5-2015

Reality Mining Techniques Applied To Telecommunication Data

Sunil Maharjan
Department of Computing, Institute of Technology Tralee, Kerry, Ireland

Reality Mining Techniques Applied To Telecommunication Data

Sunil Maharjan

Abstract

In this thesis, the application of Artificial Intelligence (A.I.) techniques are examined in order to further develop and improve the field of Reality Mining. Reality Mining refers to the collection of machine sensed environmental data, from which, it is hoped future human social behaviour can be predicted. Large datasets for Reality Mining study have been generated by recording the patterns of human activity and social interactions over an extended period of time. Reality Mining techniques can provide information on important social clues for the better understanding of how humans live their lives. This information can then be used to help in the development of real world applications for example patient diagnosis, better transportation facilities, traffic management and improvements in healthcare systems. The ubiquitous nature of mobile phones, coupled with their ever increasing sophistication, means they have become an ideal tool for collecting such large datasets. Reality Mining techniques can then be applied to find the predictable patterns of human behaviour and identify relationships between individuals.

In this thesis, Machine Learning Algorithms developed within A.I. are introduced as predictive analysis models for Reality Mining. These models are applied on individuals past location patterns to predict its future locations. This thesis shows that these models can predict the future location of an individual once the past locations of that individual are known. These patterns are observed from anonymised datasets collected using mobile phones in Reality Mining experiment. The Reality Mining experiment conducted in 2004 by Massachusetts Institute of Technology (MIT) Media Laboratory represents approximately 500,000 hours of data on user's location, communication, and device usage behaviour. This data was collected over nine months by monitoring the mobile phone usage of one hundred participants. The dataset was then anonymised and made publicly available for academic research purposes. The predictive models are then compared by measuring performance in predicting unseen test data and the time taken to train the model in question.

The location pattern of people may highly vary from one to another depending on their random movement behaviour. It is believed that this affects in predicting location of an individual. Shannon Entropy is used to measure this randomness level. With this knowledge, the affects in accuracy of location prediction by predictive models is experimented.

Reality Mining Techniques Applied To Telecommunications Data

By

Sunil Maharjan

Supervisors:

Dr. Robert Sheehy

Mr. Andrew Shields

Dr. Pat Doody

A Thesis Submitted to Quality and Qualifications Ireland in Fulfilment of the Requirements for the Master of Science Degree

May 2015

Reality Mining Techniques Applied To Telecommunication Data

Sunil Maharjan

Abstract

In this thesis, the application of Artificial Intelligence (A.I.) techniques are examined in order to further develop and improve the field of Reality Mining. Reality Mining refers to the collection of machine sensed environmental data, from which, it is hoped future human social behaviour can be predicted. Large datasets for Reality Mining study have been generated by recording the patterns of human activity and social interactions over an extended period of time. Reality Mining techniques can provide information on important social clues for the better understanding of how humans live their lives. This information can then be used to help in the development of real world applications for example patient diagnosis, better transportation facilities, traffic management and improvements in healthcare systems. The ubiquitous nature of mobile phones, coupled with their ever increasing sophistication, means they have become an ideal tool for collecting such large datasets. Reality Mining techniques can then be applied to find the predictable patterns of human behaviour and identify relationships between individuals.

In this thesis, Machine Learning Algorithms developed within A.I. are introduced as predictive analysis models for Reality Mining. These models are applied on individuals past location patterns to predict its future locations. This thesis shows that these models can predict the future location of an individual once the past locations of that individual are known. These patterns are observed from anonymised datasets collected using mobile phones in Reality Mining experiment. The Reality Mining experiment conducted in 2004 by Massachusetts Institute of Technology (MIT) Media Laboratory represents approximately 500,000 hours of data on user's location, communication, and device usage behaviour. This data was collected over nine months by monitoring the mobile phone usage of one hundred participants. The dataset was then anonymised and made publicly available for academic research purposes. The predictive models are then compared by measuring performance in predicting unseen test data and the time taken to train the model in question.

The location pattern of people may highly vary from one to another depending on their random movement behaviour. It is believed that this affects in predicting location of an individual. Shannon Entropy is used to measure this randomness level. With this knowledge, the affects in accuracy of location prediction by predictive models is experimented.

Acknowledgement

I would like to take this opportunity to acknowledge the help and guidance provided to me by my supervisors, Dr. Pat Doody, Dr. Robert Sheehy and Mr. Andrew Shields. They have provided me with valuable suggestions and assistance to elaborate my knowledge and concepts during this thesis work. I would like to thank all my supervisors for keeping their patience in reviewing my thesis.

I would also like to extend my sincere thanks to my colleagues from IMaR Gateway Technology at Institute of Technology Tralee, especially, Dr. Ultan McCarthy, Mr. Keith Faolain, Mr. Alex Martinez and Mr. Padraic Moriarty for their constant moral support, advice and technical knowledge that has helped me throughout my research.

Finally, I would like to thank my family and friends for their endless support and encouragements to do my best.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation of Project

Understanding human behaviour is crucial for modern science because the dynamics of many social, technological and economical phenomena are driven by individual human actions (Vázquez et al. 2006). Gaining insights into such human behaviour helps in predicting patterns of an individual, modelling the behaviour of large organizations, detecting trends, spotting emerging phenomena and events, mining public opinions and so on (Nikolopoulos, 2013). Armed with this knowledge, an organization, for instance could form an efficient team with an established leader based on communication behaviour and skills rather than simply following organizational hierarchy. This may lead to improved organizational performance and increase productivity when compared to groups who have no guidance and knowledge of working in a team (Pentland 2004).

Traditionally, surveys and other tools were used as instruments for collecting movement patterns of individuals within human society. The process of collecting data using these instruments resulted on biased responses and interviews from its participants (Huberman & Adamic 2003; Roethlisberger & Dickinson 1939). In addition, this process was time consuming in nature and thus leads to infrequent data collection and discontinuation of such process. The mobile phone has been considered as one of the solutions to this (Eagle 2010). This instrument equipped with special hardware capabilities (such as GPS, Bluetooth, and Accelerometer) can eliminate numerous hurdles and difficulties associated with traditional data collection tools (Raento et al. 2009).

The development of such mobile technology has recently motivated research interest in location-driven data analysis. This is an emerging domain in ubiquitous and media computing. It is possible to determine among other things, internet browsing preferences, voice and SMS text patterns (Osman 2011; Church & Oliver 2011), as well as human movements between cellular areas (Song et al. 2010; Williams et al. 2014) from datasets collected by mobile phones. Ubiquitous mobile phones can act as wearable sensors (Lane et al. 2010). It serves as a continuous data mining tool to collect sensor information on location and proximities by logging local cell towers and scanning nearby Bluetooth devices. This enabled the concept of Reality Mining (Eagle & (Sandy) Pentland 2006).

Reality Mining is concerned with the collection and analysis of machine-sensed data pertaining to human social behaviour (Eagle & (Sandy) Pentland 2006). It is the field of techniques used to extract social clues and information from complex social systems by applying machine learning algorithms known from Data Mining to real-time data (Steinbauer & Kotsis 2012). These techniques can be used to reveal significant information on human locations, physical activities, medical diagnosis, and many more.

The main motivation of this thesis is to contribute to the existing area of Reality Mining by utilizing Artificial Intelligence (A.I.) techniques. Machine learning algorithms developed within A.I. can be used to build predictive models based on the past observations to predict future events. These models are built through a learning process where it extracts useful information from past observations to learn the patterns in the dataset. In this thesis, it is proposed that machine learning algorithms such as Naïve Bayes, Classification Trees, and Multilayer Perceptron (MLP) can be adapted and applied to the MIT Reality Mining dataset to discover the patterns in human movement behaviour. Once these patterns have been discovered, it is believed that this analysis can be used to predict the locations of an individual who has participated in the study at any given time based on their historical movement patterns. Such techniques could be used to manage various problems in urban environment, for example, urban automobile traffic congestions and urban planning management.

Around the globe, traffic congestions are extremely serious problem. In the year 2010, Beijing, the capital of country China, suffered from a 62-mile, 9 day traffic congestion (Nerenberg n.d.). Commuters were highly impacted due to such long traffic congestions. One of the example of such impact can be waste of huge amount of productivity hours. Therefore, traffic monitoring projects can be developed using global positioning systems (GPS) data collected from mobile phones to track real-time locations of commuters in order to understand the flow of traffic at different times and in certain areas of the city.

Similarly, data aggregated from location based services like built-in GPS in mobile phones can also be used as powerful tool for urban analysis (Ratti et al. 2006). This data from mobile phone network can be used to estimate population distribution, types of activities in different parts in the city, how areas in the cities are connected, residential and working areas, and commuting patterns.

1.2 Aims of Project

The main aim of this research is to illustrate how to predict the future locations of individuals based on their previous location records. For this, machine learning algorithms are used to model the predictive classifier. These algorithms are applied on the anonymised MIT Reality Mining dataset. One of the goals of this thesis is also to find out which particular predictive model built using machine learning algorithms are successful in achieving high accuracy to predict the future location of individual by learning its previous locations.

This thesis is also keen to discover if the accuracy attained by predictive models on location prediction is affected by the entropy level of an individual. Therefore, entropy levels of individuals are calculated to find out if individuals in the Reality Mining datasets are either high or low entropic characters. In this context, the entropy would represent how unpredictable the location at a given time for that individual is.

1.3 Research Questions

In summary, the research questions this thesis addresses are as follows:

- Can algorithms developed within Artificial Intelligence techniques be adapted to further extend the area of Reality Mining techniques to provide insights into human behaviour and forecast future locations?
    - Machine learning algorithms are applied to mobile phone datasets to find the most suited algorithm for movement predictions.
- Evaluate the performance and accuracy of machine learning algorithms in forecasting future locations?
    - Compare the accuracy of predictions from machine learning algorithms using confusion matrix.
- Can entropy level affect the prediction of an individual’s future locations?
    - Machine learning algorithms are applied to mobile phone datasets of high and low entropic users.

1.4 Thesis Structure

This thesis follows a structured approach in order to answer the research questions as described in Section 1.3. Chapter 2 presents an introduction and background to Reality Mining, the techniques used within Reality Mining and the research currently being undertaken on it. Chapter 3 presents an overview of Artificial Intelligence, and the Machine Learning Algorithms to be used in this thesis.

The methodology used in this thesis is presented in Chapter 4. It starts with details to the pre-processing of the Reality Mining datasets. The modelling of the predictive methods are then described along with the methods to evaluate the best performing algorithm among other machine learning algorithms. The calculation of entropy level of an individual is referred in Section 4.6 of this chapter. Chapter 5 presents an analysis of the results achieved from the experiments detailed in Chapter 4 and discusses the results produced and comparisons among predictive analysis models are described. Chapter 6 presents the conclusion and the recommendations for further research. This chapter also provides summary of the achievements of this research work.

Chapter 2

Reality Mining

2.1 Introduction

Reality Mining is the application of data mining technique applied to mobile phone and sensor data to identify patterns in human social behaviour (Eagle & (Sandy) Pentland 2006). It demonstrates the power of collecting information on communication, location and proximity data from sensory capable machines, such as mobile phones, over an extended time period to find predictable patterns about human behaviour including movement, communication, and proximity to others (Eagle & (Sandy) Pentland 2006). Reality Mining provides new opportunities to develop services with respect to public health care, disease control, automobile traffic congestion, urban transport planning, urban housing management, energy consumption, home and land security, telecommunication risk factors and social networking.

The use of the mobile phone to collect data is far more preferable to the traditional methods. The evolution of modern data collection from the traditional are outlined in Section 2.2.

The amounts of data produced in this manner is very large, it is natural to examine data mining techniques. The evolution of data mining and these techniques are discussed in Section 2.3. The data available can be varied in nature due to the sophistication of modern phones, this is examined in Section 2.4. Section 2.5 will introduce briefly the concept Reality Mining, covering MIT Reality Mining Datasets. The techniques of Reality Mining is discussed in Section 2.6. New emerging trends with Reality Mining techniques are also discussed.

2.2 Evolution of Social Data Gathering

Prior to the advent of internet and technologically driven data collection, social scientists conducted a few studies into social interactions within human groups (Hesse-Biber 2011). According to Davies et al (1941), researchers started collecting data on interaction and relationship between individuals in small groups in 1930's through manual observations and survey tools that continuously took notes on the groups' behaviour. Self-report network data was the traditional method to conduct research on human social network. In this method, data was collected directly from participant's responses using observational experiments. These experiments may have involved interviews and questionnaires that were designed to observe and record the participant's feelings, attitudes, and beliefs and so on. However, this method was time consuming and it led to the research being bounded within a limited number of individuals (Freeman et al. 1987).

With the advent of internet technologies, researchers started collecting large datasets on human social networks from digital information such as email and social media platforms on internet (Barabasi et al. 2002). These human social networks represent the network connections between a large numbers of people and have a variety of interesting properties. However, the traditional method of using human observer to collect the rich interpersonal information such as face to face interactions, emotions etc. had been lost, but this was counter balanced by the fact that the new data was objective, continuous and automatically collected.

Researchers have also attempted to generate both rich and large scale data by offering agent based models to simulate behaviour of people in groups using simple rules. However, this was considered to be oversimplified and has been shown to be flawed in many cases, for example, the models of the gossip dissemination across an organization of agents make the assumption that agents travel with Brownian motion was one of such case to be exposed as totally incorrect (Moreno et al, 2004).

However, the widespread use of mobile phones has launched the world into a new era of wearable computing. Computers are perfect at generating and storing vast amounts of objective data over an extended and continuous period of time, something which was, for all intents and purposes, impossible in practice for tradition social data gathering techniques. Today, mobile phones have been seamlessly integrated into people's day to day life. They are simple tool yet very powerful and interactive technology. Because of its nature, people carry them everywhere with them around the world and this has provided social science researchers with opportunity to capture large scale of data on both individuals and hence group behaviour of people.

With the number of mobile phone accounts set to exceed the world's population in 2014 (Pramis 2013), their ubiquitous nature is undeniable. With the improving technology, mobile phones are becoming increasingly more powerful, making it possible for mobile phones to continuously generate and store or report data. This overcomes the limitation of traditional method such as face to face interactions and can provide an invaluable source of data for social science researcher, urban planning architects etc.

2.3 Evolution of Data Mining

Data mining is an analytical process of discovering interesting and meaningful information such as patterns and relationships from large sets of data (Clifton 2014). The evolution of data mining took place with the increase and rapid development in computerized databases to store business data and provide business analysts with business solutions. It processes the data using sophisticated data search capabilities and statistical algorithms to find patterns and relationships in very large datasets. It combines the disciplines of Artificial Intelligence, Machine Learning and Statistics to achieve its goal of finding hidden patterns within. These patterns, associations or relationships, amongst all this data can provide useful information. For example, let’s assume that health is related to where people go and how they interact with one another. A mobile phone applications can be developed to track this behaviour and social interaction and can subsequently improve health by showing individuals how their behaviour and social interactions could affect their fitness and by offering them discounts on health insurance and on the subscription fees of fitness centres if their mobile phone indicates that they commit to regular physical exercises in fitness centres with their colleagues. This information can then be provided to the individual and an offer can be made where the individual may be interested to opt in health insurance and fitness clubs to improve his health (Dong et al. 2011).

Business Analysts are interested in such information to predict the behaviour of its customers. If an individual feels the benefit improved health and fitness as a result of a service provided, he would not hesitate to pay for such a service. This would benefit fitness clubs with customers’ revenue and also the insurance companies’ gets lower risk on their customers.

Telecommunication industries are, by their very nature, an industry that generates large scale data and could benefit from data mining techniques being applied which could offer useful information on business solutions to retain its existing customers and attract new ones. For example, they could apply data mining techniques to enhance the proper management of their customers by estimating the likelihood of a customer moving to another provider, or if the data was available, what customer, or demographic of customer, would be productive in targeting as new customers. By predicting these, they can adopt a set of strategic and tactical retention measures to reduce the number of lost customers and maximise the number of new ones (Larivière & Van den Poel 2004).

The continuing trend and development in both the data mining technology and telecommunication systems enabled the concept of Reality Mining. Reality Mining is defined as a collection of data mining techniques or tools that mine information from the wireless sensor data to identify predictable patterns in human behaviour. Sensors such as mobile phones have high level processing units and a variety of built in sensor systems that can collect large-scale of datasets on human behaviour in real time.

2.4 Mobile Phones as Wearable Sensors

The popularity and pervasiveness of mobile phones cannot be understated. The use of mobile phones has extended beyond a simple communication device. Most modern mobile phones now come with Global Positioning System (GPS) enabled services, built-in services and integrated sensors such as Bluetooth, Infrared, Wi-Fi, accelerometer, cameras and various applications which make them capable of continuously capturing data, allowing the collection of large datasets on human and social activity. These datasets captured using mobile phones can reveal significant clues about human behaviour. These behavioural clues revealed from such mobile phone data on human behaviour could provide patterns relating to people's regional movements as well as the shape and dynamics of their social networks (Eagle & (Sandy) Pentland 2006).

Whenever a mobile phone user makes a call, it pings the nearest cell tower revealing its location. Built-in GPS enabled services and location based services in mobile phone can track the location at a given time, providing information about the user's whereabouts and the distance the phone has travelled. These patterns on user's location can be used to quantify the pace of life in different cultures, epidemiological models, and various vertical applications such as CitySense (Networks 2013). CitySense is a real time mobile application for nightlife discovery and social navigation. It shows the overall activity level of a city, top activity hotspots and places with unexpected high activity in real time.

Built-in Bluetooth is another important feature in mobile phones that gathers wireless proximity data. This proximity data can be combined with location, time and date to infer the nature of relationships between individuals (Eagle & Pentland 2005). In addition, mobile phone with an accelerometer measures the body movement of their users that could tell if a user is sitting or walking. Also all phones have built-in microphones that can be used to analyse the tone of voice, the time that user talked with someone and how often one interrupt others. These patterns can tell what roles people play in groups.

According to (Eagle 2010) mobile phones are ideally suited to provide insights into human social behaviours as these phones can collect data that includes continuous recordings of location, proximity and communication patterns.

2.5 Background of Reality Mining

Reality Mining is one aspect of digital footprint analysis that can be defined as the collection and analysis of machine sensed data pertaining to human social behaviour with the goal of modelling behavioural patterns (Eagle & (Sandy) Pentland 2006). One of the key sensors employed by Reality Mining techniques is the mobile phone that has revolutionized the human life with a digital world in recent years.

In 1982 there were 4.6 billion people living in the world, and not a single person was subscribed to mobile phones. Today, there are seven billion people in the world and over six billion of them are subscribed with mobile phones, providing mobile access for more than three-quarters (75%) of the world's population, according to a report by the World Bank (DeGusta 2012). The Geneva-based United Nations telecommunications agency (International Telecommunication Union) further claims that the number of mobile phones worldwide is set to catch up to the world’s population by 2014.

Today, humans use their mobile phone as not to only communicate with each other but also to do all their day to day work such as checking emails, marketing using social sites and uploading videos to YouTube. The increasing number of sensors embedded in this mobile phone captures a lot of contextual information about users including location as well as their social activity. So, it has now become much easier to study human behaviour with this widespread use of mobile phones.

The research group from Massachusetts Institute of Technology (MIT) Media Lab took advantage of this equipment and conducted a study with mobile phones data collected from 100 volunteers during the period of one academic year. This data mining process was later termed as Reality Mining by the pioneers, Nathan Eagle and Alex Pentland. As described in Section 2.3 of this chapter, this study is related to the field of data mining that uses large datasets collected by wireless mobile phone to find hidden patterns in real life which gives a series of important clues and keys to the way humans live their lives. The research group were able to determine which students were friends based solely on mobile phone location records in one of their experiments conducted at MIT Laboratory. They also revealed that data collected from mobile phones could shed light on workplace dynamics as well as on the well-being of whole communities. By logging and time stamping user activity, location and proximity to others, it has been proved possible to measure the underlying dynamics of groups of people working in an organization (Eagle, Pentland, et al. 2009).

2.5.1 History of Reality Mining

Pioneers Nathan Eagle and Alex Pentland along with MIT research group conducted their original Reality Mining Experiment in MIT Media Laboratory within the year 2004-2005. The aim of which was to collect massive amounts of continuous human behavioural data by using the built-in technologies of the mobile phones available at that time. The experiment gathered information on continuous proximity, location, communication and activity data from the phones of one hundred human subjects at MIT Media Lab. According to (Eagle & (Sandy) Pentland 2006), this experiment was the first to collect such a huge amount of mobile phone datasets on rich personal behaviour and interpersonal interactions to study the dynamics of both individual and group behaviour. This was because mobile phones were not powerful enough to track people before this experiment.

The main objective of this experiment was to explore the uses of the mobile phones that enable researchers studying social science to investigate human interactions beyond the traditional survey based methodology as described in Section 2.2 of this chapter. The dataset used for this experiment includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage and phone status which will be briefly discussed in Section 2.5.2, The Reality Mining Datasets. This dataset was analysed to reveal regular and predictable rules and structure in behaviour of individuals, dyads (a tribe in Reality Mining speak), teams and organizations. Eagle and his team developed discriminative and generative probabilistic graphical models as well as models based on Eigen decomposition by using advanced machine learning methods and statistical analysis. These were used to classify and predict an individual's behaviour, relationship with others and association with particular groups. The MIT Reality Mining Dataset has been anonymised and made available to general academic community for research and experimental purposes. As a result this data now has been analysed by many researchers around the world in different ways, giving rise to numerous studies and publications on human behaviour.

Reality Mining was identified as one of top 10 emerging technologies that could change the world in the year 2008 by MIT Technology Review and recently has been listed as “Big Data from Cheap phones” as one of the Top 10 Breakthrough Technologies of 2013 (Talbot 2013).

2.5.2 The Reality Mining Data Sets

The Reality Mining Experiment consists of one hundred human subjects. Seventy five of these subjects were either students or faculty staffs in MIT Media Lab and other twenty five of these subjects were incoming students at MIT Sloan Business School. Among those seventy five subjects at MIT Media Lab, twenty were incoming master’s students and five were incoming MIT freshman. All of these subjects volunteered to become a part of this historic experiment in exchange for the use of a high end smart phone, the Nokia 6600, that was pre-installed with several pieces of software and including context applications such as ContextLogger, ContextContacts, and ContextMedia, written by MIT and University of Helsinki that continuously tracks information on location from cell tower IDs, call logs, text messages, proximate Bluetooth devices, application usage, and phone status (such as charging and idle) (Raento et al. 2005).

Data was collected over the course of nine months which represents approximately 500,000 hours of data on subject’s location, communication and device usage behaviour. In this data collection, each cell tower logged by the subject’s mobile phone were assigned with a unique ID to determine its location. Subjects are prompted to give a label to these cell tower id whenever a new cell tower is detected. This enabled some towers to be labelled with a specific meaning for that particular subject, typically "Work" or "Home” etc. However, establishing a connection with a cell tower can only be expected to pin point the individual location, or the individuals’ phones location to be more precise, to within 100-200m in urban areas. This inaccuracy, especially in the urban context, gives rise to many problems making it hard to accurately infer the exact location point. For this reason, (Eagle & Pentland 2005) introduced a system for sensing complex social systems that revealed the ability of in-built Bluetooth enabled mobile phones to collect information and use in different contexts, recognize social patterns in daily user activity, infer relationships, identify socially significant locations and model organizational rhythms. Mobile phones carry two different radio frequency (RF) networks that can be used for inferring location and human activities. These two different radio frequencies can be classified as short range RF network and long range RF network. Bluetooth is a short range RF network that is capable of discovering devices that allows them to collect information on other Bluetooth enabled devices within 5-10 meters. These information include Bluetooth MAC address (BTID), device name and device type. BTID is a hex number unique to particular device. (Eagle & Pentland 2005) designed a MIDP2 application called BlueAware for Bluetooth phones that runs passively in the background and scans the nearby environment once every 5 minutes logging encountered BTIDs in proximity. Bluedar is a variation to BlueAware that was developed to continuously scan for visible devices discovered up to range of 25 meters and wirelessly transmit the detected BTIDs to the server over a wireless network (802.11b). GSM is a long range RF network that connects to cell tower making it possible for two mobile phones to communicate.

Data was collected from the phones using two methods. Approximately thirty of the subjects were provided data plans (GPRS) on their mobile phone. For these subjects, the phone gets directly connected to the data server during the night and uploads all new data logged during the previous day. For the remaining subjects in the experiment, data were stored on each phone’s internal 32MB memory card. These cards can store approximately four months of behavioural data before they need to be collected by the researchers. However, they had also collected self-report relational data from each subject that includes information about their proximity to and friendship with others.

The concept of continuously gathering large data on subject’s day-to-day behaviour for a long period of time has significant privacy implications. For this, subjects were detailed about the type of information to be collected and provided with instructions on how to temporarily disable the context logging application. In addition, phone numbers in the datasets were one way hashed (MD5), generating unique ids for the analysis (Eagle & (Sandy) Pentland 2006).

2.6 Reality Mining Techniques

In this section, predictable patterns inferred by applying Reality Mining techniques to mobile phone data are discussed. These techniques of Reality Mining are categorized according to the criteria of: recognizing human activity and inferring location and patterns in human movements.

2.6.1 Recognizing Human Activities

With the increase in the use of wearable sensors such as mobile phones, smart watch, armbands, and Google Glass, activity recognition has been the increasing area of research work for researchers focusing on the context of user's daily activities, location and environment. Wearable sensors allow continuous recording of activities across different locations and independent from external infrastructure. There are many possible applications for activity recognition with wearable sensors in the areas such as health care, elderly care, personal fitness, entertainment or performing arts. For instance, using motion sensors, an individual who is suffering from balance disorder can be monitored while at gym or clinical centre during the process of balance training (Bonato 2010).

To recognize the human daily activities, different kind of built-in sensors found in mobile phones, smart watch or standalone sensors can be used. Typically, one could expect modern phones to contain accelerometers, GPS-enabled tracking devices, RFID, Bluetooth, camera, microphone and other physiological sensors. Mobile phones are one of the powerful devices that consists of multiple sensors that provides sufficient sensor data, making it the ideal device to calculate and enable physical activity recognition. The data collected from all these sensors can be integrated with novel data mining and machine learning techniques to model a wide range of human activities.

In this section, different research studies, using these sensors to recognize the human activities, are reviewed. The motive to do this is to provide brief descriptions on existing Reality Mining techniques used on various types of wearable sensors. However, this project is mainly focused to particular movement and positioning sensors used for detecting the movement and location patterns as outlined in Section 2.6.2.

One of the main sensors used for recognizing the human movement activities is an accelerometer. An accelerometer is a common type of wearable sensor that is fairly small, cheap and consumes little energy, memory and processing power and doesn't react to environmental conditions making it relatively good in recognition of physical activities. These activities may include low and high level activities. Low level activities can be considered to be the movement activities typically lasting between couple of seconds to several minutes such as an individual walking or running, shaking hands with others or eating or doing house hold activities like making tea, dusting, and so on. On the other hand, High level activities are long term movement activities that may last more than several minutes to several hours such as cleaning the house. However, high-level activities can be very complex, time consuming and tedious (Bonato 2010).

In 2003, Choudhury and Pentland developed a wearable database acquisition board called “Sociometer” (Choudhury & Pentland 2002) to automatically and unobtrusively learn the social network structures and dynamics of communication networks within human groups. A sociometer is a wearable device that consists of an accelerometer, a microphone and an infrared sensor that is used to capture face to face interactions within the groups and detect the proximity with others. They used microphones in sociometer in order to record the audio information when the users are having conversations. The users carried this device for six hours a day for ten days while they are on MIT campus, capturing about sixty hours of data per subject. They applied the statistical pattern recognition techniques such as dynamic Bayesian network models on this data to build computational models that can learn the structure of face-to face interactions within group interactions.

Similarly, (Huynh 2007) investigated how well the activity recognition approaches can be applied to the recognition of longer and high-level activities. They collected a realistic ten hours’ worth of data using three wearable sensors which were attached to the individual’s right wrist, right hand side of the hip and right thigh. They analysed the dataset using four different machine learning algorithms (K-means, Support Vector Machines (SVM), Hidden Markov Model (HMM) and Neural Networks) to recognize both the low and high level activities contained in the dataset. Their results suggested that the recognition of high level activities can be achieved with the same approach taken for the recognition of low level activities. For all their approaches, they used simple mean and variance features derived from the accelerometer reading at 2 Hz.

However, accelerometer sensors cannot recognize the stable positions or emotional activities of human, but can be classified using additional various sensors such as GPS, camera and physiological sensor. In the paper published by (Yang & Cho 2008), explained how multiple sensor signals can be used to recognize the human activities. In this paper, they showed that applying algorithms like naïve Bayes and fuzzy Bayesian network on continuous sensor data collected using an armband that includes both the accelerometer and physiological sensors is more efficient in achieving good accuracy of 74.4% to recognize the human activities.

There are different types of sensor based activity recognition. One of the popular sensor based activity recognition is the location based activity recognition using GPS-enabled devices. GPS-enabled devices are becoming popular at obtaining location information of human daily life. It typically learns the commuting routines from user locations via GPS traces. Given the data collected from GPS-enabled devices, the user's locations can be tracked by which significant locations can be inferred such as home, workplace, friend's house, gym, stores and restaurants. The information inferred can be used to provide location based information services like searching nearby restaurants or the user's mode of transportation preferences such as foot, car, or bus.

2.6.2 Inferring location and patterns in human movements

In Reality Mining project, the main significant work carried out is the activity recognition systems using mobile phones to determine the user's activity in terms of location. (Eagle & (Sandy) Pentland 2006) used two types of information to determine the user's location. In their work, they initially estimated the cell tower probability functions to achieve relatively high location accuracy. They showed that a mobile may connect to different cell towers consecutively if it stays at one place for a long period of time depending on the network traffic and signal strength of mobile phone. The distribution of these associated cell towers at some static location can be generated if a user spends an adequate time in one particular area. However, this distribution on probability of cell towers may fluctuate with even small changes in location. So, the use of static Bluetooth device as an additional indicator of location is incorporated in order to show that it provides a significant improvement in user localization, especially within office environments.

The following Figure 1 (Eagle & (Sandy) Pentland 2006) from the Reality Mining Experiment shows the cell towers seen for a given area within a 10m radius. To make sure that users locations are within 10m distance, towers are included in the distribution only if they are also detected by common area's static Bluetooth desktop computers.

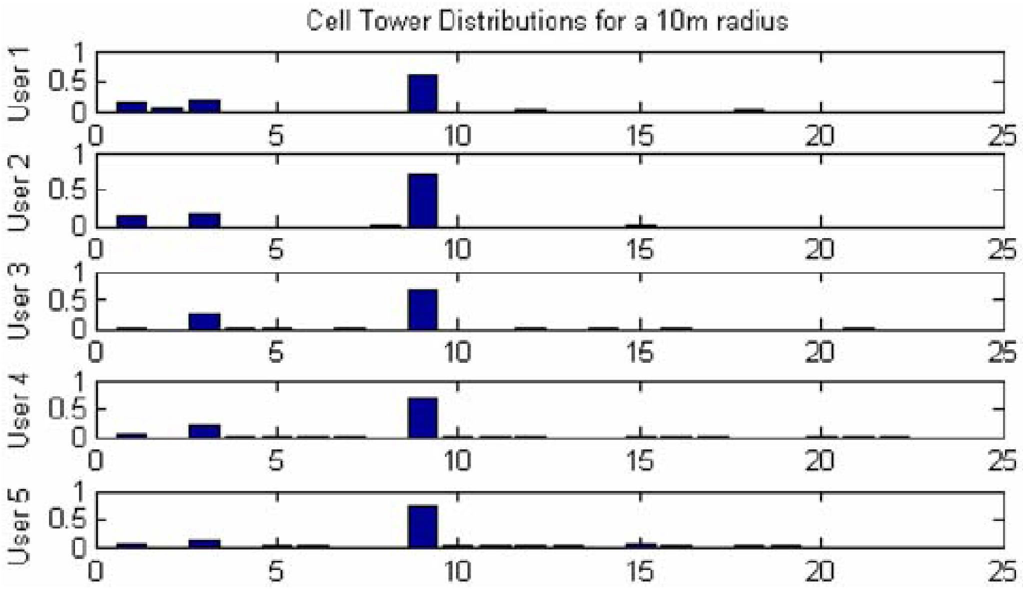


Figure 1: Cell Tower Probability Density Functions (Eagle & (Sandy) Pentland 2006).

In above Figure 1 (Eagle & (Sandy) Pentland 2006), the probability distribution of twenty five cell towers detected by five subjects from MIT experiment who work on the third floor of the same office building is plotted. On the y-axis, the probability of phone's logging to each cell tower listed on x-axis is shown. It can be seen that depending on the location of users, distribution of cell tower varies from one another except the users 4 and 5 who are officemates and have the same distribution of cell towers. In the Reality Mining study, subjects were found without mobile phone reception for 6% of time on average and during these times, they were found to be within range of static Bluetooth desktop device for 21% or another mobile phone for 29% of their time. This shows that the use of static Bluetooth device ID has successively played an important role in accurately estimating the user's location as through cell towers.

In the Reality Mining Experiment, Eagle and Pentland build a simple model using three states: Home, Work and Elsewhere for modelling user's location. The data is obtained from Bluetooth, cell tower and temporal information collected from mobile phones to classify these states. All detected cell towers from mobile phones were assigned with unique cell tower IDs so that it can be used for determining the location of a user. They used a Hidden Markov Model (HMM) conditioned on both the hour of the day and the day of the week (weekday or weekend) to capture daily and weekly routines in user behaviour. HMM (Rabiner & Juang 1986) is a statistical model in which the system being modelled is assumed to be a markov process whose states are hidden, however the output is observed and probabilistically dependant on its state. According to this assumption, the activity of a user at a given time frame only depends on his activity at previous time frame and any longer range dependency is ignored. Eagle and Pentland reported an accuracy typically greater than 95% after the training the model on one month of data from several subjects of Reality Mining Project (Choujaa & Dulay 2009).

In (Eagle, Quinn, et al. 2009), Eagle et al. repurposed the unsupervised clustering to quantify community structure within graphs to identify salient locations within a cellular tower network. They validated these clustering techniques using data from Bluetooth beacons placed in the homes of 215 randomly sampled subjects in major urban city of U.S. They used these clusters as states within several dynamic Bayesian networks (DBNs) to predict dwell times within locations and each subject's subsequent movements with over 90% accuracy. Dynamic Bayesian networks are mainly used for quantifying and predicting human behaviour. They use location co-ordinates such as data from GPS enabled devices, which are much more precise than cell tower data. Usually DBNs are trained on general human movement such as transportation routes (Liao et al. 2007). A dynamic Bayesian network is a Bayesian network that models a dynamic system by representing its state as subsequent time slices. Arcs joining nodes at consecutive time slices encode probabilistic temporal relations between them. A Bayesian network is a probabilistic graphical model where each node in graph represent a random variable and edges between the nodes represent probabilistic dependencies between the random variables. The graphical structures of Bayesian network are therefore used to represent knowledge about the classification task (Ben-Gal 2008).

Using the Reality Mining dataset, (Xiong et al. 2012) introduced the collective behavioural patterns (CBP) to predict the user's location in next 6 hours from the current locations of other users. The specific locations of users were identified from cell tower IDs. In this paper, the collective behavioural patterns is referred to the association patterns among the locations of mobile phone users. The observation was made on thousands of cases of the collective behaviours through mining the association patterns from user mobility data of Reality Mining dataset. They built a Bayesian prediction schema based on the collective behavioural patterns to learn the correlations from the current locations of users to the locations of target users in next few hours. The Markov based predictor was later integrated with this schema to build a hybrid predictor that produced the higher prediction accuracy than single Markov based predictor to forecast the individual's mobility patterns.

From the Reality Mining dataset, (Farrahi & Gatica-Perez 2008) considered a subset of 30 individuals and 121 consecutive days with the goal to analyse proximity and location driven routines from the day in the life of a person. They developed a framework using a Support Vector Machine with a Gaussian Kernel that aims to classify a user's day as weekday or weekend and whether this user is a business or engineering student. In their methodology, they computed both the location and proximity features at two different time scales. One of the time scale is a fine grained one every 30 minutes and another being a course grained one every 3-4 hours. The features were tested alone and in pairs of one location and one proximity feature. The evaluation was performed in leave-one-user-out cross validation where testing was performed on all the days for an unseen user while training on the data for all the people. Their results showed that weekdays were best recognized by detecting the cell towers whereas weekends were best identified from the proximity data. Overall, combining location and proximity features yielded to over 80% accuracy, outperforming both location and proximity features alone. Affiliations (i.e. whether the subject was Media Lab Graduate Student or First Year Undergraduate Student or Staff etc.) were best recognized from the identity of the people in proximity of the user. The generic locations considered (Home, Work and Elsewhere) were found to be little informative for this purpose. Affiliations were correctly classified with nearly 90% accuracy. However given that only 6 out of 23 students considered were business students, a most frequent baseline already provides 74% accuracy.

(Azam et al. 2012) showed that an unusual and abnormal behavioural patterns can be detected from cell tower ID and real time Bluetooth proximity data using Neural Networks and Decision Trees. In this work, the authors used the Reality Mining dataset to infer the behaviour of user with low entropy level using contextual data collected from mobile phones such as cell tower ID and Bluetooth proximity data. A low entropy user is a subject in Reality Mining experiment whose behavioural routines are regular and consistent. They created a framework of four different levels to analyse this data for behaviour detection. In first step, they classified the contextual data into different locations to obtain the movement patterns of the users. In the second and third step, they generated a probability matrix from the hour of the day and day of the week. This matrix was utilized to create the training data. In fourth step, this training data is used in the Multilayer Perceptron Neural Network and Decision Trees for the detection of abnormality in low entropy level user behaviour. In this training data there are three inputs (location, hour, and day of the week) and one output (abnormal behavioural level). The behaviour of a user was categorized in four different levels depending upon the probability of being at any of the four places (Home, Work, Elsewhere, and No Signal) on a specific day and hour. They trained the decision engines with 70% of this training data and remaining was used for cross-validation and testing purposes. They found that neural networks are more accurate as compared to the Decision Tress in behavioural detection. However, Decision Trees seem to require less data and less time consuming.

In their results, they have shown that cell tower ID data gives behaviour of the user on high level scale for example movement patterns in GSM cells that does not help to identify any lower level activities such as attending the lecture, travelling in the bus, whereas Bluetooth data gives more information about the lower level activities depending on the social relations and close proximity of the users.

For predicting user movement, some general knowledge of the location and mobility behaviour of a user is required. The system should be built in such a way that it collects movement behaviour for certain time to acquire information about user's movement. (Vukovic et al. 2009) presented an idea in their paper to track a user for certain period of time for recognizing regular movement patterns that provides a basis for user movement prediction. Many of the research conducted in the area of mobility management such as (Quintero 2005; Kausik Majumdar 2005) shows that there is a certain degree of regularity in the user's movement. For instance, most of the users work in an office during the day hour and stay at home during the night. It is suggested that if user with such behaviour are tracked for certain period of time, the system would be able to recognize the regular patterns in user's movement such as “at home”, “at work”, “going home from work” etc. However there is a high chance that a user may not show a regularity in their movement behaviour, for example, a salesman travelling to different places would be hard for movement predictions.

Predicting user movements can be done in several ways. Research (Ashbrook & Starner 2002; Kausik Majumdar 2005; Lovrek & Sinkovic 2004) done in area of user mobility are using probability models that predicts the user location with a certain probability, while others are using learning approach based on Neural Networks (Vukovic et al. 2007; Quintero 2005; Kausik Majumdar 2005).

In the paper published from (Vukovic et al. 2009), they have proposed an adaptive model for user movement prediction based on statistical movement analysis. They have distinguished specific phases in this paper. They are: collecting information about user movement, analysing and learning user movement patterns and applying movement knowledge to prediction.

Similarly, in the work of (Hill et al. 2010) showed that individual movement models can be built by logging time-stamped cell tower locations. In this paper, they used two sets of data for the month of October from Reality Mining dataset and Kenyan location dataset. Their goal was to predict whether the user was at Home, Work or Elsewhere at a given time. They built a Decision Tree model to predict this user location conditioned on the hour of the day and day of the week. In addition to this, they used Shannon entropy level as well to quantify and compare movement behaviour.

The paper published by (Gambs et al. 2012) has also shown that the next location of an individual can be predicted based on the observations of its mobility behaviour over some period of time and the past locations that an individual has visited. For this experiment, they have used a mobility model called Mobility Markov Chain (MMC) to incorporate *n* number of previously visited locations and developed a novel algorithm for next location prediction based on the mobility model that they coined as n-MMC.

An n-MMC is a MMC in which the states do not correspond only to single point of interest but rather represent the sequence of the n previous visited point of interests (POIs). A Mobility Markov Chain (MMC) models the mobility behaviour of an individual as a discrete stochastic process in which the probability of moving to a state (.i.e. POIs) depends only on the previous visited state and the probability distribution of the transitions between states. POI is point of interest that is discovered using a variant of k-means clustering algorithm on the individual's mobility traces.

In order to predict the next location based on the n last positions in the n-MMC model, they computed a modified form of the transition matrix whose rows represent the n last visited positions. Experiments on the three different dataset shows that the accuracy of their prediction algorithm ranges from 70% to 95%. However, the accuracy of the prediction grows with n, choosing $n > 2$ doesn't seem to bring an important improvement to the cost of a significant overhead in terms of the mobility model.

2.7 Summary

This chapter has outlined the background, history, and operation of Reality Mining techniques. The widespread use of mobile phones by humans has provided huge opportunity to closely study patterns of human behaviour. This study of behaviour has been described in two main categories in this chapter as: Recognizing human activities, and, Inferring location and patterns in human movements.

Most of the studies have shown that Reality Mining techniques are used to measure the patterns of daily human interactions. By applying machine learning algorithms of Artificial Intelligence techniques to information collected using mobile phones, Reality Mining infers the social relationships and behaviour of an individual. With this knowledge gained, in this thesis, these techniques are considered to be used on location data, gathered via the mobile phones of individuals, to extract the hidden movement patterns of their past locations so that its future location can be predicted at a given time.

It is shown that the significance of these techniques can be implemented in large scale, in various fields, for data analysis purposes. For example, Reality Mining techniques can be used in anti-terror tools by tracking terrorists from the unusual patterns of movement and communications learned from their mobile phone network data. Similarly, it can also be used to track traffic congestion problems by analysing real time GPS-enabled mobile phone data and facilitate other road users with alternative routes. Moreover, it may have applications in Search and Rescue in the development of a set of digital footprints of tourists using mobile phones.

These techniques may bring huge opportunities to develop various applications for society, but they also have limitations. These limitations include, for example, the concept of gathering large datasets on human behaviour in real-time in legal manner. Other limitations include the significant pre-processing of collected datasets for cleaning the noise present within the data, and, the multimodal aspects of human behaviour. In the next chapter, Artificial Intelligence techniques to be applied on location data, will be introduced.

Chapter 3

Artificial Intelligence

3.1 Introduction

Artificial Intelligence (A.I.) is the field of study in computing science that defines the abilities of making machines to learn, work, and solve problems in an intelligent, faster, and better manner than humans do. Since 1950s, scientists and technologists around the world are conducting range of research on A.I. to build systems that behave and think as of human intelligence (Harris 2011). The recent development of such A.I. research based machines can be Google's Self Driving Car (Urmson 2014), Smartphone's Speech and Text Recognition (Eaton 2013) and Defence Advanced Research Projects Agency's (DARPA) Autonomous Robots (Davis 2014). The main goal of A.I. is to make machines do things that would require intelligence if done by humans (Negnevitsky 2005). To make an intelligent machine, it requires computers to act without being explicitly programmed. Such field of study is referred as Machine Learning in computer science.

Machine Learning is an essential and vital part of A.I. techniques. It involves the study and development of computational models of learning processes. The goal of machine learning is to build computers applications that are capable of improving their performance with practice and of acquiring knowledge on their own (Michalski et al. 1986). In this modern age, machine learning is one of the most exciting recent technologies within A.I. People probably use machine learning algorithms dozens of times a day without even knowing it. For instance, whenever people use search engines such as Google, learning algorithms like Page Rank algorithm (Page et al. 1999) is executed behind the scenes to make millions of searches efficient and satisfying for the users.

In this chapter, some of the specific machine learning algorithms developed within A.I. are discussed. These machine learning algorithms are used to build the predictive models for our thesis as discussed in Chapter 1. In Section 3.2, A.I techniques are introduced with detail description on its background and machine learning as its vital part of the study. In Section 3.3, basic concepts of machine learning algorithms and its use in data mining problems is discussed. The chapter then presents various machine learning algorithms used in this thesis.

3.2 Artificial Intelligence Techniques

Artificial Intelligence is one of the emerging technologies in the field of computer and information science. It is concerned with the development of hardware and software systems that can store knowledge and effectively use this knowledge to solve problems and accomplish tasks. This field of study includes the research and development of machines such as robots, autonomous car and airplanes and smart military weapons.

- In the summer of 1956, John McCarthy and his colleagues (Russell & Norvig 2010) organized a two month workshop at Dartmouth College where they discussed the field of intelligence which was later termed as Artificial Intelligence for this field of study. Altogether, there were 10 attendees who later became the leaders of A.I. research for next 20 years. They and their students and colleagues initiated to develop A.I. systems that can solve problems with intelligence. When a system learns how to perform a task with high accuracy and without any human interference, it can be said that the system is intelligent. Two of the main possible sources of knowledge to make machines intelligent are: Human knowledge that has been converted into a format suitable for use by an A.I. system.
- Knowledge generated by A.I. system, perhaps by gathering data and information and by analysing data, information and knowledge as its disposal.

A.I. programs provide solutions to numerous complex problems. However, these solutions are always based on mathematics and computer science. In both Mathematical and Computer Science such solutions are known as ‘algorithms’. An algorithm is a usually, but not exclusively, deterministic sequence of instructions that is to be followed to enable the solving of a problem. It is applied to an input to obtain an output. For instance, if an input is a set of numbers then output of it is ordered list where algorithm is sorting the numbers. However, an algorithm requires knowledge in addition to work on complex tasks such as classifying spam emails. This knowledge can be gained by learning examples of spam emails. This ability to learning denotes the intelligence of a machine (Alpaydin 2010).

3.2 Machine Learning Algorithms

Machine Learning algorithms are one of the branches within the A.I. field. Most of its concepts are originated from a careful investigation of human learning. Hence, the development process of machine learning has also contributed in the understanding of human learning (Schank 1995). In 1959, Arthur Samuel defined Machine Learning as "the field of study that gives computers the ability to learn without being explicitly programmed". Supporting this statement, Tom Mitchell in 1997 also defined it as "computer programs that improves its performance at some task through experience" (Mitchell 1997).

In general, a process where knowledge can be gained through experience is termed as Learning. For humans, learning begins on the day that they are born and this process continues all the way through their life. During this process, they gather knowledge and experience which ultimately helps in the improvement of what already has been learnt. Equally, to make computers learn like humans, machine learning algorithms have been adopted as an approach to build intelligent machines. Machine learning is a field of study that is concerned with the development, understanding and evaluation of algorithms and techniques to allow a computer to learn. Computer systems with learning algorithms automatically learn programs from data. Algorithms read and analyse the provided data to build models. This is why the study of statistics would be considered one of the most important parts of machine learning (Polumetla 2006).

As discussed in Chapter 2, large datasets can be collected through different sensors such as mobile phones. The collected dataset holds large amount of behavioural characteristics of human being. However, any dataset this large will look random and will be not meaningful to an observer when observed with the naked eye. However a closer examination of these datasets may reveal patterns and relationships hidden within them. The process of discovering these hidden patterns and relationships from large datasets is known as Data Mining. It is all about solving problems by analysing data that already exists in databases (Witten & Frank 2005). The patterns revealed from data mining are useful information that can be used for significant predictions on new data. For instance, the loyalty of a customer in telecommunications industry is vital key for its growth in highly competitive market. For this, a telecommunication industry can utilize their existing database of past customers that holds the information about the customers' profile and its choices. Finding the patterns of behaviour in these datasets and analysing them can identify critical aspects of those who are both likely to remain loyal to a particular telecommunications company and those who switch between different companies. Hence, a telecommunication industry can use this information to classify present customers that may switch to other rival industries and then can offer them with attractive incentives to change their mind.

Machine Learning algorithm uses the data mining techniques to learn the patterns, behaviours and trends within large datasets. It involves making computer systems or applications that learn general concepts (patterns, behaviours, trends) and act accordingly. For instance, if a computer system could learn the typical behaviour of a customer and if a customer was behaving in strange manner, it could mark such behaviour, effectively profiling cases that may need further examination, whether it be for dishonesty or some other possible actionable behaviour. Generally, machine learning algorithms learn the general concepts from training sets generated from an existing dataset. A training set is a dataset that consists of group of examples used for learning where the target value or output is already known. A computational model is built from this training set during its learning process. The model built is then used as a predictor to classify or predict previously unseen data. This new set of data is called test set that evaluates the performance of a model built by examining its accuracy of prediction. This accuracy is evaluated by calculating the difference between the predicted values with that of actual values in the test set. A computational model can be built using different forms of machine learning algorithms. Some of these forms are search control rules, decision lists, state transition networks, concept hierarchies and inference networks. The concepts and workings of various machine learning algorithms are different but their common goal is to learn from the data provided to build computational model (Polumetla 2006).

In this thesis, machine learning algorithms are used to model the predictive classifiers. There are several types of machine learning algorithms. The main two types are Unsupervised Learning and Supervised Learning. In simple words, the main idea of unsupervised learning algorithm is that the computers learn how to do something by itself whereas in supervised learning algorithm, they are taught how to do these particular things. Section 3.2.1 and 3.2.2 of this chapter gives the detailed information on these two types of algorithms. In this thesis, supervised learning algorithms are adopted to build the predictive models for prediction purposes. Some of these selected algorithms are described in Section 3.3.

3.2.1 Unsupervised Learning Algorithm

Unsupervised Learning algorithm are one of the important types of machine learning algorithm. In unsupervised learning, systems learn how to do something without being explicitly told how to do it (Oladipupo 2010). The models built using unsupervised learning algorithms do not have labels and known target results. Instead, they are organized by inferring statistical structures present in the overall collection of patterns in the training dataset (Dayan 1999**;** Jasonb 2013).

There are number of different types of unsupervised machine learning algorithms, such as collaborative filtering, association rules mining and so on. However, Cluster Analysis is one of the most popular and important type of unsupervised machine learning algorithm. It is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modelled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance. Following Figure 2 is an example of clustering where the instances are grouped into three clusters.

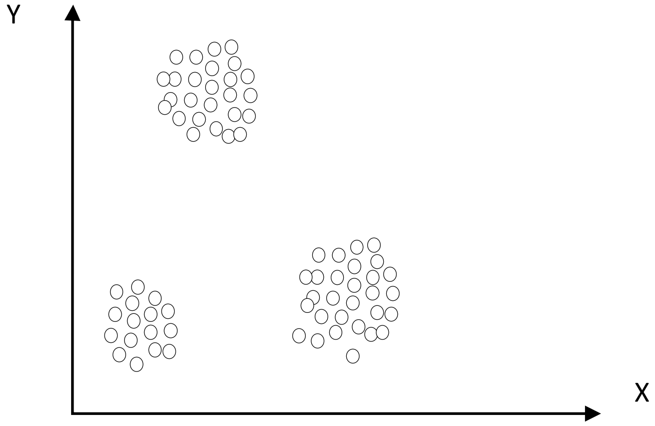


Figure 2: Example of Unsupervised Machine Learning algorithm (Clusters)

Unsupervised learning methods are used in bioinformatics for sequence analysis and genetic clustering; in data mining for sequence and pattern mining; in medical imaging for image segmentation and in computer vision for object recognition.

3.2.2 Supervised Learning Algorithm

Supervised Learning Algorithm is one of the most popular and common types of machine learning algorithm. This algorithm is mainly used to get the computers to learn the classification systems from examples of datasets. Therefore, it solves commonly the classification problems and regression problems. The most common examples of classification problems are Digit recognition, fraud detection, spam email filtering etc.

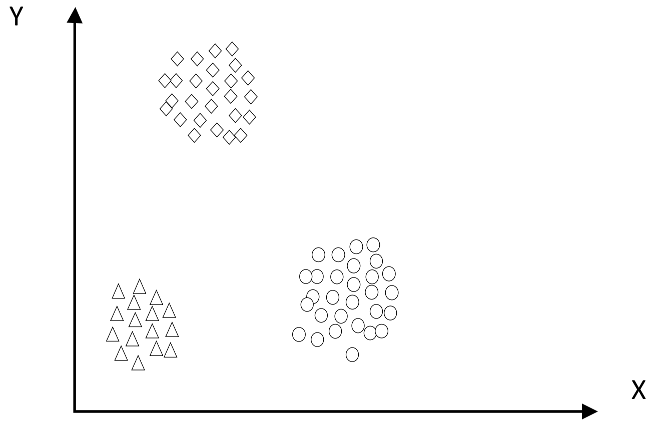


Figure 3: Example of Supervised Machine Learning algorithm

In supervised machine learning algorithm, a model is built by a training set which has its target values as outputs. Training data is an input data that has a known labels such as spam or not spam for emails or fraud or not fraud for customer categorization. A predictive model is prepared using these training data through a training process where the model is required to make predictions and in case of wrong predictions, they are required to correct them. This training process is continued till the model is successful to achieve the desired level of accuracy on the training data.

For this thesis, the most common examples of classifications learning algorithms such as Naïve Bayes, Multilayer Perceptron Neural Networks, and Classification Trees (Decisions Trees) are used to build the predictive models for location prediction purposes. These algorithms are broadly described in following Section 3.3.

3.3 Predictive Models

Predictive models can be defined as a classifier that is developed using machine learning algorithms to predict the future location of people based on its past history of locations. This predictive model extracts the information from large mobile dataset to find the hidden patterns that helps in forecasting the location of an individual. The predictive models in this thesis are built by utilizing supervised machine learning algorithms. Some of these popular algorithms from supervised machine learning methods are as follows:

3.3.1 Naïve Bayes

Naïve Bayes is a simple classification method based on Bayes Theorem of conditional probability. It works under the assumption that each input attribute is conditionally independent of all other attributes present in the dataset. Prior to further description on this algorithm, it is beneficial to understand what Bayes theorem is. A Bayes Theorem as described by Thomas Bayes is a mathematical formula that is used for calculating conditional probability of an event occurring given the probability of another event that has already occurred. Mathematically, it can be stated as follows:

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \qquad \textbf{(1)}$$

Where, $\mathrm{P(A|B)}$ is the probability of event A occurring given that event B has occurred. In this algorithm, $\mathrm{P(A|B)}$ is known as posterior probability and $\mathrm{P(B|A)}, \mathrm{P(A)}$ and $\mathrm{P(B)}$ are known as prior probabilities. Bayes Theorem defines the relationship of these two probability types. It allows to calculate the probability of observing event A given B when the probability of observing event B given A is known and the individual probabilities of A and B is also known.

Naïve Bayes algorithm is a simple probabilistic model that makes use of all the attributes present in the given data and studies each attribute with equal importance and independence. For example, an animal can be considered to be an elephant if it has huge size, long trunk, large tusks and short tail. However, all these attributes seem to depend on each other to consider an animal to be an elephant but a Naïve Bayes classifier observes all these attributes to be independent of each other and analyses them individually. So, if a Naïve Bayes classifier is classifying an animal to be an elephant, it will not check whether it is huge in size with long trunk having large tusks and short tail. Instead, it will separately check whether the animal has a trunk, whether it has tusks, whether it is huge in size etc. It works under the assumption that each attribute present in the dataset are independent of each other.

The naïve Bayes algorithm assigns a vMAP value to the output attribute, Output_C for a given case with two input attributes, input_A with value a and input_B with value b. This value (vMAP) is the highest probable value across all possible values of output attribute. This is referred to as the maximum-a-posteriori (MAP) rule. Given the input attribute values, the probability of the output attribute taking a value $\mathrm{v_j}$ can be represented as

$$\mathrm{P(v_j|a,b)} \qquad \textbf{(2)}$$

Calculating this probability value as such is difficult. Hence, Bayes theorem can be applied to this equation,

$$\mathrm{P\left(v_j\middle|a,b\right) = \frac{P(a,b\,|v_j)\,P(v_j)}{P(a,b)} = P\left(a,b\,\middle|v_j\right)\,P(v_j),} \qquad \textbf{(3)}$$

Where the probability of observing $\mathrm{v_j}$ as the output value is represented by $\mathrm{P(v_j)}$, and the probability of observing input attribute values $\mathrm{a, b}$ together when $\mathrm{v_j}$is the output value is represented by $\mathrm{P(a,b|v_j)}$. However, if there is large number of input attributes $\mathrm{a, b, c, d}$ .... then the data will be insufficient to calculate the probability $\mathrm{P(a,b,c,d\,...\,|v_j)}$.

The naïve Bayes algorithm provides the solution for this kind of problem by assuming that the input attributes $\mathrm{a, b, c, d}$ ... are all conditionally independent of one another given the output $\mathrm{v_j}$. In other words, it assumes that the values taken by an attribute is independent of the values of other attributes in the instance of any given output. By using the assumption of conditional independence, the probability of each input, for a given output value, can be multiplied to determine the probability of observing an output value for the inputs. The probability value $\mathrm{P\left(a,b\middle|v_j\right)}$ can then be reduced to

$$\mathrm{P\left(a,b\middle|v_j\right) = P\left(a\middle|v_j\right)P(b|v_j),} \qquad \textbf{(4)}$$

Where, $\mathrm{P\left(a\middle|v_j\right)}$ is the probability of observing value a for the input attribute, input_A when the output value is $\mathrm{v_j}$, and $\mathrm{P\left(b\middle|v_j\right)}$ is the probability of observing value b for the input attribute, input_B. Thus, the probability of an output value $\mathrm{v_j}$ to be assigned for the given input attribute is

$$\mathrm{P\left(v_j\middle|a,b\right) = P\left(v_j\right)\,P\left(a\middle|v_j\right)P(b|v_j)} \qquad \textbf{(5)}$$

Learning in the Naïve Bayes Algorithm includes discovering the probabilities of $\mathrm{P}(\mathrm{v_j})$ and $\mathrm{P}(\mathrm{a_i}|\mathrm{v_j})$ for all possible values taken by input and output attributes based on the training set provided. The probability $\mathrm{P}(\mathrm{v_j})$is acquired from the proportion of the number of time the value $\mathrm{v_j}$ is seen for the output attribute to the total number of instances in the training set. For an attribute at position i with value $\mathrm{a_i}$, the probability $\mathrm{P}(\mathrm{a_i}|\mathrm{v_j})$is obtained from the number of times $\mathrm{a_i}$ is seen in the training set when the output value is $\mathrm{v_j}$.

In naïve Bayes algorithm, all attributes in the dataset are required to be discrete. The attributes with continuous values have to be discretized prior to their use. It is not allowed for an attribute to have missing values as it may create problems to calculate the probability values for that attribute. A default value for that attribute can be used as a replacement to deal with the missing values (Polumetla 2006).

3.3.2 Multilayer Perceptron

A Multilayer perceptron is one of the most common form of feed forward neural networks. Feed forward neural networks are the simplest type of neural network. A neural network, widely known as Artificial Neural Network is a mathematical information processing model inspired by biological neural network. It is composed of large number of highly interconnected processing units, appropriately also called a neuron. Figure 4 (Vidal n.d.) shows the basic representation of an artificial neuron.

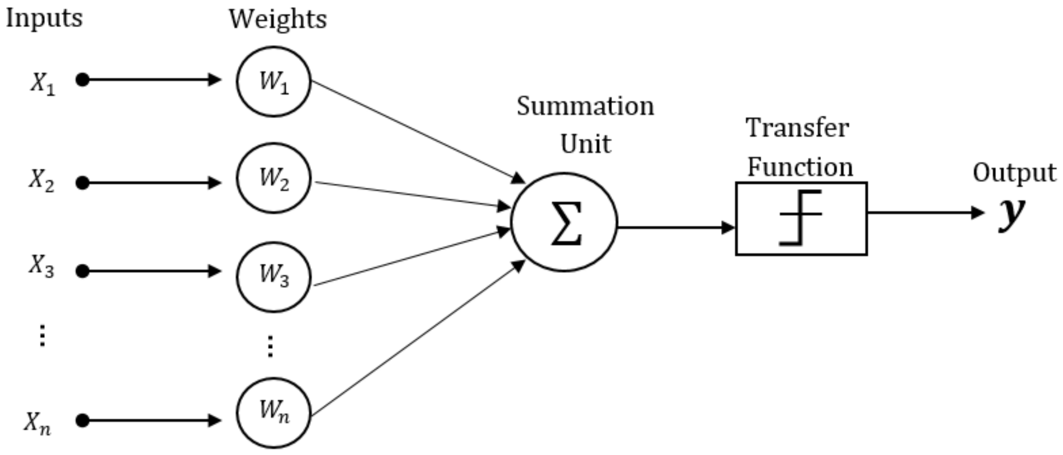


Figure 4: Artificial Neuron (Vidal n.d.)

The basic component of a multilayer perceptron is the neuron. The basic computational neuron is often represented as a node or unit which receives one of more inputs ($\mathrm{X_{1\ldots n}}$) from other units. Each of these inputs are multiplied by weights ($\mathrm{W_{1\ldots n}}$). Every neuron in one layer are usually connected to every neuron in an adjacent layer and the activation function of the neurons is generally sigmoid or linear. In fact, architectures and various types are determined both by the different topologies as well as the choice of the activation function. The values of the functions that are associated with the connections are called “weights”. The summation unit is computed to produce the sum of results which then will be computed by transfer function that determines the activation of the neuron.

An MLP is one of a kind of neural network that is trained with a backpropagation learning algorithm. It is made up of interconnected multiple layers of computational units that are linked in a feed forward manner forming a direct connection between units of lower and subsequent layers. An MLP consists of a basic structure of an input layer, one or more hidden layers and one output layer. The term “hidden” is applied to the units in the hidden layer because their output is used only within the network and is never visible outside the network. To predict the location in the Reality Mining dataset, an MLP with one hidden layer can be used, as shown in Figure 5. The units in subsequent layer of MLP uses the output from a lower unit as an input. It has an associated weight for connections between units in subsequent layer.

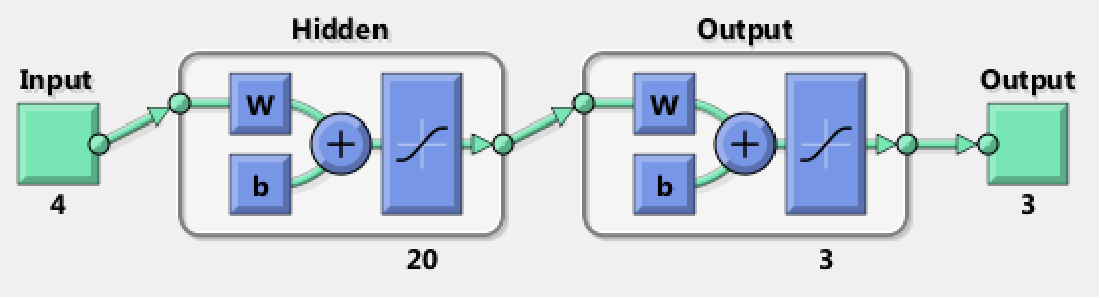


Figure 5: A Multilayer Perceptron with hidden layer

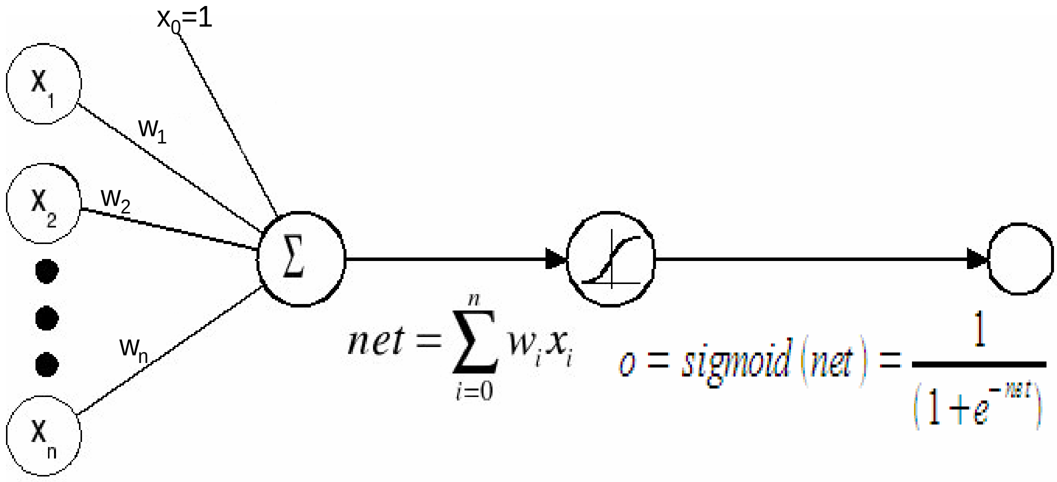

$x_0=1$
X_1
X_2
X_n
W_1
W_2
W_n
Σ
$net = \sum_{i=0}^{n} w_i x_i$
$o = sigmoid(net) = \frac{1}{(1+e^{-net})}$

Figure 6: Sigmoid Threshold Unit that takes Inputs x_i with Weights w_i (Polumetla 2006)

The hidden and output units are based on sigmoid units. A sigmoid unit is shown in Figure 6 (Polumetla 2006). It computes a linear combination of its input, then applies a threshold to the result. The sigmoid function, for net input (x) can be defined as:

$$\mathrm{sigmoid}(x) = \frac{1}{(1+e^{-x})} \qquad \textbf{(6)}$$

Sigmoid (x) is the output of a sigmoid unit. It is a continuous function of its input(x) and is in the interval (0, 1). In addition to the inputs provided to it, the sigmoid unit also takes in a constant input of 1.

Using the backpropagation algorithm, an MLP learns to adjust its weights (Rumelhart et al. 1986). A set of training instances are used by the backpropagation algorithm for the learning process. The weights in the feed forward network are initialized to small random numbers for training. Each training instance is sent through the network and the output from each unit is calculated. This output computed by the network is compared with the target output calculating the error, which is fed back through the network. For adjusting the weights, the backpropagation algorithm uses a gradient descent to minimize the squared error. At each unit in the network, from output unit to hidden units, connection weights are adjusted and errors are reduced by using its error value. The weights are updated by using:

$$w_{ji} = w_{ji} + \eta \delta_j \, x_{ji} \qquad \textbf{(7)}$$

Where, w_{ji} is the weight from unit i to unit j, η is the learning rate, δ_j is an error gained at unit j, and x_{ji} is the input from unit i to unit j. This process of using training instances to adjust the weights is repeated for a fixed number of times or until the error value is small (Carrasco-Ochoa et al. 2010). To improve the performance of backpropagation algorithm, the weights updated at the n^{th} iteration of the backpropagation is made partially dependent to the amount of weight changed in the $n - 1$ iteration. The term “momentum (α)” determines this amount of weight change that is contributed by the $n - 1$ iteration. Now, the weight at n^{th} iteration is updated by using the new rule:

$$\Delta w_{ji}(n) = \eta \delta_j \, x_{ji} + \alpha \, \Delta w_{ji}(n-1) \qquad \textbf{(8)}$$

To achieve faster convergence to a minimum, this momentum term is added.

3.3.3 Classification Trees

Classification Trees (also known as Decision Trees) are predictive models commonly used in data mining for classification or prediction of outcomes. In classification trees, a model is created that predicts the value of a target variable based on several input variables. The variables which go into the classification can be numerical or categorical values. Classification Trees are useful as they provide fair and comprehensive predictors in situations where there are many variables which interact in complicated and non-linear ways.

In classification trees, the process of classifying is started with a single node, and then look for the binary distinction which gives the most information about the class. It then takes each of the resulting new nodes and repeat the process there, continuing the recursion until it reaches some stopping criterion.

The first step of classification tree algorithm is to find the initial split. It starts its process with a training set consisting of pre-classified records. Pre-classified refers to the target value or dependent variable that has a known class. Its main goal is to build a tree that classifies output among the target values. The criteria of splitting process initially generalizes to multiple target values and any multi-way partitioning can be achieved through repeated binary splits. To choose the best splitter at a node, the algorithm considers each input value to be sorted. Then it tries and considers each possible split and the best split is one that produces the largest decrease in diversity of the classification label within each partition. The process is repeated for all input fields and best split is chosen for that node before continuing the process to next node in same manner, generating full decision tree at the very end. The resulting tree is often too large so it requires pruning using cross-validation.

Pruning is the process of removing leaves and branches to improve the performance of the decision tree when it moves from the training data, where the classification is known to real-world applications, where the classification is unknown and is to be predicted. The best split is made by the tree-building algorithm at the root node because a large number of records and a lot of information is located there. Each subsequent split has a smaller and less representative population with which to work. Towards the end, idiosyncrasies of training records at a particular node display patterns that are peculiar only to those records. These patterns can become meaningless and sometimes harmful for prediction if an attempt is made to extend rules based on them to larger populations. For example, say the classification tree is trying to predict height and it comes to a node containing one tall person named X and several other shorter people. It can decrease diversity at that node by a new rule saying "people named X are tall" and thus classify the training data. In a wider universe this rule can become less than useless. Pruning methods solve this problem. It lets the tree to grow to maximum size and then removes the smaller branches that fails to generalize.

Since the tree is grown from the training data set, when it has reached full structure it usually suffers from over-fitting i.e. it is "explaining" random elements of the training data that are not likely to be features of the larger population of data. This results in poor performance on real life data. Therefore, it has to be pruned using the validation dataset (FrontLine Systems 2012).

3.4 Summary

The purpose of this chapter has been to present an overview of techniques in the field of Artificial Intelligence. The basic concepts, history, and some of the specific learning algorithms in machine learning techniques have been presented along with the motive to build predictive models in this thesis.

Based on past observations, machine learning techniques study how to automatically learn to make accurate predictions. It is shown that supervised machine learning algorithms have been used effectively in the modelling of classification problems such as Digit Recognition, Fraud Detection, and Spam Email Filtering etc. This suggests that supervised machine learning algorithms may have the ability to assist in predicting the future locations of individuals at a given time by identifying patterns present in their past location data.

Naïve Bayes, Multilayer Perceptron, and Classification Trees have been identified as an excellent learning algorithms in supervised machine learning techniques to develop the predictive classifier using Reality Mining data for location prediction. Each of these algorithms to be applied, have the key ability of generalizing, from the provided training set of data, to new unseen sets of data. Some of the major advantages of using these algorithms are:

- Naïve Bayes - short computational time for training, easy to implement.
- Classification Trees - intuitive classification of data based on limiting features.
- Multilayer Perceptron - good at solving pattern recognition and forecasting problems.

The process of how each of these machine learning algorithms work has also been presented in this chapter. In the next chapter, the methodology of implementing these techniques on location-driven datasets of the Reality Mining project for the purpose of developing predictive models, that predict the future location at a given time, is discussed.

Chapter 4

Methodology

4.1 Introduction

The aim of this thesis is to find the predictable patterns in movement behaviour of an individual's daily life from mobile phone data. To this end, predictive models are developed using machine learning methods that can forecast the future locations of individuals based on the past location data records collected from their mobile phones. As outlined in Chapter 2, the Reality Mining Project is one of the largest mobile phone projects ever designed to capture broad range of information on human behaviour using mobile phones. Hence, datasets from this project are used for learning human movement patterns to build location prediction methods. The learning algorithms selected to build these methods are described in Chapter 3.

In order to predict the future locations of individuals at a given time, the previous data records of its locations can be utilized. Data collected from mobile phones such as cell tower ID and time-stamp records can be used to find patterns in individual's movement behaviour. Therefore, these features from Reality Mining dataset are used to develop the machine learning methods. Each of these methods are trained and tested using earlier six months of training and later three months of testing set generated from the Reality Mining dataset of nine months. The process of generating this training and testing set from complete Reality Mining dataset is further detailed in Section 4.4.

In the first section of this chapter, dataset of the Reality Mining project is presented with an intention to describe location driven data records that are used for building predictive models. For an effective predictive model, it requires the data to have the well-organized form of features. Therefore, dataset of Reality Mining project is pre-processed, as described in Section 4.3. In Section 4.5, the use of this pre-processed data for building predictive models are discussed in detail. It also describes how performance of each predictive models are measured. Finally, in the last section, the process of calculating the entropy level of an individual is described. Here, the entropy level of an individual is defined as the amount of randomness found in an individual for being at certain places at a given time. This section is introduced in order to find out the affects in performance of all predictive models when applied to predict the future location of high entropic individuals versus low entropic individuals.

4.2 Datasets

The dataset of Reality Mining project is a location driven data collected using mobile phones that were installed with context aware software application, as described in Chapter 2. The pioneers of Reality Mining project have anonymised this collected data and were made available for academic researchers to freely download from their official website of Reality Mining project, “http://realitymining.com/download.php”. However, the dataset made available in the form of MySQL relational database only covers nine months of data for one single subject. For this reason, a request was made to Nathan Eagle, one of the pioneers of Reality Mining project, to provide the complete dataset. As a result, anonymised complete dataset was provided in MySQL relational database that is used in this thesis. This dataset contains ten large tables that represents cell tower ID records, call records, text messages, activity logs and proximate Bluetooth device records.

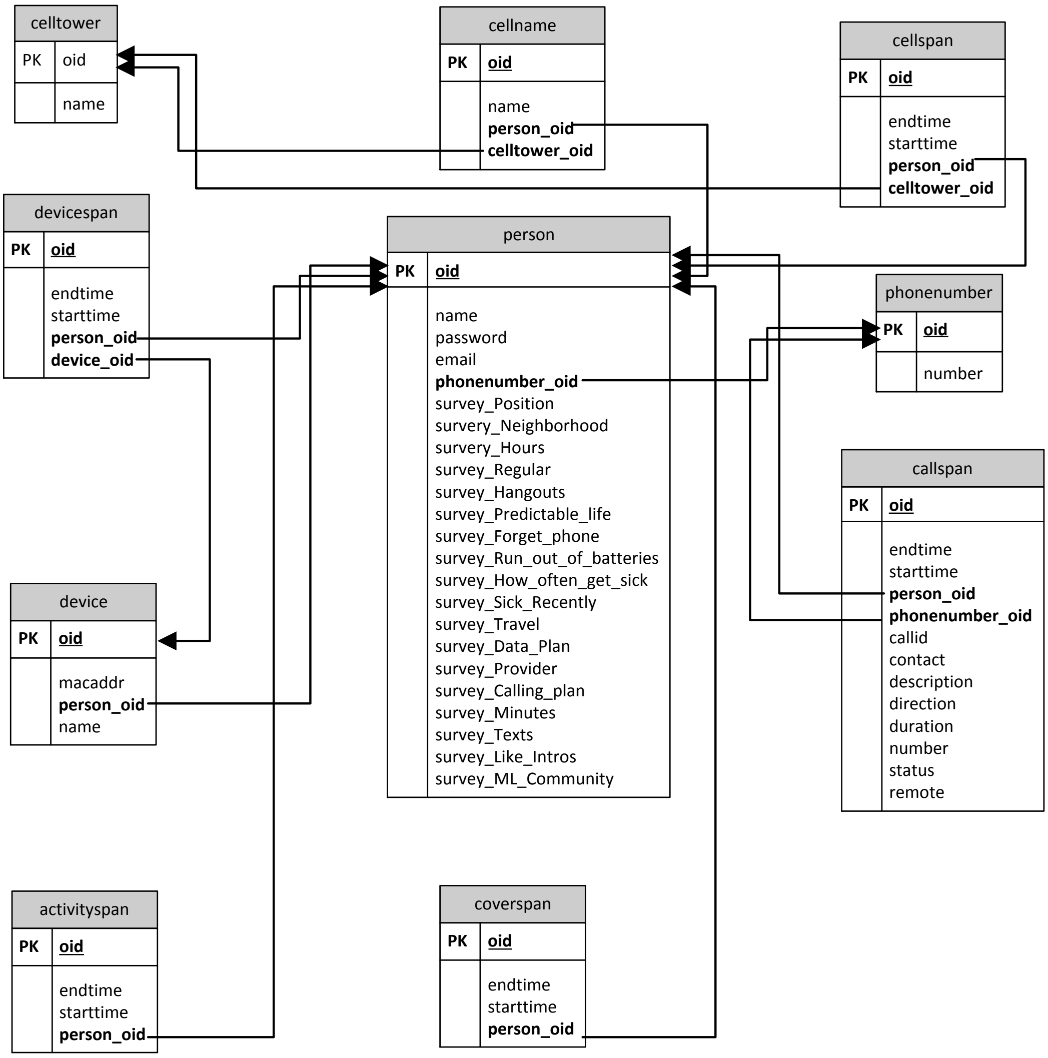


Figure 7: MySQL relational database of the Reality Mining Dataset

As detailed in Section 2.5.2 of Chapter 2, these table of records are gathered from one hundred subjects at MIT over the course of nine months and holds information on human behaviour such as location, nearby subjects, communications and phone usages. Figure 7 illustrates this complete anonymized dataset of Reality Mining project that is made available in MySQL relational database. However, only three tables are needed for this study. These three tables are named as 'cellspan' 'cellname' and 'celltower' in the Reality Mining dataset. The details of these tables are also shown in Figure 7.

4.2.1 Celltower Data

In the Reality Mining dataset, 'celltower' is a data table that holds the record of unique cell towers that are registered in mobile phones of all participants in the study. There are altogether 32,656 unique cell towers registered in this table. The cell tower is registered into an individual's mobile phone whenever it is connected to previously unregistered cell tower. Researchers of the Reality Mining project later assigned a unique number to these cell towers to identify each unique cell tower that are registered for the study. Following Table 1 demonstrates the structure of this data.

oid	name
41	2421, AT&T Wirel
42	111, AT&T Wirel
43	40793, T-Mobile
44	40333, T-Mobile
45	40341, T-Mobile

Table 1: Example of table 'Celltower' in the Reality Mining Dataset

4.2.2 Cellname Data

In the table "cellname" of the Reality Mining dataset, there are altogether 825 unique cell names labelled by the subjects of the Reality Mining project. In total, seventy seven out of one hundred subjects have labelled these cell towers. For instance, subjects in the project have labelled the cell towers with names such as 'Home' which represents location of that tower relative to the association that the subject has with that particular area. However, this can cause problems such as cell tower ID labelled with the name, 'Home' for one may not be same for other subjects. Following Table 2 demonstrates the sample data present in 'cellname' table.

Oid	name	person_oid	celltower_oid
40	Home	75	4
41	Red house	75	5
42	ML	43	1842
43	Cambridge st	43	1843
44	Media lab	43	1844

Table 2: Example of table 'Cellname" in the Reality Mining Dataset

4.2.3 Cellspan Data

In the table, "cellspan" of the Reality Mining dataset, there are altogether 2.54 million records on mobility patterns of human behaviour. This patterns are recorded over the course of nine months in the year 2004 – 2005. The mobile phone carried by the each subject from Reality Mining project records the continuous data of subject's movement activity whenever their mobile phone makes a move from one cell tower to another. This data includes the timestamps (starttime, endtime), cell tower IDs and person IDs. Following Table 3 demonstrates this table.

oid	starttime	endtime	person_oid	celltower_oid
998	2004-09-18 13:10:34	2004-09-18 14:12:56	94	3
999	2004-09-18 14:12:56	2004-09-18 19:36:55	94	4
1000	2004-08-19 19:36:55	2004-08-19 19:46:31	94	22

Table 3: Example of table 'Cellspan' in the Reality Mining Dataset

4.3 Pre-processing of Dataset

In this section, the approaches used to address the inconsistencies found within anonymised Reality Mining dataset are discussed. The location data from Reality Mining consists of continuous time stamped transitions, as shown in Table 3 of Section 4.2.3. These time stamped transitions indicates the amount of time, a mobile phone of an individual was in connection with a particular cell tower before moving to other. The gathering of this time stamped location data from subjects in the Reality Mining project took place for 9 months. However, many of the subjects in this project are missing this time-stamped location data for certain times. The example of this problem is demonstrated in the following Table 4.

oid	starttime	endtime	person_ oid	celltower _oid
66	2004-07-30 16:21:36	2004-07-30 16:21:49	94	42
67	2004-07-30 16:21:49	2004-07-30 16:22:19	94	42
68	2004-08-24 14:05:24	2004-08-24 14:06:09	94	38

Table 4: Example of missing fields on continuous timestamps

In the above Table 4, Subject 94 is missing records of location data for approximately a month as shown in between order IDs (oid) 67 and 68. It is believed that these continuous time-stamped data are missing due to the subjects turning off their mobile phones and data corruption. Nevertheless, this record table holds approximately 85% of the time on average since the data collection took place (Eagle, 2005). For this reason, the default missing data in the Reality Mining dataset are ignored in this thesis.

As described in Section 4.2, the three tables of the Reality Mining dataset selected for this thesis holds the location information of one hundred participants in the study. The location of an individual can be determined from the cell tower IDs recorded within their mobile phones. A cell tower ID is a unique number assigned to each recorded cell tower that holds the information about the individual's location and its movement patterns. However, to infer the location of an individual from the celltower ID, it requires the cell tower ID to be labelled with some type of physical place name that tells where the specific cell tower is located at. This process of labelling cell tower ID and methods for categorizing each of these labels to three different states are described in subsection 4.3.1. Once this process is done, pre-processing to extract the features and variables from labelled data is described to show how it can be used for building machine learning methods in subsection 4.3.2.

4.3.1 Labelling of Celltower ID

In this section, methods used for providing standardized physical location names to the number of unique cell tower ID are described. These physical location names are then used for corresponding to three different states as Home, Work, and Elsewhere.

In the Reality Mining project, number of subjects have manually entered various cell names to different cell towers that are registered to their mobile phones. These cell names are the physical place names that represent the position of the registered cell towers, associating it to the subject's whereabouts. However, most of these cell towers are labelled with unstandardized physical location names, such as "ML" which essentially stands for "Media Laboratory". Furthermore, many of the registered cell towers are also left empty as "Null", meaning no cell names for certain cell tower IDs. This causes problems to label such unlabelled cell towers with physical location names.

oid	starttime	endtime	person_oid	celltower_oid	name
538378	2004-08-19 19:36:46	2004-08-19 19:37:50	96	1844	ML
538379	2004-08-19 19:37:50	2004-08-19 19:56:55	96	1842	Office
538380	2004-08-19 19:56:55	2004-08-19 19:58:31	96	38	ML

Table 5: Example of new dataset, merging Cellspan and Cellname

In this thesis, the two tables 'cellspan' and 'cellname' are merged together to create a new dataset that relates the cell tower ID with its place name for a given time event, as shown in Table 5. To do this, the initial step was taken to list out all the cell names labelled by each subject to its unique cell tower ID. These cell names having ambiguous meaning or unstandardized names are then labelled with standard physical location names using Google Maps and technical document called 'Supporting Online Information' released by MIT Human Dynamics Laboratory. In order to assist academic researchers with the understanding of anonymised Reality Mining datasets, MIT Human Dynamics Laboratory released this document, 'Supporting Online Information' which provides all relevant details on data collection and analysis, including an explanation of the subject pool, data collection, protocols, a description of variable construction and a summary of data analysis (Eagle & Pentland, 2005). This publication also suggests that cell names labelled by each subjects in MIT Reality Mining Experiment can be used to infer which particular cell towers are associated with "Work", "Home" and "Elsewhere". As a result, each location names labelled by each subject is then used to correspond these three states as Home, Work and Elsewhere.

However, towers without cell names from subjects in the project are not included in this thesis for further data analysis. This is due to the fact that inferring physical location names without hints of subject's cell names would have lead only to assumptions than assertion of the exact physical position of that cell tower.

4.3.2 Features for Machine Learning Methods

In this thesis, the available features in location data of the Reality Mining project, such as time-stamps, celltower ID and person ID are used in machine learning methods for building location prediction methods that can predict the future locations of individuals. These features are chosen for machine learning methods because they represent the essential aspects of the location data that can be used for predicting future locations at given times. However, pre-processing on the variables of these features are required to make this data usable for machine learning methods. In this section, the techniques used to transform available features of the Reality Mining dataset into common format is described. These refined variables are then used as input and output features to train and test the machine learning methods.

In the Reality Mining dataset, the recorded time-stamps of location data represents the time intervals that took place when connection was made between particular cell towers and individual's mobile phones. For machine learning methods in this thesis, these recorded time-stamps namely 'starttime' and 'endtime' as shown in Table 5 of Section 4.3.1 were transformed to represent only number of specific day of the week ['1 for Monday', '2 for Tuesday'… '7 for Sunday'], hour of that day [0, 1, 2 … 23] and length of the connection time in seconds [21, 3, 12 … 45] for each record in the dataset.

The person_oid in location data is the unique number assigned to specific individual who participated in the Reality Mining project for data collection. The dataset to be used for machine learning methods includes location data for number of individuals. Therefore, a celltower oid representing 'Work' for one individual may not be same for others. With this in knowledge, person_oid is included as input feature in training and testing datasets.

The celltower_oid is one of the important feature in location data. It defines the specific location area in the study. The cell names labelled to these cell tower IDs can be used to figure out the physical positioning of each cell towers in relation to associated subjects. This type of information can be used to reveal particular subject's location. Therefore, celltower_oid is included as input feature for training and testing machine learning methods. However, cell names associated to these cell tower IDs are eliminated from the dataset to be used, once the corresponding states are evaluated from it. The corresponding states are categorized location areas used for grouping number of different cell names labelled to cell tower IDs into three precise states as “Home”, “Work” or “Elsewhere” for location records of individuals. In this thesis, these corresponding states are the output class features that are meant to be determined by the predictive models developed using machine learning methods. The following Table 6 demonstrates this pre-processed dataset to be used for machine learning methods to build the predictive models.

day	hour	time_length	person_oid	celltower_oid	states
6	13	21	94	3	Work
6	14	245	94	4	Home
6	19	45	94	22	Elsewhere

Table 6: Pre-processed MIT Reality Mining datasets

4.4 Training and Testing sets

To build a predictive model, using machine learning algorithms that can predict the future locations, a training set is required. A training set consists of data related to past location patterns of an individual. Once the model is built, the performance of it is then evaluated using test set. A test set consists of unseen data on location patterns. As described in Section 2.5.2 of this chapter, the Reality Mining Dataset is a location data collected using mobile phone for nine months. In this thesis, this location driven dataset of nine months period is broken down into two sets to create training and testing set.

Machine learning algorithms require training and testing set, comprising of input and output features where output feature is the target value for prediction. It uses the information from input and output features of training set to build the model. The input attributes in test set is then applied to this model to predict the output values which is then compared with the actual value to estimate the performance of the model built. The features with input and output values for machine learning algorithm takes the form as follows:

input 1, input 2, input 3…, input n, output

In this thesis, the training set is made up of earlier six months of the Reality Mining dataset. As presented in Table 3 of Section 4.2 of this chapter, {‘day’, ‘hour’, ‘person_oid’, ‘celltower_oid’} are the input features for the predictive model whereas {‘states’} that is associated to cell tower ID is an output feature in the training set. On contrary, testing set represents similar set of data but holds the later three months of the Reality Mining data. However, some of the subjects are missing in either months of training or testing set after the division of the Reality Mining data since all one hundred subjects did not take part in entire period of nine months experiment consistently.

4.5 Location Predictive Model

The machine learning methods are used to build a predictive model to classify the future location of the subjects based on its previous location patterns. As described in Section 3.3 of this thesis, the machine learning algorithms namely Classification Trees, Naïve Bayes, and Multilayer Perceptron are used to forecast the individual’s locations. In all these classification algorithms, the input features that contains a sequence of previous day, hour and recorded cell towers at those times are included along with the past mobility patterns that is to be predicted as if the user is either at Home, Work or Elsewhere.

The dataset with location information were derived from the Reality Mining project. This dataset is then split into training and testing set on the basis of prior six and later three months. The machine learning algorithms use the training set to build a model and the test set is used to evaluate the performance of that model.

In the Reality Mining dataset, the model built using machine learning algorithm takes a given set of input attributes (day, hour, recorded celltower ID for that time, and person’s ID whose mobile phone recorded all these information) to predict the value for the output as subject’s whereabouts. The results on location prediction are represented in the form of a confusion matrix, with rows corresponding to actual values and columns corresponding to predicted values for the output attributes. Each block in the confusion matrix gives the number of times the actual class is predicted as the class given by the column. The numbers in the diagonal blocks give the number of time the predicted class value was equal to the actual class value. Thus, the sum of entries along the diagonals divided by the total number of instances present in test set, gives percentage of the number of correctly classified instances.

The performance of machine learning algorithms can be determined by the difference between the actual value and predicted value, which gives the amount of error in the prediction made. The closeness of the predicted value to the actual value depends on the efficiency of the model. For a model whose prediction accuracy is high can be considered to be the best predictive model for location prediction.

4.6 Entropy

An entropy level of a person determines the predictability of its behavioural patterns (Eagle, 2005). Thus, achieving good accuracy results in predicting the future location of people highly depends on its entropy level. Generally, an entropy level of a person refers to the nature of randomness in its behavioural patterns. For instance, if a person is said to have a high entropy level, it means that the person has extremely random behavioural patterns which makes it hard to predict the person’s behaviour. On the other hand, person with low entropy level refer to less randomness in their behaviour due to which predicting its behavioural patterns can be easy. In this section, it is intended to explain how a low and high entropy level subjects are selected and how they differ in achieving good accuracy results while prediction of its future location.

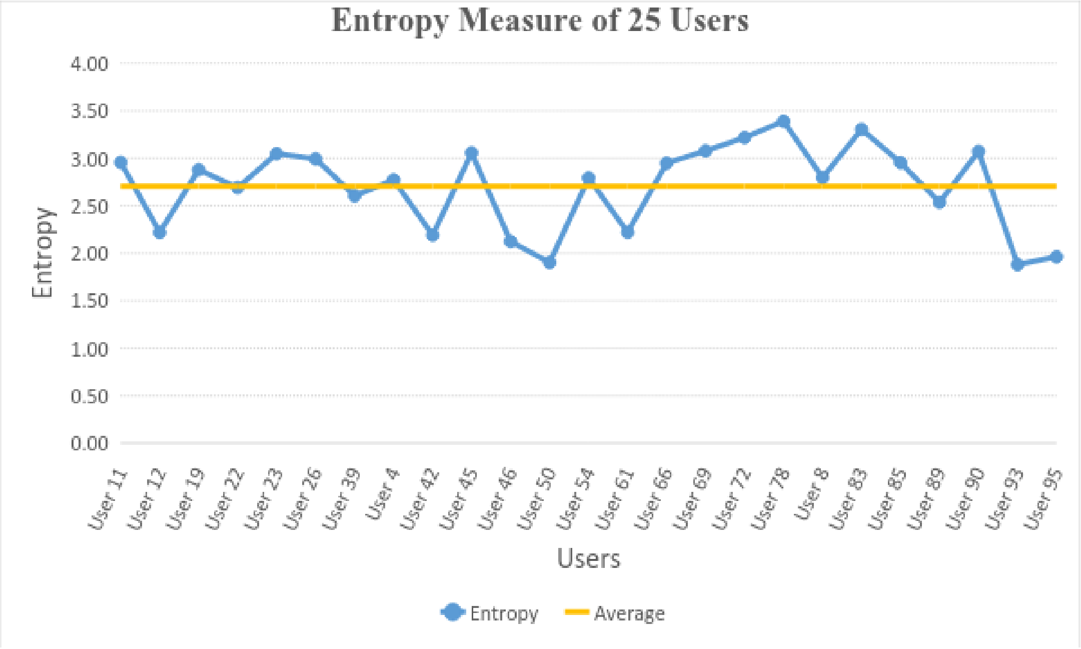


Figure 8: Behavioural Level measured by Shannon Entropy

In order to perform Shannon Entropy on subjects of the Reality Mining dataset, twenty five subjects were selected in random order to measure their entropic behavioural level. Initially, all unique celltower records covered by each subject are listed. For each subject, the frequency of each cell tower ID found in their mobile phone for prior six months of the Reality Mining Experiment is then calculated. Frequencies of all detected celltower

IDs are summed up together to find the total occurrences of overall celltower IDs. The probability of each celltower ID is then evaluated using this sum for each subject. The entropy level for each celltower is determined using Shannon Entropy on this probability of each celltower occurrence. The total sum of Shannon Entropy from all occurring cell towers for each subject gives the entropy level of a subject.

In the Figure 8, it demonstrates the graph plot of both high and low entropy level of twenty five subjects. The orange straight line in the graph is an average entropy level calculated from entropy level of all selected twenty five subjects. This line is used as a divider to categorize high and low entropy subjects. In the Figure 8, it is suggested that the highest entropic subject is User 78 with entropy level of 3.390 whereas the lowest entropy level is 1.881 for the User 93, making it the lowest entropic subject among randomly selected twenty five subjects from the Reality Mining dataset.

In this section, the topic of interest is to find if a subject with entropic behavioural level affect in achieving the successful accuracy results while predicting the future location. For this, prior six months of dataset of twenty five entropic users is created in same manner as mentioned earlier in Section 4.3 of this chapter. This training set is intended to model different machine learning algorithms mentioned in Chapter 3 of this thesis as the predictive classifier to find next location of entropic users. The testing set is created from the later 3 months of each subject from twenty five subjects. From the Figure 8, high entropic subjects and low entropic subjects are categorized. The testing set of highest entropic users [78, 83, and 90] on later 3 months of the Reality Mining dataset is created. Similarly, the testing set of lowest entropic users [50, 93 and 95] on later 3 months of the Reality Mining dataset is created. In Chapter 5, the experiments for achieving successful accuracy with various machine learning algorithms to predict the future location of high and low entropic level users are discussed.

Chapter 5

Experiments & Results

5.1 Introduction

This chapter presents the experiments performed to predict the future locations of individuals using machine learning algorithms. Initially, series of experiments are carried out to predict the future location using different supervised machine learning algorithms. These algorithms are varied with methods and types of learning algorithms that changes the performance of each of these machine learning algorithms. This is also discussed in this chapter. For each experiments, the methodology, the results obtained and discussion on the accuracy of that method in predicting location is also presented. In the next section, finding out the accuracy of predicting future locations depending on the entropy level of an individual is discussed.

5.2 Naïve Bayes

To apply Naïve Bayes on training data, classifier available in MatLab named “Naïve Bayes classifier” is used. A naïve Bayes classifier assigns a new observation to the most probable class, assuming the features are conditionally independent given the class value. A “fit” method was used to create naïve Bayes classifier object by fitting the training data. Once the object is created, the method “predict” was used to predict the class label for testing datasets. The following syntax shows how to build a naïve Bayes classifier object in MatLab.

nb = NaïveBayes.fit (training, class, distribution, distribution type, …)

Here, ‘nb’ defines the naïve Bayes classifier object that is trained by using training dataset of numeric matrix. In the Reality mining dataset, rows of training data represents the observations and its columns represents the features. The labels in the Reality Mining dataset is a class that classifies the variable for training dataset. Each elements in class defines which label the corresponding row of training belongs to. To successfully run the naïve Bayes classifier in MatLab, dataset in “training” must have the same number of rows as in “class” dataset. The other parameters in syntax is a distribution parameter that is followed with type of distribution to be used to train the naïve Bayes classifier. The available types of distribution in MatLab are Normal Gaussian Distribution, Kernel Distribution, Multivariate Multinomial Distribution for discrete data, and Multinomial distribution for classifying the count-based data such as the bag-of-tokens model. The following column chart shows the accuracy performance on predicting the location on the Reality Mining dataset.

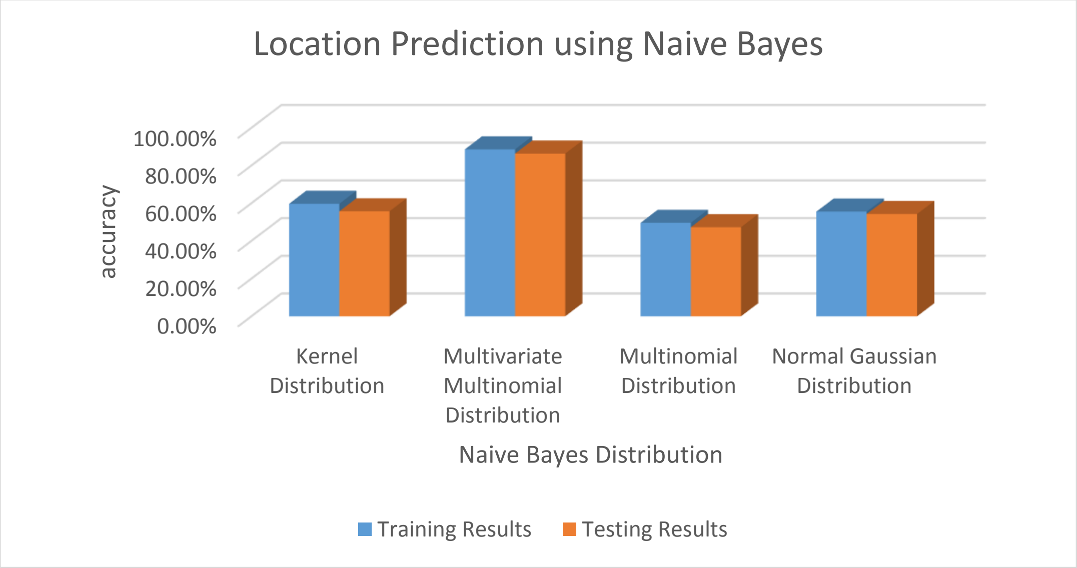


Figure 9: Naïve Bayes Results on location prediction

5.2.1 Discussion

From the above Figure 9, it can be said that Multivariate Multinomial Distribution for discrete data performs better on predicting location acquiring accuracy of 86.31% on testing datasets. This may be because the observations on dataset is categorical and the multivariate multinomial distribution is an appropriate prediction for such types of observations. Using this distribution, Naïve Bayes classifier records a separate set of distinct predictor levels for each predictor. It also computes a separate set of probabilities for the set of predictor levels for each class. This classification assumes each individual feature follows a multinomial model within a class. The parameters for a feature include the probabilities of all possible values that the corresponding feature can take.

5.3 Classification Trees

To classify the future locations, Classification Trees are also applied as a predictor on the Reality Mining dataset. Classification Trees can handle both categorical and numerical data. It builds classification models in the form of a tree structure that breaks down the dataset into smaller subsets developing associated further decision tree. The final result is a tree with a decision nodes and leaf nodes. Classification Trees from MatLab tool are used as decision trees that classifies the data based on the input and output data provided for this experiment. To predict the location, a training dataset with input and output variables are provided to create a classification tree using “classregtree”. The following syntax shows how to build a classification trees in MatLab.

t = classregtree (X, y, ‘Name’, value)

Here, “t” represents a decision tree that predicts the class “y” using a classification function “classregtree” on the observations values X. Here, class “y” is a categorical value that represents the location of an individual as “Home”, “Work” and “Elsewhere” whereas X, a matrix of n by m, represents the observation values or predictor values (days, hours, persons, and cell towers) that are recorded from individual’s mobile phone. As class “y” is a categorical value, the function “classregtree” performs classification problem on provided data. The other parameters ‘Name’ and ‘value’ are optional parameters that take name and value pairs. For this experiment, the name is a ‘method’ and its value is ‘classification’ as the experiment’s motive is to predict the locations.

The following Figure 10 and Figure 11 below shows the tree structure built using classification trees on training set of the Reality Mining dataset. Here, each node represents a single variable and each branch represents a value that variable can take. To make the prediction of location, the classification tree chooses the variable that is at the top of the tree and creates a branch for each possible value. This process continues for each branch until it reaches to a leaf node, which denotes the location to be classified.

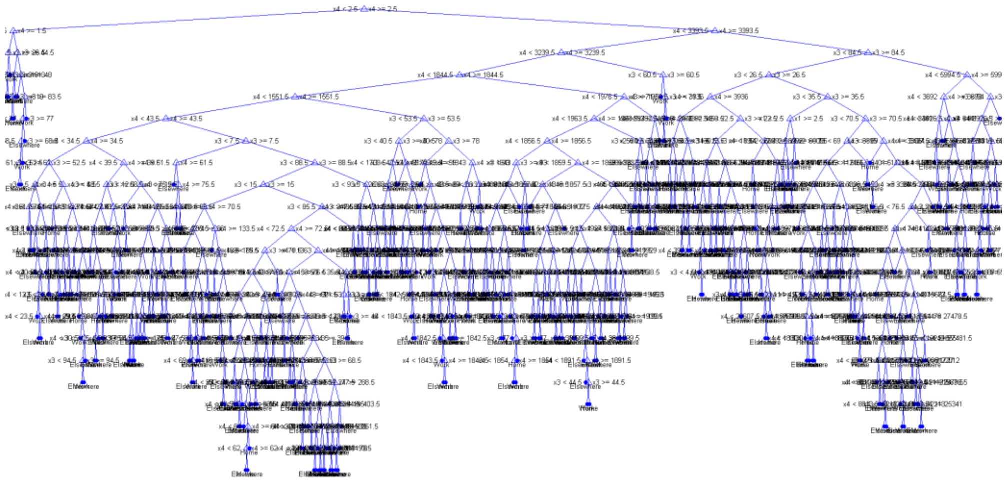


Figure 10: Classification Trees classifying locations

(See Appendix B for larger figure)

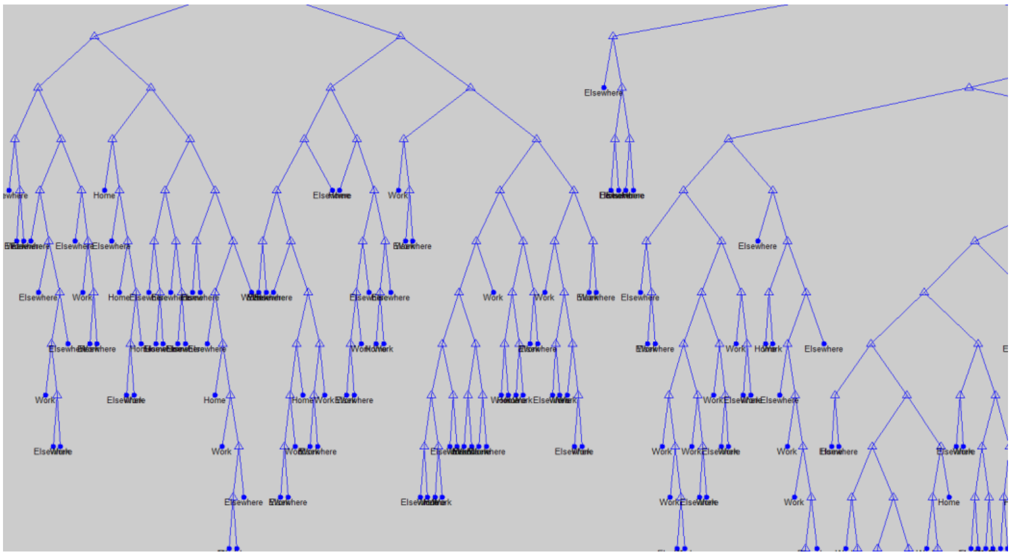


Figure 11: Classification Trees with Leaf Nodes (only)

(See Appendix B for larger figure)

435407	0	3
1	233583	0
0	2	185942

Table 7: Confusion Matrix of Classification Trees

5.3.1 Discussion:

The confusion matrix of classification trees shows that it achieves the highest predicting accuracy for location. The accuracy rate for classification trees is 99.99% for training set and 97.90% for testing set. This rate of accuracy for location prediction by the use of classification trees assures that this machine learning technique is one of the best performing algorithm for location prediction.

5.4. Artificial Neural Networks

Neural Networks are good at pattern recognition and classification. Neural Network toolbox in MatLab provide tools for designing, implementing, visualizing and simulating neural network for classification, clustering, feature mining, prediction and pattern recognition (Kadu & Dhande 2012). To define a pattern recognition problem in Reality Mining dataset, Neural Pattern Recognition tool (nprtool) of MatLab has been used in this thesis. This tool comprises of feed forward neural network. In this network, the information moves in only one direction, forward, from the input node through the hidden node to the output node. This network is mainly used in the area of pattern recognition and prediction. To implement this network in the Reality mining dataset, the training and testing dataset are arranged in a set of input vectors as columns in a matrix. Then another set of target vectors are also arranged as columns in a matrix to indicate the output to which the input vectors are assigned. The output of any location data is either Work, Elsewhere or Home. In order to introduce these three terms into the network, these labels are converted to three element vectors of 1 and 0. The characters '1 0 0' is assigned to target value, 'Work', and '0 1 0' to 'Elsewhere' and '0 0 1' to Home' in both the training and testing output set. So for each set of 4 input entries, there is an output set of three entries. In the case of location being at work, the three entries will be '1 0 0' and so on.

To create a network, a pattern recognition network is defined with a hidden layer, a training algorithm and input/output data divided into three sets: 70% are used for training network and 15% are used for validation that the network is generalizing and to stop the training before over-fitting occurs. The last 15% are used as a completely independent test of network generalization. In this network, the hidden layer sizes are varied to be the 5 to 20. However, it is found that defining size of 20 hidden layer performs best for most of the training algorithms used to train the algorithm. Altogether, 8 different training algorithms are applied to the Reality Mining dataset in order to find the best performing training algorithm in feed forward neural network. Therefore the network created is a feed forward neural network with 4 input entries, 20 hidden layers, and 3 output entries trained with 8 different training algorithms.

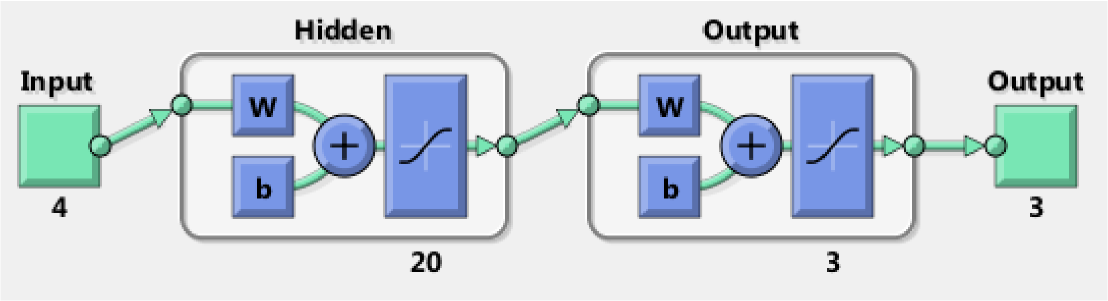


Figure 12: Neural Network Architecture

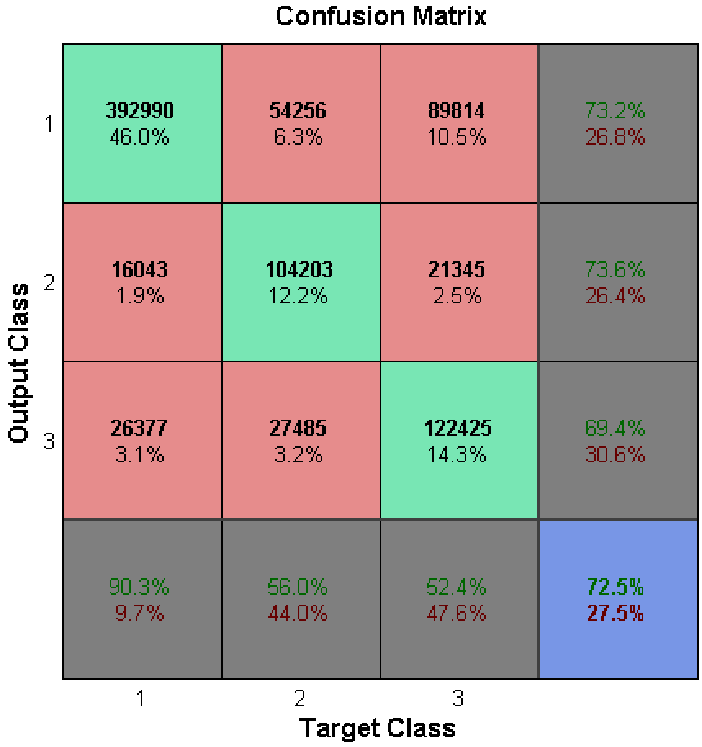
Confusion matrices are typically used for validating pattern recognition applications. The training, validation and testing are altogether represented in confusion matrix. Each row of the matrix represents the cases in a predicted class, while each column represents the cases in an actual class. One benefit of a confusion matrix is that it is easy to see if the developed network is confusing three classes. Considering the test confusion matrix, the developed network can be said if it is trained well with good accuracy performance. The following Table 8 shows how each of the training algorithms performed.

Description	**Algorithms**	**Hidden Layer size**	**Accuracy**	**MSE**
Scaled Conjugate Gradient	trainscg	20	62.60%	0.18241
Conjugate Gradient with Powell/Beale Restarts	traincgb	20	63.30%	0.16447
Fletcher-Powell Conjugate Gradient	traincgf	20	58.30%	0.18602
Polak-Ribiére Conjugate Gradient	traincgp	20	60.40%	0.17
One Step Secant	trainoss	20	59.60%	0.1767
Variable Learning Rate Backpropagation	traingdx	20	55.30%	0.19017
Levenberg-Marquardt	trainlm	20	72.50%	0.13767
BFGS Quasi-Newton	trainbfg	20	63.70%	0.16591
Resilient Backpropagation	trainrp	20	61.20%	0.17377

Table 8: Performance of different training algorithms in Neural Network

5.3.1 Discussion:

From the above table, it can be said that the training algorithm Levenberg-Marquardt has performed well acquiring accuracy of 72.50% on the dataset. The confusion matrix produced by this network is as follows:



Output Class \ Target Class	1	2	3	
1	392990 46.0%	54256 6.3%	89814 10.5%	73.2% 26.8%
2	16043 1.9%	104203 12.2%	21345 2.5%	73.6% 26.4%
3	26377 3.1%	27485 3.2%	122425 14.3%	69.4% 30.6%
	90.3% 9.7%	56.0% 44.0%	52.4% 47.6%	72.5% 27.5%

Figure 13: Confusion Matrix of Levenberg-Marquardt on the Reality Mining dataset

5.5 Summary

For location prediction, the machine learning algorithms, naïve Bayes, Classification Trees (Decision Trees), and MLP are applied to the Reality Mining Dataset. The following chart shows the performance of these algorithms for predicting future locations of an individual.

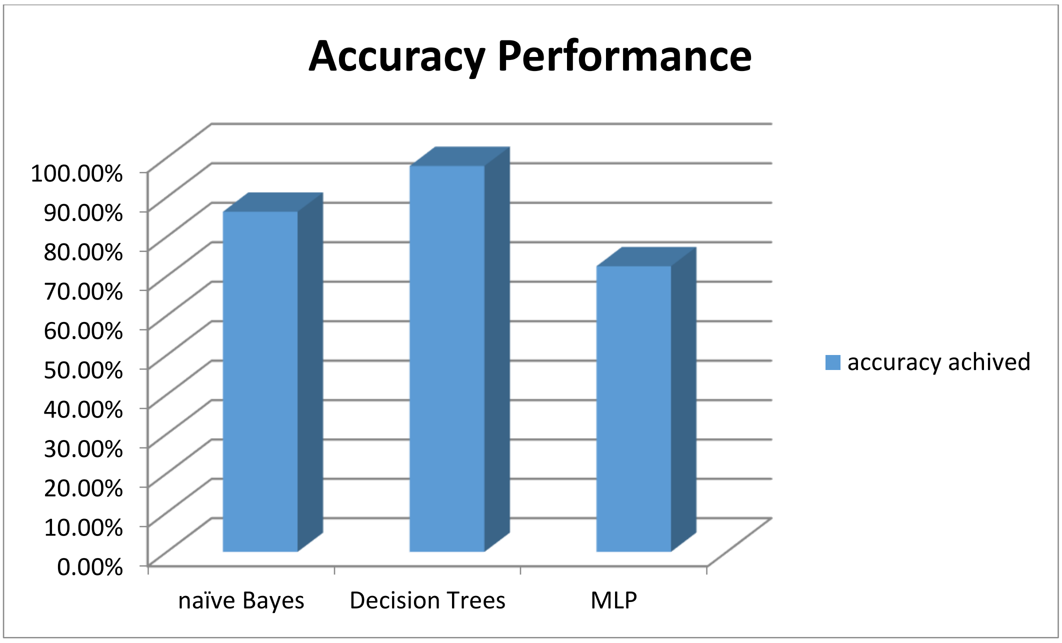


Figure 14: Accuracy performance of Machine Learning algorithms

From the above chart, it can be clearly said that Decision trees (Classification Trees) perform better than naïve Bayes and MLP algorithms. Though, MLP has secured 72.50% on accuracy, Naïve Bayes and Decision Trees (Classification Trees) have surpassed this performance by achieving 86.50% and 97.60% respectively. Therefore, Decision Trees (Classification Trees) are best performing algorithms for predicting future location.

In appendix A, the MatLab Codes used to process all the experiments are presented. With this thesis, the attached CD provides all the data (formatted) to run these experiments. The provided resources include: MatLab Codes for all machine learning algorithms, Dataset for Naïve Bayes, Classification Trees and Multilayer Perceptron. This also includes the resources for Entropy Experiments in similar manner.

5.6 Entropy Experiments

Entropy is the randomness level of outcome. In this experiment, the machine learning algorithms are applied to the dataset of users with the highest and lowest entropy level to predict the future location. The main idea of this is to find if the entropy level has any affect to the performance of machine learning algorithms to be applied. As from calculation made for entropy in Section 4.6, User 78 and User 93 are presented as the highest and lowest entropy level respectively in this experiment. The machine learning algorithms applied to entropic users dataset are processed in similar manner to predicting future location.

5.6.1 Machine Learning algorithm on Entropy Users

The initial machine learning algorithms applied was a Naïve Bayes classifier on both the low and high entropy users. This algorithm secured are 98.11% on low entropy level 'User 93' and 99.63% on high entropy level 'User 78'. The confusion matrix acquired from Naïve Bayes algorithm applied to high entropy level User 78 for its testing set is as follows:

1994	0	30
0	5144	0
0	0	1013

Table 9: Confusion Matrix acquired from Naïve Bayes applied to User 78

Similarly, Classification Trees were also applied to low and high entropy level users. It performed well to achieve the highest accuracy amongst other machine learning algorithms on both the dataset of low and high entropic users. The accuracy achieved using classification trees are 99.62% for high entropic user and 99.25% for low entropic user. The following Figure 15 and Figure 16 shows the Classification Trees for high and low entropic users.

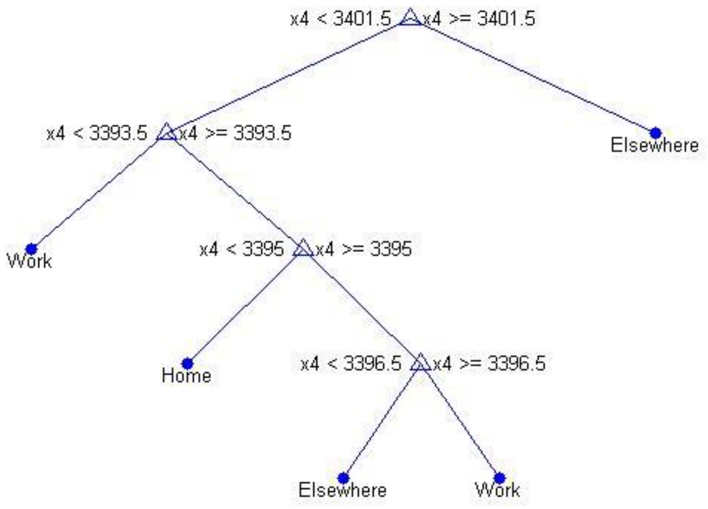


Figure 15: Classification Trees for High Entropy User 78

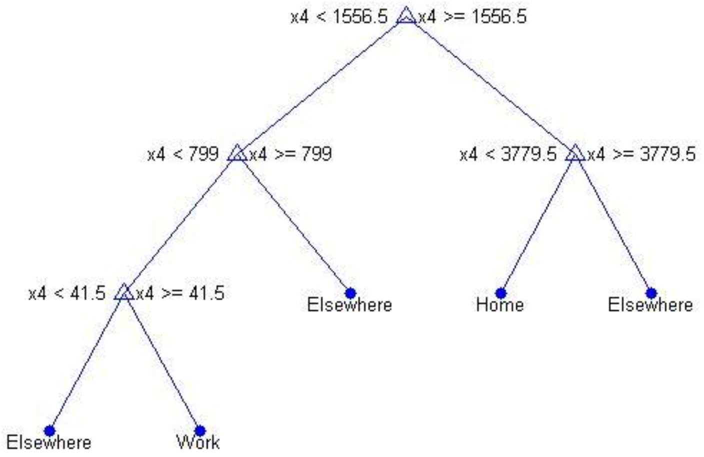
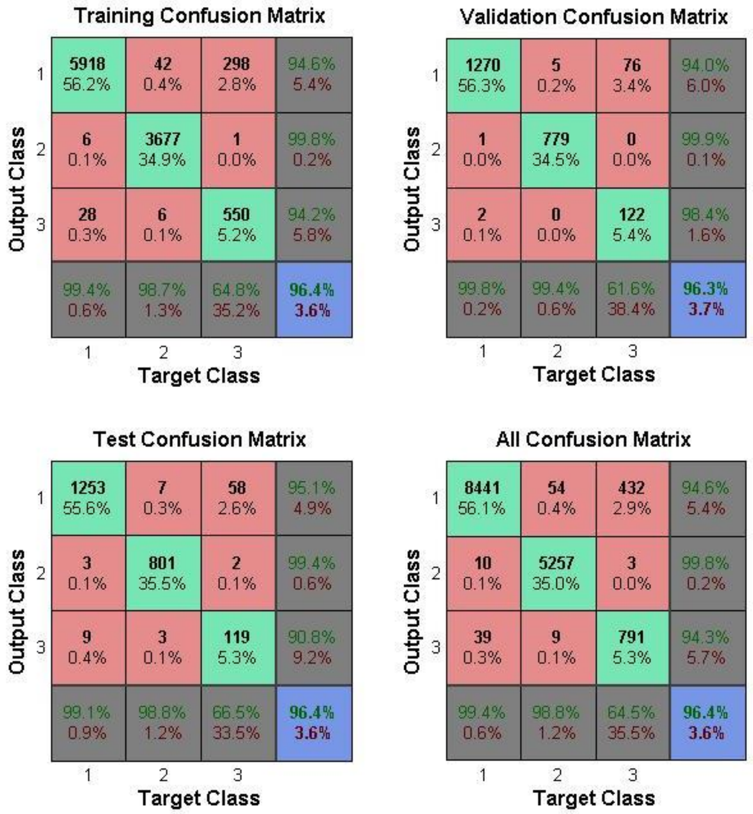


Figure 16: Classification Trees for Low Entropy User 93

Finally, Multilayer perceptron is applied to both the datasets. MLP on high entropic user performed well with the overall accuracy of 96.4% with the mean squared error of 0.1371. The confusion matrix and mean squared error (MSE) acquired on a high entropic user is as follows:



Training Confusion Matrix

Output Class	1	2	3	
1	5918 56.2%	42 0.4%	298 2.8%	94.6% 5.4%
2	6 0.1%	3677 34.9%	1 0.0%	99.8% 0.2%
3	28 0.3%	6 0.1%	550 5.2%	94.2% 5.8%
	99.4% 0.6%	98.7% 1.3%	64.8% 35.2%	96.4% 3.6%

Target Class

Validation Confusion Matrix

Output Class	1	2	3	
1	1270 56.3%	5 0.2%	76 3.4%	94.0% 6.0%
2	1 0.0%	779 34.5%	0 0.0%	99.9% 0.1%
3	2 0.1%	0 0.0%	122 5.4%	98.4% 1.6%
	99.8% 0.2%	99.4% 0.6%	61.6% 38.4%	96.3% 3.7%

Target Class

Test Confusion Matrix

Output Class	1	2	3	
1	1253 55.6%	7 0.3%	58 2.6%	95.1% 4.9%
2	3 0.1%	801 35.5%	2 0.1%	99.4% 0.6%
3	9 0.4%	3 0.1%	119 5.3%	90.8% 9.2%
	99.1% 0.9%	98.8% 1.2%	66.5% 33.5%	96.4% 3.6%

Target Class

All Confusion Matrix

Output Class	1	2	3	
1	8441 56.1%	54 0.4%	432 2.9%	94.6% 5.4%
2	10 0.1%	5257 35.0%	3 0.0%	99.8% 0.2%
3	39 0.3%	9 0.1%	791 5.3%	94.3% 5.7%
	99.4% 0.6%	98.8% 1.2%	64.5% 35.5%	96.4% 3.6%

Target Class

Figure 17: Confusion Matrix acquired from MLP for High Entropic User 78

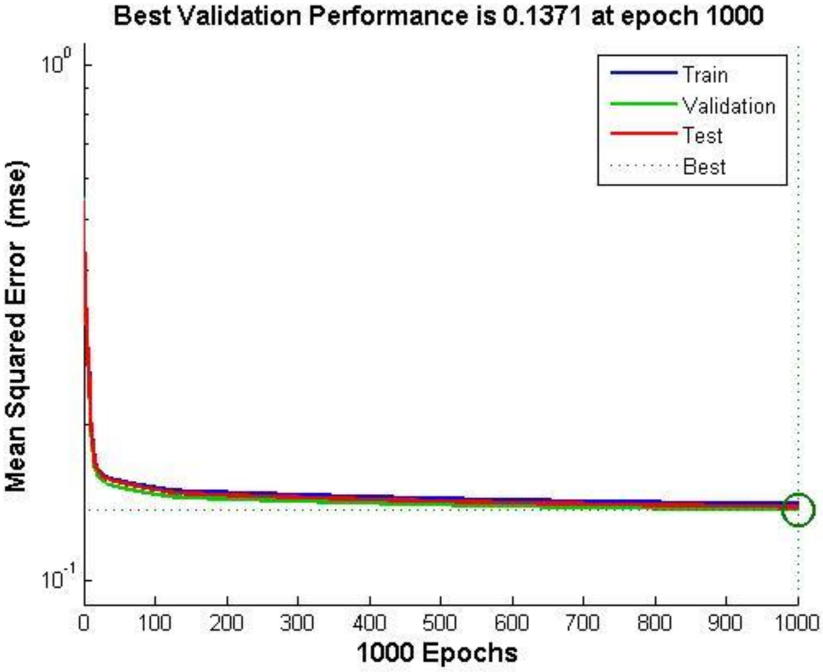


Figure 18: MSE acquired from MLP for High Entropic User 78

MLP on the dataset of low entropic user 93 acquired only 51.4% accuracy with the MSE of 0.17001. This result from MLP on mobile dataset is the lowest accuracy achieved. The following figures show the confusion matrix and MSE for a low entropic user.

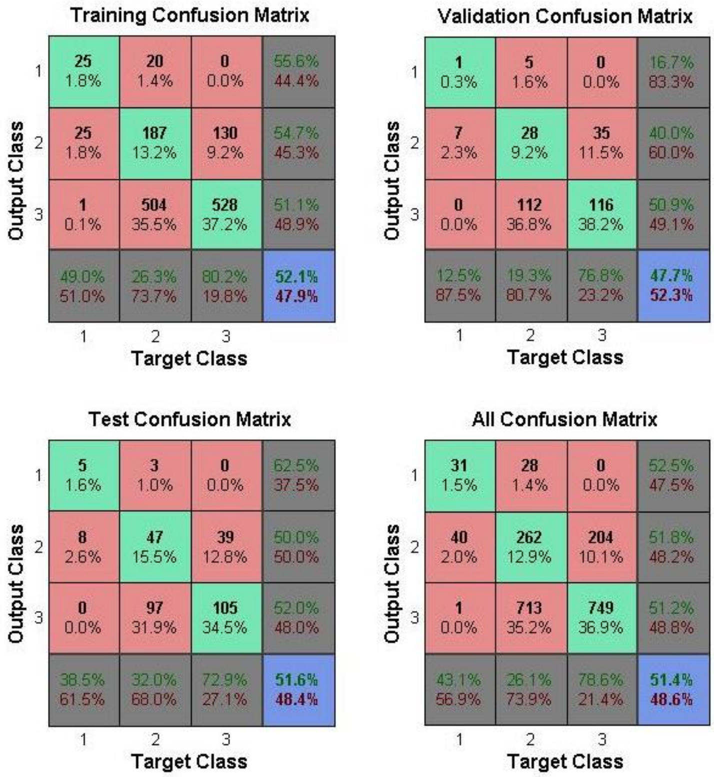


Figure 19: Confusion Matrix acquired from MLP for Low Entropic User 93

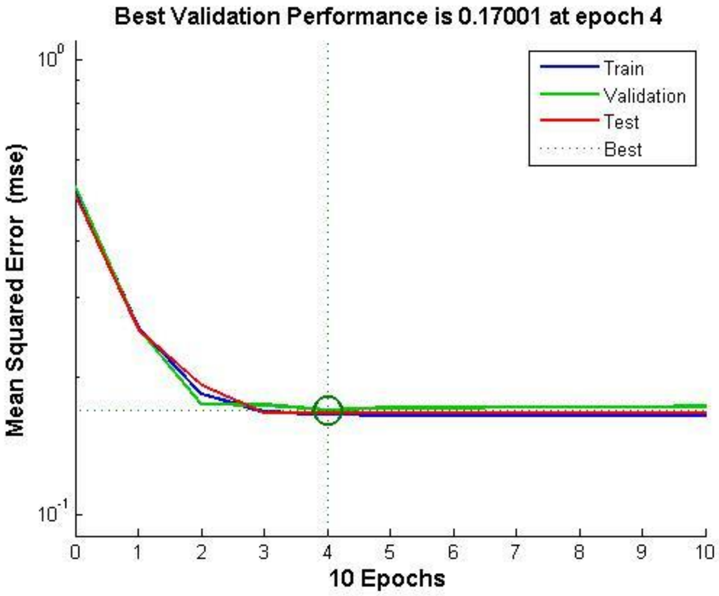


Figure 20: MSE acquired from MLP for High Entropic User 78

5.6.2 Discussion on Entropy Results

The results for the performance of machine learning algorithms on low entropic users was expected to be higher than high entropic users as low entropic users show the regularity in their user movement patterns. However, this expectation is demonstrated to be incorrect as the results attained from each machine learning algorithms were performing well for high entropic users in compare to low entropic users.

Despite the expected results for entropy, Classification Trees shows the overall best performance in predicting future locations for the users. In contrast to classification trees, the Multilayer Perceptron algorithm didn’t seem to perform well on the dataset for low entropic users.

Chapter 6

Discussion & Conclusions

The primary aim of this thesis was to develop a location predictive model by applying Reality Mining techniques which would incorporate the use of machine learning algorithms. As described in Chapter 2, Reality Mining techniques have been applied to various verticals for activity recognition, and finding patterns in human movements and relationships. In this thesis, it was hoped that the application of various machine learning algorithms to the field of Reality Mining techniques would improve the predictive models for predicting future locations of individuals.

A number of literatures on Reality Mining and Machine Learning algorithms were reviewed. The Reality Mining publications by Alex Pentland and Nathan Eagle were taken as the main priority for this research. This was, in large part, due to the fact that the Reality Mining dataset was made available and was the dataset used by this project. Other literature reviews included the papers on activity recognitions, inferring movement patterns and location predictions with (Choudhury & Pentland 2002; Huynh 2007) on activity recognition, (Eagle, Quinn, et al. 2009; Azam et al. 2012; Vukovic et al. 2009; Hill et al. 2010) on inferring movement patterns and location predictions particularly relevant.

For the experimental purposes on this thesis, the datasets were requested from Nathan Eagle. Then the pre-processing of datasets took place to convert the original format of the Reality Mining dataset into certain formats as described in Chapter 4. This pre-processing is required due to the absence of certain desired features in datasets. These features were transformed into particular formats to enable the application of the chosen machine learning algorithms.

Appendix A in this project contains all the data used for the experiments on Reality Mining. It includes the codes used to run the algorithms, the pre-processed dataset in CSV and particularly formatted dataset in MatLab for different machine learning algorithms used.

From the results obtained from Machine Learning methods applied on different representations of the Reality Mining dataset to predict the future location of an individual, it is found that Classification Trees perform best with less errors for location prediction. On the other hand, though the machine learning algorithms, Naïve Bayes and MLP perform well for location prediction, they could have performed better if the parameters were significantly modified. The models built using Classification Trees for location prediction is close to 100% accuracy when tested with unseen data for a given individual present in the training set outlined in Section 4.4 of Chapter 4.

In this thesis, Classification Trees applied to the location data of Reality Mining experiment have excellent results for predicting the location activity of an individual at a given time as either Home, Work or Elsewhere. These results obtained are a significant improvement on previous work conducted by (Eagle, Quinn, et al. 2009), where they reported that a dynamic Bayesian network was used to predict the subject's next location and reached very high accuracy of over 90%. Similarly, as described in Section 2.6.2 of Chapter 2, the authors in (Gambs et al. 2012) presented an algorithm for next place prediction based on a mobility model of an individual called a n-MMC that keeps track of the n-previous locations visited. They applied this algorithm on three different datasets and have shown that the accuracy of this algorithm ranges from 70% to 95%.

The predictive model generated in this thesis may be applied to develop various vertical applications such as user profiling that keeps track of the daily movement activities for a certain range of times. This application will be helpful for arranging meetings with co-workers or making plans for gatherings with colleagues. Another feasibility of the application may be to use it as a monitoring tool at workplaces that sends an alert to managers if the staff are found out to have not arrived at a given location when they were expected.

The entropy experiment conducted with the dataset from high and low entropic users gave opposite results than expected. It was expected that the high entropic users are hard to predict and will have less accuracy in predicting future location. However, it did not meet the expectations and instead performed better for high entropic users rather than for low entropic users. The reason for this is not fully understood and requires further study.

6.1 Recommendations for Further Research

There are number of ways in which this thesis could be complemented by further research. One of the possible areas would be the expansion of Reality Mining techniques on a new set of data. The dataset used in this thesis is nearly a decade old. New datasets are continuously becoming available (Bell et al. 2011; Laurila et al. 2012). The machine learning algorithms selected in this thesis can also be applied in similar manner to new datasets and used to analyse the performance of predictive models built.

The Classification Trees produced remarkable results when trained on six months of data. It would be interesting to examine the performance of the Classification Trees when reducing the amount of training data, in particular how much data would be required as a minimum for a given desired accuracy.

As stated earlier, the results of Entropy experiments could also be analysed to find out the reason behind such contrasting results.

Another area could be to further refine the classifications used to predict the future movements of an individual at a given time. In this thesis, the locations are labelled as either Home, Work or Elsewhere. Possible refinements could include place labels such as travelling to work or driving home etc.

Additionally, (Xiong et al. 2012; Farrahi & Gatica-Perez 2008) have shown that relationships between people could be inferred from these datasets. If a relationship is identified between an unknown individual and an individual known to a predictive model, can the model be used to predict the location of the unknown individual?

Bibliography & References

Alpaydin, E., 2010. *Introduction to Machine Learning* Second., Cambridge, Massachusetts London, England: The MIT Press. Available at: http://www.realtechsupport.org/UB/MRIII/papers/MachineLearning/Alppaydin_MachineLearning_2010.pdf.

Ashbrook, D. & Starner, T., 2002. Learning significant locations and predicting user movement with GPS. In *Sixth International Symposium on Wearable Computers, 2002. (ISWC 2002). Proceedings*. Sixth International Symposium on Wearable Computers, 2002. (ISWC 2002). Proceedings. pp. 101–108.

Azam, M.A. et al., 2012. Human behaviour analysis using data collected from mobile devices. *International Journal on Advances in Life Sciences*, 4(1 and 2), pp.1–10.

Barabasi, A.L. et al., 2002. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3-4), pp.590–614.

Bell, S., McDiarmid, A. & Irvine, J., 2011. Nodobo: Mobile phone as a software sensor for social network research. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, pp. 1–5. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5956319.

Bonato, P., 2010. Wearable sensors and systems. From enabling technology to clinical applications. *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society*, 29(3), pp.25–36.

Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F. & Kittler, J., 2010. *Advances in Pattern Recognition: Second Mexican Conference on Pattern Recognition, MCPR 2010, Puebla, Mexico, September 27-29, 2010, Proceedings*, Springer Science & Business Media.

Choudhury, T. & Pentland, A., 2002. *The Sociometer: A Wearable Device for Understanding Human Networks*,

Choujaa, D. & Dulay, N., 2009. Activity Recognition from Mobile Phone Data: State of the Art, Prospects and Open Problems.

Church, K. & Oliver, N., 2011. Understanding Mobile Web and Mobile Search Use in Today’s Dynamic Mobile Landscape. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI ’11. Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. New York, NY, USA: ACM, pp. 67–76. Available at: http://www.karenchurch.com/blog/wp-content/uploads/2012/02/mobilehciFullPaperCRC.pdf [Accessed October 2, 2014].

Clifton, C., 2014. data mining | computer science. *Encyclopedia Britannica*. Available at: http://www.britannica.com/EBchecked/topic/1056150/data-mining [Accessed January 3, 2015].

Davis, C., 2014. Robots That Work Autonomously Being Developed by DARPA. *Guardian Liberty Voice*. Available at: http://guardianlv.com/2014/03/robots-that-work-autonomously-being-developed-by-darpa/ [Accessed September 2, 2014].

Dayan, P., 1999. Unsupervised Learning. *The MIT Encyclopedia of the Cognitive Sciences*. Available at: http://www.gatsby.ucl.ac.uk/~dayan/papers/dun99b.pdf.

DeGusta, M., 2012. Are Smart Phones Spreading Faster than Any Technology in Human History? *MIT Technology Review*. Available at: http://www.technologyreview.com/news/427787/are-smart-phones-spreading-faster-than-any-technology-in-human-history/ [Accessed January 5, 2015].

Dong, W., Lepri, B. & Pentland, A. (Sandy), 2011. Modeling the Co -evol ution of Behaviors and Social Relationships Using Mobile Phone Data. *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*.

Eagle, N., 2010. Mobile Phones as Social Sensors. In *The Handbook of Emergent Technologies in Social Research*. Oxford University Press.

Eagle, N. & Pentland, A., 2005. Social serendipity: mobilizing social software. *IEEE Pervasive Computing*, 4(2), pp.28–34.

Eagle, N., Pentland, A. (Sandy) & Lazer, D., 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, p.pnas.0900282106.

Eagle, N., Quinn, J.A. & Clauset, A., 2009. Methodologies for Continuous Cellular Tower Data Analysis. In H. Tokuda et al., eds. *Pervasive Computing*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 342–353. Available at: http://link.springer.com/chapter/10.1007/978-3-642-01516-8_23 [Accessed October 20, 2014].

Eagle, N. & (Sandy) Pentland, A., 2006. Reality Mining: Sensing Complex Social Systems. *Personal Ubiquitous Comput.*, 10(4), pp.255–268.

Eaton, K., 2013. Dragon Dictation and Other Speech Recognition Apps - Review. *The New York Times*. Available at: http://www.nytimes.com/2013/04/18/technology/personaltech/dragon-dictation-and-other-speech-recognition-apps-review.html [Accessed September 2, 2014].

Farrahi, K. & Gatica-Perez, D., 2008. What Did You Do Today?: Discovering Daily Routines from Large-scale Mobile Data. In *Proceedings of the 16th ACM International Conference on Multimedia*. MM ’08. New York, NY, USA: ACM, pp. 849–852. Available at: http://doi.acm.org/10.1145/1459359.1459503 [Accessed January 5, 2015].

Freeman, L.C., Romney, K.A. & Freeman, S.C., 1987. Cognitive Structure and Informant Accuracy. *American Anthropological Association*, 89(2), pp.310–325.

FrontLine Systems, I., 2012. Classification Tree - Intro. *solver*. Available at: http://www.solver.com/classification-tree-intro [Accessed January 4, 2015].

Ben-Gal, I., 2008. Bayesian Networks. In *Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons, Ltd. Available at: http://onlinelibrary.wiley.com/doi/10.1002/9780470061572.eqr089/abstract [Accessed October 20, 2014].

Gambs, S., Killijian, M.-O. & del Prado Cortez, M.N., 2012. Next Place Prediction Using Mobility Markov Chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. MPM '12. New York, NY, USA: ACM, pp. 3:1–3:6. Available at: http://homepages.laas.fr/mkilliji/docs/workshops/MPM12.pdf [Accessed January 5, 2015].

Harris, M.C., 2011. *Artificial Intelligence*, Marshall Cavendish Benchmark, New York. Available at: http://books.google.ie/books?id=Cmf5cp4YBKMC&printsec=frontcover&dq=what+is+artificial+intelligence&hl=en&sa=X&ei=Lvm7U6OCLaPF7Aagj4CACg&ved=0CDYQ6AEwAQ#v=onepage&q=what%20is%20artificial%20intelligence&f=false.

Hesse-Biber, S.N., 2011. *The Handbook of Emergent Technologies in Social Research*, Oxford University Press.

Hill, S. et al., 2010. Hill 2010 realityminingafrica.pdf. Available at: http://www.cis.upenn.edu/~ngns/docs/References/Hill%202010%20realityminingafrica.pdf.

Huberman, B.A. & Adamic, L.A., 2003. Information Dynamics in the Networked World. in: Eli Ben-Naim, Hans Frauenfelder, Zoltan Toroczkai, (Eds.), "Complex Networks", Lecture Notes in Physics. *Springer*. Available at: http://www.hpl.hp.com/research/idl/papers/infodynamics/infodynamics.pdf.

Huynh, 2007. Scalable Recognition of Daily Activities with Wearable Sensors.

jasonb, A Tour of Machine Learning Algorithms | Machine Learning Mastery. Available at: http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ [Accessed August 8, 2014].

Kadu, S. & Dhande, S., 2012. Effective Data Mining Through Neural Network. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(3). Available at: http://www.ijarcsse.com/docs/papers/March2012/volume_2_Issue_3/V2I3009.pdf.

Kausik Majumdar, N.D., 2005. Mobile User Tracking Using A Hybrid Neural Network. *Wireless Networks*, 11, pp.275–284.

Lane, N.D. et al., 2010. A Survey of Mobile Phone Sensing. *IEEE Communications Magazine*, pp.140–150.

Larivière, B. & Van den Poel, D., 2004. Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: the case of financial services. *EXPERT SYSTEMS WITH APPLICATIONS*, 27(2), pp.277–285.

Laurila, J.K. et al., 2012. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*. Available at: http://infoscience.epfl.ch/record/192489 [Accessed February 11, 2015].

Liao, L. et al., 2007. Learning and Inferring Transportation Routines. *Artif. Intell.*, 171(5-6), pp.311–331.

Lovrek, I. & Sinkovic, V., 2004. Mobility Management for Personal Agents in the All-mobile Network. In M. G. Negoita, R. J. Howlett, & L. C. Jain, eds. *Knowledge-Based Intelligent Information and Engineering Systems*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1143–1149. Available at: http://link.springer.com/chapter/10.1007/978-3-540-30132-5_155 [Accessed January 5, 2015].

Michalski, R.S. et al., 1986. *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann.

Mitchell, T.M., 1997. *Machine Learning* 1st ed., New York, NY, USA: McGraw-Hill, Inc.

Negnevitsky, M., 2005. *Artificial Intelligence: A Guide to Intelligent Systems*, Pearson Education.

Nerenberg, J., China Looks to Control Traffic Congestion With Cell Phones. *Fast Company*. Available at: http://www.fastcompany.com/1733470/china-looks-control-traffic-congestion-cell-phones [Accessed August 28, 2014].

Networks, S., 2013. CitySense^{TM} | Sense Networks. Available at: https://www.sensenetworks.com/products/macrosense-technology-platform/citysense/ [Accessed January 5, 2015].

Oladipupo, T., 2010. Types of Machine Learning Algorithms. In Y. Zhang, ed. *New Advances in Machine Learning*. InTech. Available at: http://www.intechopen.com/books/new-advances-in-machine-learning/types-of-machine-learning-algorithms [Accessed August 8, 2014].

Osman, M., 2011. A Study of the Trend of Smartphone and its Usage Behavior in Malaysia. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1(2). Available at: http://sdiwc.net/digital-library/a-study-of-the-trend-of-smartphone-andits-usage-behavior-in-malaysia [Accessed October 2, 2014].

Page, L. et al., 1999. The PageRank Citation Ranking: Bringing Order to the Web. *Stanford InfoLab*. Available at: http://ilpubs.stanford.edu:8090/422/.

Pentland, A., 2004. “Reality Mining” the Organization. *MIT Technology Review*. Available at: http://www.technologyreview.com/news/402609/reality-mining-the-organization/ [Accessed January 5, 2015].

Polumetla, A., 2006. *Machine Learning Methods For The Detection Of RWIS Sensor Malfunctions*. UNIVERSITY OF MINNESOTA. Available at: http://www.d.umn.edu/~rmaclin/student-publications/aditya-ms.pdf.

Pramis, J., 2013. Number of mobile phones to exceed world population by 2014. *Digital Trends*. Available at: http://www.digitaltrends.com/mobile/mobile-phone-world-population-2014/ [Accessed January 3, 2015].

Quintero, A., 2005. A user pattern learning strategy for managing users' mobility in UMTS networks. *IEEE Transactions on Mobile Computing*, 4(6), pp.552–566.

Rabiner, L.R. & Juang, B.H., 1986. An introduction to hidden Markov models. *IEEE ASSp Magazine*.

Raento, M. et al., 2005. ContextPhone: a prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2), pp.51–59.

Raento, M., Oulasvirta, A. & Eagle, N., 2009. Smartphones: An Emerging Tool for Social Scientists. *Sociological Methods & Research*, 37(3), pp.426–454.

Ratti, C. et al., 2006. Mobile Landscapes: using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design*, 33(5), pp.727 – 748.

Roethlisberger, F.J. & Dickinson, W.J., 1939. *Management and the Worker: an account of a research program conducted by the Western electric company.*, Cambridge, MA.: Hardvard University Press.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323(6088), pp.533–536.

Russell, S.J. & Norvig, P., 2010. *Artificial Intelligence: A Modern Approach* Third., Prentice Hall. Available at: http://51lica.com/wp-content/uploads/2012/05/Artificial-Intelligence-A-Modern-Approach-3rd-Edition.pdf.

Schank, R.C., 1995. *Tell Me a Story: Narrative and Intelligence*, Northwestern University Press.

Song, C. et al., 2010. Modeling the scaling properties of human mobility. *Nature Physics*, 6(10), pp.818–823.

Steinbauer, M. & Kotsis, G., 2012. Building an Information System for Reality Mining Based on Communication Traces. In *2012 15th International Conference on Network-Based Information Systems (NBiS)*. 2012 15th International Conference on Network-Based Information Systems (NBiS). pp. 306–310.

Talbot, D., 2013. Researchers Use Data from Cheap Cell Phones in the Developing World to Combat Disease Outbreaks. *MIT Technology Review*. Available at: http://www.technologyreview.com/featuredstory/513721/big-data-from-cheap-phones/ [Accessed January 5, 2015].

Urmson, C., 2014. Just press go: designing a self-driving vehicle. *Official Google Blog*. Available at: http://googleblog.blogspot.com/2014/05/just-press-go-designing-self-driving.html [Accessed September 2, 2014].

Vázquez, A. et al., 2006. Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73(3), p.036127.

Vidal, J.M., n.d. Sigmoid Unit. *Artificial Neural Networks*. Available at: http://jmvidal.cse.sc.edu/talks/ann/sigmoid.html [Accessed March 9, 2015].

Vukovic, M., Lovrek, I. & Jevtic, D., 2007. Predicting user movement for advanced location-aware services. In *15th International Conference on Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007*. 15th International Conference on Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. pp. 1–5.

Vukovic, M., Vujnovic, G. & Grubisic, D., 2009. Adaptive user movement prediction for advanced location-aware services. In *17th International Conference on Software, Telecommunications Computer Networks, 2009. SoftCOM 2009*. 17th International Conference on Software, Telecommunications Computer Networks, 2009. SoftCOM 2009. pp. 343–347.

Williams, N.E. et al., 2014. Measures of Human Mobility Using Mobile Phone Records Enhanced with GIS Data. *arXiv:1408.5420 [physics, stat]*. Available at: http://arxiv.org/abs/1408.5420 [Accessed October 2, 2014].

Witten, I.H. & Frank, E., 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition* 2 edition., Amsterdam ; Boston, MA: Morgan Kaufmann.

Xiong, H. et al., 2012. Predicting Mobile Phone User Locations by Exploiting Collective Behavioral Patterns. In *2012 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC)*. 2012 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC). pp. 164–171.

Yang, S.-I. & Cho, S.-B., 2008. Recognizing human activities from accelerometer and physiological sensors. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008*. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. pp. 100–105.

Appendix A

Machine Learning MatLab Codes

MatLab codes for Classification Trees

ClassificationTrees.m (Classification Trees)

```
% import training input data as column vectors, giving column names
"day",
% "hour", "person_id" , "celltower_id"
% import training output data as column vector giving column name
"label"

%# construct predicting attributes and target class
vars={'day' 'hour' 'person_id' 'celltower_id'};
label=char(label);
x=[day hour person_id celltower_id];
y=cellstr(label);
%x=transpose(x);
%y=transpose(y);
%t=classregtree(x,y, 'method', 'classification', 'names', vars,
'categorical', [2 4], 'prune', 'off');
%t = classregtree(x, y, 'method','classification', 'names',vars, ...
%'categorical',[2 4], 'prune','off');
%view(t)
%t = classregtree(x, y, 'method','classification', 'names');
t = classregtree(x, y, 'method','classification');
view(t)
yPredicted=eval(t,x);

%cm =confusionmat(y,ypredicted);
cm =confusionmat(y,yPredicted);
accDecisionTrees = 100*sum(diag(cm))./sum(cm(:));
fprintf('Classifier1:\naccuracy = %.2f%%\n', accDecisionTrees);
fprintf('Confusion Matrix:\n'), disp(cm)

N=sum(cm(:));
%err=(N-sum(diag(cm))/N;
err=(N-sum(diag(cm)))/N;
tt=prune(t,'level',1);
view(tt)

% test unseen data

%inst=[2 9 70 41];
%prediction=eval(tt,inst)
%inst=[1 16 50 1842]; %114679
%prediction=eval(tt,inst)
%view(err)
%view(cm)


unseenData=[dayT hourT person_idT celltower_idT];
predictData=eval(tt,unseenData);

cm2=confusionmat(labelT,predictData);
accDecisionTrees2 = 100*sum(diag(cm2))./sum(cm2(:));
fprintf('Classifier1:\naccuracy = %.2f%%\n', accDecisionTrees2);
fprintf('Confusion Matrix:\n'), disp(cm2)
```

MatLab Codes for Naïve Bayes

NaiveBayes.m (Naïve Bayes)

```
%Initially import TrainingsetInputs as "matrix" with labels defined
(day, hour,
%person_id, celltower_id
%Import TrainingsetOutputs as "cellarray"


ord=randperm(size(TrainingsetInputs,1));
TrainMeas=TrainingsetInputs(ord,:);
classSpecs=TrainingsetOutputs(ord);
%scatter plot to check people at different towers indicating either
home,
%work or elsewhere
%gscatter(TrainMeas(:,3),TrainMeas(:,4),classSpecs,'rgb','osd');

%divide trainset and test set
train=TrainMeas(1:800000,:);
test=TrainMeas(800001:854938,:);
%training class labels
t_class=classSpecs(1:800000);

%test_class labels
tt_class=classSpecs(800001:854938);




%fit Naive Bayes on training set
%nb=NaiveBayes.fit(train,t_class,'Distribution','kernel'); % kernel
distribution
nb=NaiveBayes.fit(train,t_class,'Distribution','mvmn','prior','uniform
');%Multivariate multinomial distribution for discrete data % good so
far
%nb=NaiveBayes.fit(train, t_class, 'Distribution','mn');% Multinomial
distribution for classifying the count-based data such as the bag-of-
tokens model.
%nb=NaiveBayes.fit(train, t_class, 'Distribution','normal'); %normal
Gaussian Distribution
%prediction on testing set
y=nb.predict(test);
% confusion matrix
confMat=confusionmat(tt_class,y);

acc1 = 100*sum(diag(confMat))./sum(confMat(:));
fprintf('Classifier1:\naccuracy = %.2f%%\n', acc1);
fprintf('Confusion Matrix:\n'), disp(confMat)

%add 1st row
%frSum=sum(confMat(1,:));
%srSum=sum(confMat(2,:));
%trSum=sum(confMat(3,:));

%rowSum=frSum+srSum+trSum;
%diagSum=confMat(1,1)+confMat(2,2)+confMat(3,3);
```

```
%accuracy=(diagSum/rowSum)*100


%Test Unseen Data here
y=nb.predict(TestingSetInputs);
c=confusionmat(TestingSetOutputs,y);
accTest = 100*sum(diag(c))./sum(c(:));
fprintf('Classifier2:\naccuracy = %.2f%%\n', accTest);
fprintf('Confusion Matrix:\n'), disp(c)
```

LowEntropyNaiveBayes.m (Low Entropy Naïve Bayes)

```
%Initially import User93_LE_InpTrain as "matrix" with labels defined
(day, hour,
%person_id, celltower_id
%Import User93_LE_OutTrain as "cellarray"


ord=randperm(size(User93_LE_InpTrain,1));
TrainMeas=User93_LE_InpTrain(ord,:);
classSpecs=User93_LE_OutTrain(ord);
%scatter plot to check people at different towers indicating either
home,
%work or elsewhere
%gscatter(TrainMeas(:,3),TrainMeas(:,4),classSpecs,'rgb','osd');

%divide trainset and test set
train=TrainMeas(1:1500,:);
test=TrainMeas(1501:2028,:);
%training class labels
t_class=classSpecs(1:1500);

%test_class labels
tt_class=classSpecs(1501:2028);



%fit Naive Bayes on training set
%nb=NaiveBayes.fit(train,t_class,'Distribution','kernel'); % kernel
distribution
nb=NaiveBayes.fit(train,t_class,'Distribution','mvmn','prior','uniform
');%Multivariate multinomial distribution for discrete data % good so
far
%nb=NaiveBayes.fit(train, t_class, 'Distribution','mn');% Multinomial
distribution for classifying the count-based data such as the bag-of-
tokens model.
%nb=NaiveBayes.fit(train, t_class, 'Distribution','normal'); %normal
Gaussian Distribution
%prediction on testing set
y=nb.predict(test);
% confusion matrix
confMat=confusionmat(tt_class,y);

acc1 = 100*sum(diag(confMat))./sum(confMat(:));
fprintf('Classifier1:\naccuracy = %.2f%%\n', acc1);
fprintf('Confusion Matrix:\n'), disp(confMat)

%add 1st row
%frSum=sum(confMat(1,:));
%srSum=sum(confMat(2,:));
%trSum=sum(confMat(3,:));

%rowSum=frSum+srSum+trSum;
%diagSum=confMat(1,1)+confMat(2,2)+confMat(3,3);


%accuracy=(diagSum/rowSum)*100


%Test Unseen Data here
```

```
y=nb.predict(TestingSetInputs);
c=confusionmat(TestingSetOutputs,y);
accTest = 100*sum(diag(c))./sum(c(:));
fprintf('Classifier2:\naccuracy = %.2f%%\n', accTest);
fprintf('Confusion Matrix:\n'), disp(c)
```

HighEntropyNaiveBayes.m (High Entropy Naïve Bayes)

```
%Initially import TrainingsetInputs78 as "matrix" with labels defined
(day, hour,
%person_id, celltower_id
%Import TrainingsetOutputs78 as "cellarray"


ord=randperm(size(TrainingsetInputs78,1));
TrainMeas=TrainingsetInputs78(ord,:);
classSpecs=TrainingsetOutputs78(ord);
%scatter plot to check people at different towers indicating either
home,
%work or elsewhere
%gscatter(TrainMeas(:,3),TrainMeas(:,4),classSpecs,'rgb','osd');

%divide trainset and test set
train=TrainMeas(1:10000,:);
test=TrainMeas(10001:15036,:);
%training class labels
t_class=classSpecs(1:10000);

%test_class labels
tt_class=classSpecs(10001:15036);



%fit Naive Bayes on training set
%nb=NaiveBayes.fit(train,t_class,'Distribution','kernel'); % kernel
distribution
nb=NaiveBayes.fit(train,t_class,'Distribution','mvmn','prior','uniform
');%Multivariate multinomial distribution for discrete data % good so
far
%nb=NaiveBayes.fit(train, t_class, 'Distribution','mn');% Multinomial
distribution for classifying the count-based data such as the bag-of-
tokens model.
%nb=NaiveBayes.fit(train, t_class, 'Distribution','normal'); %normal
Gaussian Distribution
%prediction on testing set
y=nb.predict(test);
% confusion matrix
confMat=confusionmat(tt_class,y);

acc1 = 100*sum(diag(confMat))./sum(confMat(:));
fprintf('Classifier1:\naccuracy = %.2f%%\n', acc1);
fprintf('Confusion Matrix:\n'), disp(confMat)

%add 1st row
%frSum=sum(confMat(1,:));
%srSum=sum(confMat(2,:));
%trSum=sum(confMat(3,:));

%rowSum=frSum+srSum+trSum;
%diagSum=confMat(1,1)+confMat(2,2)+confMat(3,3);


%accuracy=(diagSum/rowSum)*100


%Test Unseen Data here
```

```
y=nb.predict(TestingsetInputs78);
c=confusionmat(TestingsetOutputs78,y);
accTest = 100*sum(diag(c))./sum(c(:));
fprintf('Classifier2:\naccuracy = %.2f%%\n', accTest);
fprintf('Confusion Matrix:\n'), disp(c)
```

MatLab codes for Multilayer Perceptron

MLPToolTranspose.m (MLP)

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
% Created Wed Nov 19 17:54:21 GMT 2014
%
% This script assumes these variables are defined:
%
% Data are required to be transposed and should be imported as matrix

%  TrainsetInpMLP - input data.
%   TrainsetOutMLP - target data.

inputs = TrainsetInpMLP;
targets = TrainsetOutMLP;

% Create a Pattern Recognition Network
hiddenLayerSize = 25;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm';  % Scaled conjugate gradient

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotconfusion', 'plotfit'};


% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)
```

```
% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(targets,outputs)
%figure, plotroc(targets,outputs)
%figure, ploterrhist(errors)
```

HighEMLP.m (High Entropy MLP)

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
% Created Tue Dec 23 13:37:17 GMT 2014
%
% This script assumes these variables are defined:
%
%   HETrainInput78 - input data.
%   HETrainOutput78 - target data.

inputs = HETrainInput78;
targets = HETrainOutput78;

% Create a Pattern Recognition Network
hiddenLayerSize = 20;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm';  % Scaled conjugate gradient

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotconfusion', 'plotfit'};


% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
```

```
testPerformance = perform(net,testTargets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(targets,outputs)
%figure, plotroc(targets,outputs)
%figure, ploterrhist(errors)
```

LowEMLP.m (Low Entropy MLP)

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
% Created Tue Dec 23 14:00:31 GMT 2014
%
% This script assumes these variables are defined:
%
%   LETrainInput93 - input data.
%   LETrainOutput93 - target data.

inputs = LETrainInput93;
targets = LETrainOutput93;

% Create a Pattern Recognition Network
hiddenLayerSize = 20;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainlm';  % Scaled conjugate gradient

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotconfusion', 'plotfit'};


% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
```

```
testPerformance = perform(net,testTargets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(targets,outputs)
%figure, plotroc(targets,outputs)
%figure, ploterrhist(errors)
```

Appendix B

Larger Figures

Figure 10: Classification Trees classifying locations (larger figure)

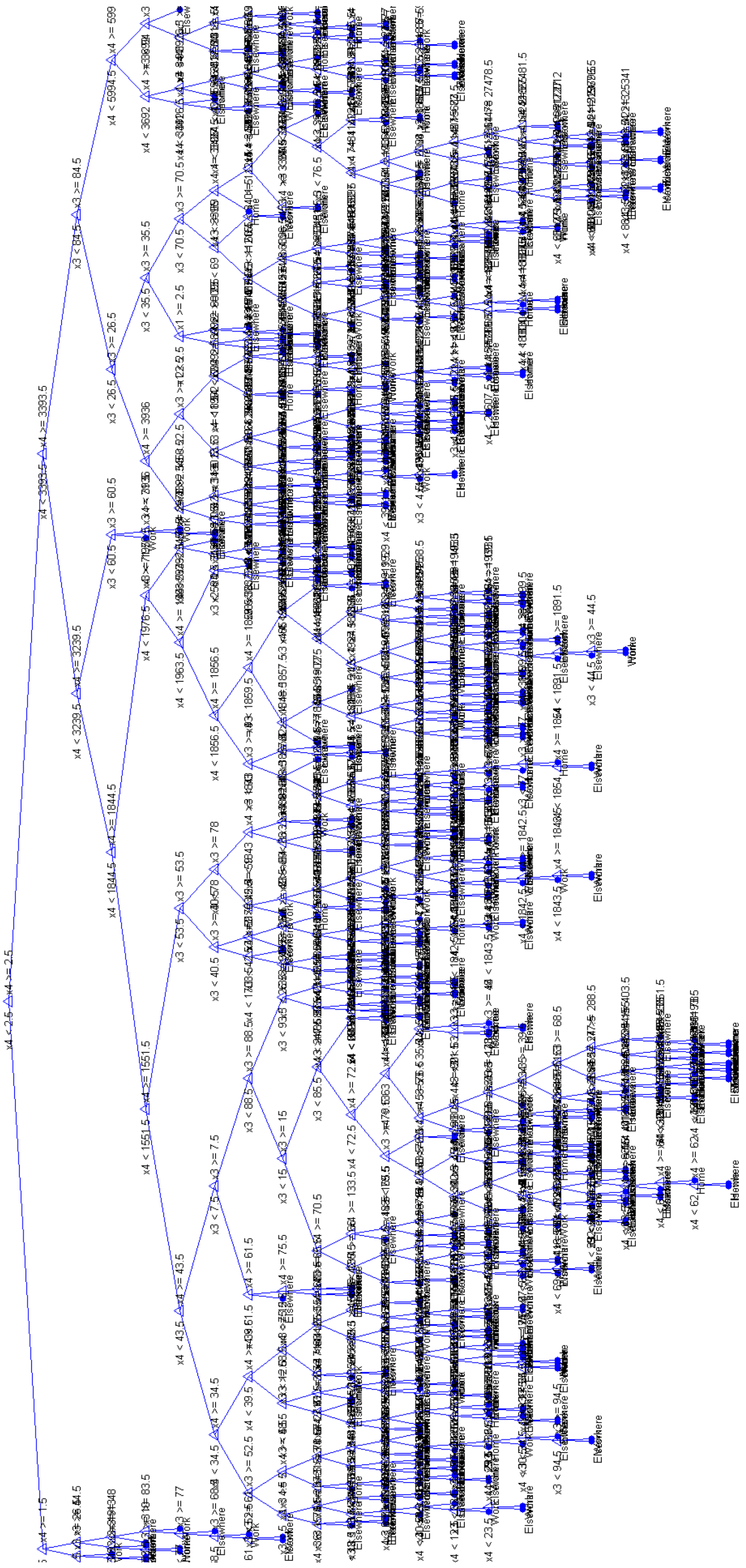
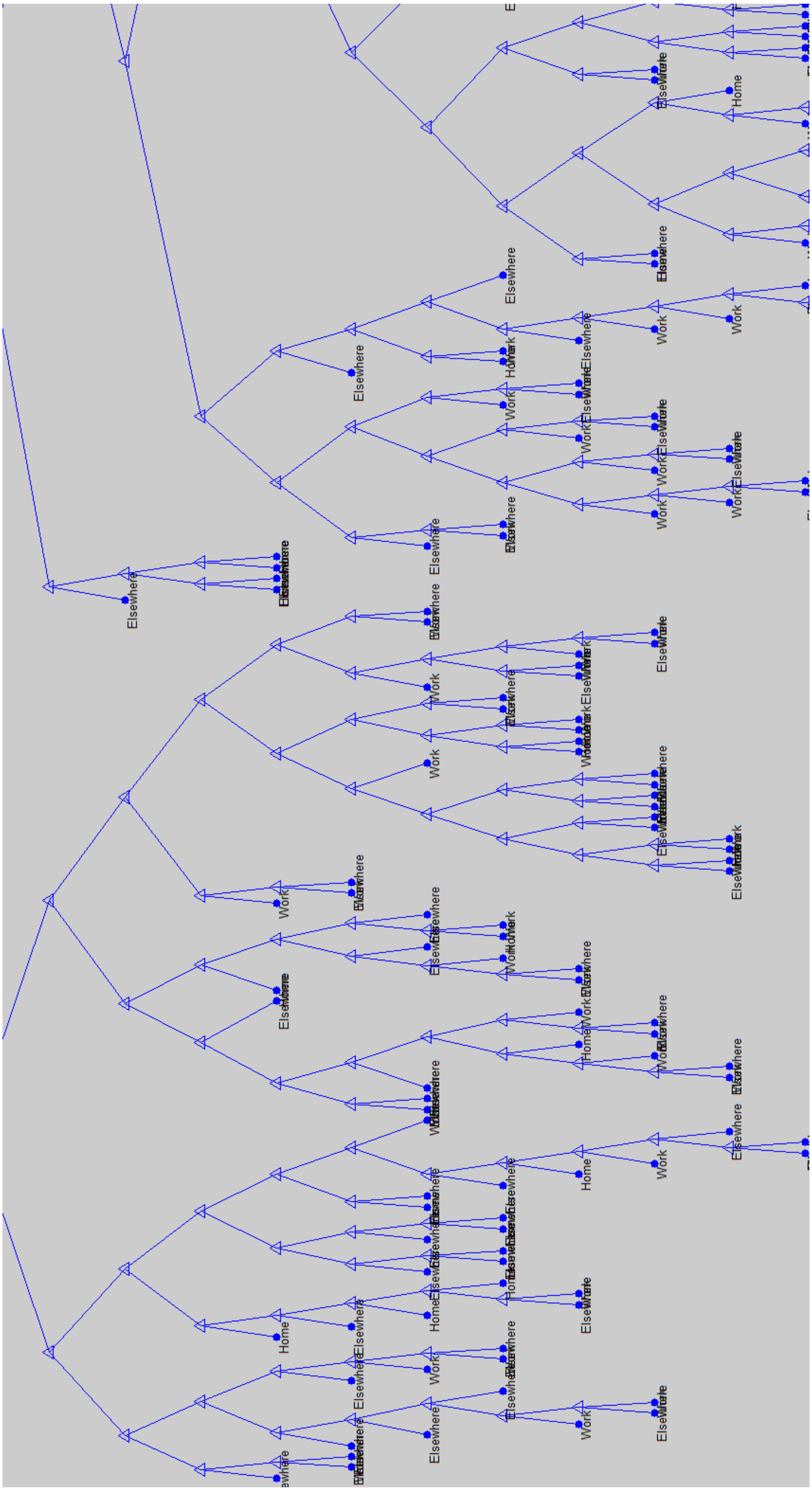


Figure 11: Classification Trees with Leaf Nodes (larger figure)


Elsewhere
Elsewhere
Home
Work
Work
Elsewhere
Elsewhere
Work
Work
Work
Work
Elsewhere
Elsewhere
Elsewhere
Work
Work
Work
Elsewhere
Work
Elsewhere
Home
Work
Elsewhere
Elsewhere
Elsewhere
Elsewhere
Elsewhere
Home
Work
Elsewhere
Home
Elsewhere
Elsewhere
Elsewhere
Elsewhere
Work
Elsewhere