers. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1124–1127, 2012.

[SKJH06] Libo Song, David Kotz, Ravi Jain, and Xiaoning He. Evaluating next-cell predictors with extensive wi-fi mobility data. *Mobile Computing, IEEE Transactions on*, 5(12):1633–1649, 2006.

[SLJ+14] Rijurekha Sen, Youngki Lee, Kasthuri Jayarajah, Archan Misra, and Rajesh Krishna Balan. Grumon: Fast and accurate group monitoring for heterogeneous urban spaces. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, pages 46–60, New York, NY, USA, 2014. ACM.

[SMM+11] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive Computing*, pages 152–169. Springer, 2011.

[Soc12] I. C. Society. Ieee standardd 802.11. 2012.

[SP13] Akane Sano and Rosalind W Picard. Stress recognition using wearable sensors and mobile phones. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 671–676. IEEE, 2013.

[SS90] Karl J Sandin and Barry S Smith. The measure of balance in sitting in stroke rehabilitation prognosis. *Stroke*, 21(1):82–86, 1990.

[Stu12] Hex Studio. Life assitance, May 2012.

[Tek] Tekscan. Flexiforce. `http://www.tekscan.com/flexiforce.html`.

[TSP01] Hong Z Tan, Lynne A Slivovsky, and Alex Pentland. A sensing chair using pressure distribution sensors. *Mechatronics, IEEE/ASME Transactions on*, 6(3):261–268, 2001.

[Tur13] Turnstyle. Consumer analytics for the real world. `http://getturnstyle.com/product/analytics`, 2013.

[VDN11] Long Vu, Quang Do, and Klara Nahrstedt. Jyotish: A novel framework for constructing predictive model of people movement from joint

wifi/bluetooth trace. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 54–62. IEEE, 2011.

[vEBNS96] Marleen van Eck, Hans Berkhof, Nancy Nicolson, and Jose Sulon. The effects of perceived stress, traits, mood states, and stressful daily events on salivary cortisol. *Psychosomatic medicine*, 58(5):447–458, 1996.

[Vit07] Jordi Vitrià. Beyond the user: A review of socially aware computing. In *Proceedings of the 2007 conference on Artificial Intelligence Research and Development*, pages 6–8, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.

[VvDdG00] Tanja GM Vrijkotte, Lorenz JP van Doornen, and Eco JC de Geus. Effects of work stress on ambulatory blood pressure, heart rate, and heart rate variability. *Hypertension*, 35(4):880–886, 2000.

[Wei99] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, July 1999.

[WHO15] World health organization. http://www.who.int/mediacentre/factsheets/fs311/en/, January 2015.

[Wik12] Wikipedia. Additive smoothing, Aug 2012.

[Wik13] Wikipedia. Poor posture. `http://en.wikipedia.org/wiki/Poorposture`, July 2013.

[Wik14] Wikipedia. Information gain. `http://en.wikipedia.org/wiki/Information_gain_in_decision_trees`, July 2014.

[XLH+11] Wenyao Xu, Zhinan Li, Ming-Chun Huang, N. Amini, and M. Sarrafzadeh. ecushion: An etextile device for sitting posture monitoring. In

Body Sensor Networks (BSN), 2011 International Conference on, pages 194–199, 2011.

[YPT+11] Xiao Yu, Ang Pan, Lu-An Tang, Zhenhui Li, and Jiawei Han. Geo-friends recommendation in gps-based cyber-physical social network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 361–368. IEEE, 2011.

[ZLC+08] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 312–321. ACM, 2008.

[ZMH+11] Jinfeng Zhuang, Tao Mei, Steven CH Hoi, Xian-Sheng Hua, and Shipeng Li. Modeling social strength in social media community via kernel-based learning. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 113–122. ACM, 2011.

[ZMT03] Manli Zhu, Aleix M Martinez, and Hong Z Tan. Template-based recognition of static sitting postures. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 5, pages 50–50. IEEE, 2003.

[ZTS03] Kelly H Zou, Kemal Tuncali, and Stuart G Silverman. Correlation and simple linear regression 1. *Radiology*, 227(3):617–628, 2003.

[ZY11] V.W. Zheng and Q. Yang. User-dependent aspect model for collaborative activity recognition. *Proceedings of IJCAI'11*, 2011.

[ZZXM09] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.