INFERRING SOCIAL AND INTERNAL CONTEXT USING A MOBILE PHONE

Santi Phithakkitnukoon, B.S., M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2009

APPROVED:

Ram Dantu, Major Professor
Parthasarathy Guturu, Committee Member
Philip H. Sweany, Committee Member
Joao Cangussu, Committee Member
Ian Parberry, Interim Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Michael Monticino, Dean of the Robert B. Toulouse School of Graduate Studies

Phithakkitnukoon, Santi. Inferring Social and Internal Context Using a Mobile Phone. Doctor of Philosophy (Computer Science and Engineering), December 2009, 240 pp., 47 tables, 89 figures, references, 254 titles.

This dissertation is composed of research studies that contribute to three research areas including social context-aware computing, internal context-aware computing, and human behavioral data mining. In social context-aware computing, four studies were conducted. First, mobile phone user calling behavioral patterns are characterized in forms of randomness level where relationships among them are then identified. Next, a study was conducted to investigate the relationship between the calling behavior and organizational groups. Third, a method is presented to quantitatively define mobile social closeness and social groups, which are then used to identify social group sizes and scaling ratio. Last, based on the mobile social grouping framework, the significant role of social ties in communication patterns is revealed. In internal context-aware computing, two studies were conducted where the notions of internal context are intention and situation. For intentional context,the goal is to sense the intention of the user in placing calls. A model is thus presented for predicting future calls envisaged as a call predicted list (CPL), which makes use of call history to build a probabilistic model of calling behavior. As an incoming call predictor, CPL is a list of numbers/contacts that are the most likely to be the callers within the next hour(s), which is useful for scheduling and daily planning. As an outgoing call predictor, CPL is generated as a list of numbers/contacts that are the most likely to be dialed when the user attempts to make an outgoing call (e.g., by flipping open or unlocking the phone). This feature helps save time from having to search through a lengthy phone book. For situational context, a model is presented for sensing the user's situation (e.g., in a library, driving a car, etc.) based on embedded sensors. The sensed context is then used to switch the phone into a suitable

alert mode accordingly (e.g., vibrate mode while in a library, handsfree mode while driving, etc.). Inferring (social and internal) context introduces a challenging research problem in human behavioral data mining. Context is determined by the current state of mind (internal), relationship (social), and surroundings (physical). Thus, the current state of context is important and can be derived from the recent behavior and pattern. In data mining research area, therefore, two frameworks are developed for detecting recent patterns, where one is a model-driven approach and the other is a data-driven approach.

ACKNOWLEDGMENTS

I wish to thank various individuals who have not just helped inspired this dissertation but have also assisted its direction and supported me in many ways. First and foremost, I would like to sincerely acknowledge my advisor Dr. Ram Dantu, who has given me motivation, direction, and invaluable suggestions for this dissertation. Dr. Dantu has given me the opportunities to explore my research interests and helped develop my potential to become a critical thinker as well as an independent researcher. I would also like to thank Dr. João Cangussu, Dr. Philip H. Sweany, and Dr. Parthasarathy Guturu for serving on my dissertation committee. I am also indebted to all of my teachers in Thailand and professors at Southern Methodist University (SMU), in particular, Ajarn Kasimada (my Thai high school advisor) for her teaching and advice, Dr. Tom Chen (SMU) for his supports and allowing me to work on the Phisherman project, Dr. Dinesh Rajan (SMU) for answering numerous questions regarding information theory, Dr. James G. Dunham (SMU) for walking me through my first research paper writing, and Dr. Mandyam D. Srinath (SMU) for giving me the opportunities to experience teaching at SMU as a TA. I would also like to thank my students in EE3350 for voting for me to receive the Outstanding TA Award in 2004. This award means a lot to me and it gives me a great motivation to pursue my Ph.D. as I have found that what I would love to do for living is teaching in college.

This dissertation would not have been possible without unconditional love and support from my mother, Mea Jan, and godmother, Mea Nood, whom I love and respect so much. My mother teaches me that "education is a treasure." My godmother teaches me to be a good person. I am so thankful for their love and teaching. I wish my father was here to see my accomplishment. I know that he has been around and looking out for me. I love you Pa! Finally, I would like to express my deepest appreciation to my wife, Raktida, who has stood by me and supported me through this long process. She is a constant source of joy and inspiration. Her father, Po Boon, mother, Mea Luean, and brother, Nong Aut have been my greatest supporters. They are always in my heart.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Context-aware computing, a relatively new research area in computer science that refers to a computing paradigm in which a single or multiple computing devices (e.g., computer, sensor, handheld device) senses the user's context and responses to that context to support the user in carrying out everyday life activities. The idea of context-aware computing is originated from ubiquitous computing, which was first introduced by Mark Weiser (1) as a computing paradigm that made multiple computing devices available throughout the physical environment and effectively invisible to the user. Recently, it is also referred to as pervasive computing and ambient intelligence. The current research areas include software design (e.g., (2; 3; 4; 5; 6)), privacy and security (e.g., (7; 8; 9; 10; 11)), ubiquitous data access (e.g., (12; 13; 14; 15; 16)), sensing (e.g., (17; 18; 19; 20; 21)), resource scarcity (e.g., (22; 23; 24; 25; 26)), wearable computing (e.g., (27; 28; 29; 30; 31)), user interfaces (e.g., (32; 33; 34; 35; 36)), mobile social software (e.g., (37; 38; 39; 40; 41)), and context-aware computing (e.g., (42; 43; 44; 45; 46)).

The term context-aware computing was introduced in 1994 by Schilit et al. (47) who defined "context" as where you are, who you are with, and what resources are nearby. Schilit et al. divided context into three categories: computing context (network connectivity, communication costs, communication bandwidth, nearby resources e.g., printers, displays, workstations), user context (user's profile, location, people nearby, current social situation), and physical context (lighting, noise levels, traffic conditions, temperature). This groundbreaking research leads to several attempts to formally define context as follows:

- In 1999, Schmidt et al. (48) defined context as "knowledge about the user's IT device's state including surroundings, situation, and location."

- In 2000, Chen and Kotz (49) defined context as "the set of environmental states and settings that either determines an applications behavior or in which an application event occurs and is interesting to the user." They divided context into four categories: computing context, user context, physical context, and time context (time of a day, week, month, and season of the year) by adding the "time context" to the original definition proposed by Schilit et al..
- In 2001, Dey (50) defined context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."
- In early 2003, Hofer et al. (51) divided context into physical and logical context, where the physical context was defined as a representation of environment sensors, while the logical context was defined as a representation of more abstract information about the environment.
- Later in 2003, Prekop and Burnett (52) divided context into external and internal context, where the external context referred to the context that can be measured by hardware sensors (e.g., location, light, air pressure), whereas the internal context was defined as the context that is specified by the user or identified by by monitoring user interations (e.g., the user's goals, tasks, business processed, the user's state emotional state).
- In 2005, Bradley and Dunlop (53) divided context into several categories: task context (the functional relationship of the user with other people and objects and the benefits or constraints this relationship places on the user achievement of his or her goal), physical context (the environmental location, including the orientation, position, state, and purpose of those objects and the types of information they transmit through audio and visual means, odor, texture, temperature, and movement.), social context (the relationship with; dialogue from; and density, flow, noise, and behavior of surrounding people), temporal context (what gives a current situation meaning,

based on past situations/occurrences; expected future events; and the higher level temporal context relating to the time of day, week, month, or season), cognitive context (the users cognitive processing abilities), and application context (the capabilities and limitations of both the application and the sources from which data are derived).

- In 2007, O'Connor et al. (54) defined context as "an abstraction of environmental situations (recognizable by sensors) that have similar meaning." The environmental situations refers to distinct states in which the environment can be.
- In 2008, Han et al. (55) divided context into physical, internal, and social context. Physical context referred to real world nearby user, making up of physical things, such as computer, print, fax, building and so on. Internal context was composed by abstract things inside people, such as feeling, thought, task, action, interest and so on, which was very related to people. Social context meant user's social surrounding, that was to say, social relationship of user. It consisted of persons related to user.

As the context-aware computing research continues to evolve, I believe that the definition of the "context" will continue to move towards the inferential center.

1.1. Context-aware Mobile Computing

In mobile computing area, the context is sensed and used to enable mobile device to better serve for the user. Mobile device observes, records, and analyzes its user's behavior, and responds to the context of the user with minimum user interaction to assist the user by providing information, helping the user making decision, reminding the user of special events, and so on.

1.2. Motivation

People increasingly engage in and rely on mobile phone communications for both personal and business purposes. Hence mobile phones become an indispensable part of life for many people. As mobile networks are expanding rapidly to facilitate the rising number of mobile phone population, more services are expected to be offered (56; 57). To meet this need,

mobile computing research has been focused on developing technologies for handheld devices such as mobile phones, notebook computers, and mobile IP (*e.g.*, (58; 59; 60; 61)). Today, emphasis is increasing on context-aware computing, which is aimed to build the intelligence into mobile devices to sense and respond to the user's context (*e.g.*, (62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79)).

Inferring the context is a challenging task. When humans talk with humans, they are able to user implicit situational information, or *context*, to increase the conversational bandwidth. Unfortunately, this ability to convey ideas does not transfer well to humans interacting with computers (50). It becomes even more difficult especially when a computer tries to determine the context of a human who does not necessarily try to interact with it. Context-aware mobile computing thus requires a blend of the interdisciplinary computing knowledge and domain areas.

Today's mobile technologies suggest that the future of the context-aware mobile computing applications is leaning towards real-time social and internal context with minimum user interaction. To move forward in this direction, this dissertation is centered around designing and building the models for inferring social and internal context of the mobile phone users. The challenges in the study include data collection, verification, and validation methodology. In addition, due to the human-centric nature of the context-aware computing paradigm, it faces a big challenge in dealing with the complexity and dynamics of human behavior. To extract what in a human mind, one must incorporate a mechanism that characterizes human dynamics (*i.e.,* changes and trends in behavior), which appear to be one of the top challenges of human behavioral data mining.

This dissertation is aimed to provide methods and algorithms for inferring and analyzing social context, frameworks for internal context-aware mobile applications, and data mining techniques for human behavioral data.

Context-aware computing services can offer contextually relevant information to the users to facilitate their daily life activities, but at the same time, the risks of data misuse threaten the information privacy of individual users as well as the service providers' business model.

Without a secure infrastructure and privacy-preserving contextual information sharing systems, there is potential of misuse of data and technology. I are aware of the issue, and at the same time, not wanting to let my design to be limited by these concerns. I believe that the use or misuse of technology will continue to be a human issue that is not easily discerned. There are always a positive side as well as a negative side of each technology. Therefore, in dissertation, the frameworks that I are developing is done hoping that the users will benefit most from them.

1.3. Contributions

This dissertation makes contributions to three major areas:

(1) Social computing: The contributions in social computing area are centered around mining mobile phone call logs to extract and analyze hidden patterns for better understanding of mobile phone user behavior as well as face-to-face/mobile social context. As the results, I have carried out the following studies:

   (a) *Quantifying randomness in calling behavior of a mobile phone user*: I analyze the behavior of mobile phone users and identify behavior signatures based on their calling patterns. I quantify and infer the relationship of the user's randomness level using information entropy based on the user's location, time of the calls, inter-connected time, and duration of the calls.

   (b) *Inferring organizational groups using mobile phone call logs*: I develop a model for inferring face-to-face organizational groups based solely on mobile phone call logs using kernel-based naïve Bayesian learning. I also introduce normalized mutual information for feature selection process.

   (c) *Identifying mobile social group sizes and their successive ratio*: I develop a mobile social grouping scheme, which allows us to derive the social group sizes and scaling ratio. The results are compared with the face-to-face's. I conclude that social mobile network is a subset of the face-to-face social network, and both groupings are not necessary the same, hence the scaling ratios are distinct.

(d) *Revealing the impact of social context on calling patterns*: I reveal the significant role of social context on similarity in calling patterns and inter-connect time. I also show that social tie and mobile life pattern can be predicted accurately with a discrete Markov model and a Moving Average model, respectively.

(2) Context-aware mobile computing: The contributions in context-aware computing area are focused on determining the internal context of the mobile phone user. my notion of context is defined as (i) intention of the user/callers *i.e.,* what number to be dialed/received, and (ii) state of mind of the user regarding the alert mode *i.e.,* what alert mode (vibrate, handsfree, ringer) to be set. Within this scope, the following models have been developed to sense and respond to the user context:

(a) *A model for predicting incoming calls for the next 24 hours*: I develop a model for sensing caller intention of making phone calls to the user by predicting incoming calls for the next 24 hours based on call history. The predictor can be used to assist daily scheduling.

(b) *A model for listing potential callers and callees*: I design a call predictor based on naïve Bayesian classifier that makes use of the call logs to predict incoming as well as outgoing phone calls. As an incoming call predictor, my model makes use of the user's call history to generate a list of numbers/contacts that are the most likely to be the callers within the next hour. On the other hand, when the user wants to make an outgoing call (e.g., the user flips open the phone or unlocks the phone, etc.), the outgoing call predictor generates a list of most likely number/contacts to be dialed.

(c) *A model for context-aware alert mode control*: I develop a context-aware mobile computing model that intelligently configures the mobile phone alert mode according to the user's context (e.g., in a meeting, in a movie theater, driving, etc.). I propose a three-step approach in designing the model based on the embedded sensor data from accelerometer, GPS antenna, and microphone of a G1 phone.

(3) Human behavioral data mining: Dealing with human behavioral data is inevitable in designing a context-awareness system. Human behavioral data is a stream data at a variety rate. This rate of data generation depends on the type of behaviors that are being monitored. The changes in the recent data are normally more significant than the old data. Especially, human behavior tends to repeat periodically, which creates a pattern that alters over many periods due to countless factors. Generally, future behavior is more relevant to the recent behavior than the older ones. Thus, my contributions in this area are concentrated in designing frameworks for detecting adequacy and recent pattern of human behavioral data. The frameworks are useful for predictive modeling and dimension reduction for human behavioral data in context-aware system.

   (a) *A framework for detecting adequacy of temporal data*: I develop a framework for detecting the adequacy of historical data (how much of the historical data is enough) to capture a caller's calling behavior. This adequate amount of historical data is empirical proven to be more relevant to the future caller behavior than considering the entire historical data and hence useful for constructing a predictive model for caller behavior. This framework can also be used for any type of temporal data.

   (b) *A framework for detecting recent pattern of temporal data*: I design a framework for detecting recent pattern of human behavioral time series data. Human behavioral data are usually high-dimensional time series. Thus, the detected recent pattern is used for dimension reduction to improve the efficiency of computation and indexing. This framework can also be used for any type of time series data.

1.4. Dissertation Road Map

This dissertation discusses methods for sensing and mining social and internal context of a mobile phone user. The content in this dissertation is organized into 11 chapters.

- Chapter 1 (Introduction): In this chapter, I give an overview of context-aware mobile computing. In this overview, I review previous attempts to define and provide a characterization of "context" and context-aware computing. Furthermore, I describe the motivation of my studies and outline the contributions of this dissertation.
- Chapter 2 (Behavioral Entropy of a Mobile Phone User): In this chapter, I carry out a behavior analysis of mobile phone users and identify behavior signatures based on their calling patterns. I quantify and infer the relationship of a person's randomness levels using information entropy based on the location of the user, time of the call, inter-connected time, and duration of the call. I use real-life call logs of 94 mobile phone users collected at MIT by the Reality Mining Project group for a period of nine months. I am able to capture the user's calling behavior on various parameters and interesting relationship between randomness levels in individual's life and calling pattern using correlation coefficients and factor analysis. This study extends our understanding of cellular phone user behavior and characterizes mobile phone users in form of randomness level.
- Chapter 3 (Inferring Social Group using Call Logs): For a given call log, how much can we say about the person's social group? Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which I believe that it can be extracted to infer its user's social group information. In this chapter, I present an end-to-end system for inferring social networks based on "only" call logs using kernel-based naïve Bayesian learning. In addition, I introduce "normalized mutual information" for feature selection process. my model is evaluated with actual call logs and it yields promising results.
- Chapter 4 (Mobile Social Group Sizes and Scaling Ratio): In this chapter, I present a method to quantify mobile social closeness and describe a social grouping scheme, which is then used to identify social sizes and scaling ratio. my social grouping approach has been validated with the real-life datasets with high accuracy. With my mobile social grouping results, I identify a group sizes' scaling ratio of close to

"8" based on two different analyses where one is based on mean group sizes and the other is based on all raw group clusters. my results are compared with the findings from previous studies of face-to-face social groups. I draw a conclusion that the mobile social network is a subset of the face-to-face social network, where both have distinct groupings and constant group sizes' scaling ratios.

- Chapter 5 (Mobile Social Context and Communication Patterns): In this chapter, I reveal the significant role of social context (described in Chapter 4) on similarity in calling patterns and inter-connect time. my results show that (i) the closer the social tie, the higher the similarity, (ii) a closer tie implies higher reciprocity, and (iii) the inter-contact time increases as social closeness becomes distant.
- Chapter 6 (Call Predictor: Phone Call-based Daily Planner): In this chapter, I present a model for call predictor that computes the probability of receiving calls and makes prediction of incoming calls for the next 24 hours, based on caller behavior and reciprocity.
- Chapter 7 (Call Predicted List: List of Potential Callers and Callees): In this chapter, I describe a model for a incoming/outgoing call predictor based on the naïve Bayesian classifier. As an incoming call predictor, my model makes use of the user's call history to generate a list of numbers/contacts that are the most likely to be the callers within the next hour. On the other hand, when the user wants to make an outgoing call (e.g., the user flips open the phone or unlocks the phone, etc.), the outgoing call predictor generates a list of number/contacts to be called.
- Chapter 8 (Context-aware Alert Mode for a Mobile Phone): Forgetting to switch to vibrate mode while in a movie theater or a meeting, and taking the risk of picking up a phone call while driving can be avoided if the phone is smart enough to recognize its user's situational context. As the first step towards that direction, in this chapter, I present a design of a model that can intelligently switches the alert mode according to the user's context. The alert mode is to be set according to the

recognized context state as vibrate, handsfree, and ringer mode. I describe a three-step approach in design based on the embedded sensor data from accelerometer, GPS antenna, and microphone of a G1 phone. I have evaluated my model in several aspects using training and testing data collected from participating subjects. Based on the experiments, my model has shown promising results.

- Chapter 9 (Adequacy of Data for Characterizing Caller Behavior): In this chapter, I describe a method for characterizing caller behavior based on caller's call arrival, inter-arrival, and talk time using probabilistic approach. A probabilistic model is generally used to predict or estimate the future observation, which is conditioned by a knowledge of the historical data. The question is how much historical data is adequate? I answer this question by presenting a technique to detect and compute the adequate amount of historical data to capture the caller behavior. In fact, this adequate amount of historical data has been proven empirically to be more relevant to the future caller behavior than considering the entire historical data. Hence it is useful for constructing a predictive model for caller behavior. In addition, I show the improvement in the performance of a Call Predictor (described in Chapter 6) when applying adequacy of data.
- Chapter 10 (A Recent-pattern biased Dimension-reduction Framework for Time Series Data): High-dimensional time series data need dimension-reduction strategies to improve the efficiency of computation and indexing. In this chapter, I present a dimension-reduction framework for time series data. Generally, recent data are much more interesting and significant for predicting future data than old ones. The basic idea is to reduce to data dimensionality by keeping more detail on recent-pattern data and less detail on older data. I distinguish my work from other recent-biased dimension-reduction techniques by emphasizing on recent-pattern data and not just recent data. I experimentally evaluate my approach with synthetic data as well as real data. Experimental results show that my approach is accurate and effective as it outperforms other well-known techniques.

- Chapter 11 (Conclusion): In this chapter, I conclude this dissertation with a summary of the contributions and a vision of the future studies.

CHAPTER 2

BEHAVIORAL ENTROPY OF A MOBILE PHONE USER

2.1. Introduction

Mobile phone has moved beyond being a mere technological object and has become an integral part of many people's social lives. This has had profound implications on both how people as individuals perceive communication as well as in the patterns of communication of humans as a society. In this chapter, I try to capture the behavior of phone users based on their calling patterns and infer trend of behavior dependencies using techniques such as Entropy, principal factor analysis, and correlation function. I present a new method for precise measurement of randomness of phone user based on their calling patterns such as location of the call, talk time, calling time, and interconnected time; and infer relationship among them.

Recently there has been increasingly growing interests in the field of mobile social network analysis, but due to the unavailability of data, there have been far fewer studies. The Reality Mining Project at Massachusetts Institute of Technology (MIT) (80) has made publicly available large datasets from their projects. I implement my techniques on the Reality Mining dataset which was collected over nine months by monitoring the cell phone usage of 84 participants. The information collected in the call logs includes user IDs (unique number representing a mobile phone user), time of call, call direction (incoming and outgoing), incoming call description (missed, accepted), talk time, and tower IDs (location of phone users). These 84 phone users are students, professors, and staffs.

Using purely objective data first time the researchers can get an accurate glimpse into human behaviors. My interest in this data set is to study the behavior of the phone user using information theory, data mining, and data reduction techniques.

In (81), the authors attempted to quantify the amount of predictable structure in an individual's life using entropic metric and discovered that people who live high-entropy lives tend to be more random or less predictable than people who live low-entropy lives. This raises the question about how this entropy-based randomness level is related to the randomness level in calling behavior. Does it mean that people who have high-entropy lives also have high-entropy calling patterns? To answer this question, I find it interesting to study the relationship between the randomness level in individuals life and calling pattern.

2.2. Randomness Level Computation

While individual phone user's calling behavior is random, some users might be more predictable than others. Being more predictable can also mean being less random. To quantify the randomness or amount of predictable structure in an individual calling pattern, the information entropy can be used.

The information entropy or Shannon's entropy is a measure of uncertainty of a random variable. The information entropy as given in Equation (1) was introduced by Shannon (82), where X is a discrete random variable, $x \in X$, and the probability mass function $p(x) = Pr\{X = x\}$.

$$H(X) = -\sum_{x} p(x) \log_2 p(x). \tag{1}$$

The calling pattern can be observed from the calling time, inter-connected time (elapsed time between two adjacent call activities), and talk time (duration of call). Let C, I, and T be random variables representing calling time, inter-connected time, and talk time respectively. The entropy of calling time can be calculated by Equation (2).

$$H(C) = -\sum_{c=1}^{24} p(c) \log_2 p(c), \tag{2}$$

where the probability $p(c)$ is a ratio of the number of calls during c^{th} hour slot to the total number of calls of all time slots (N).

Similarly, the entropy of inter-connected time can be calculated by Equation (3) where $p(i)$ is a ratio of the number of inter-connected time whose value is in the interval $[i-1, i)$

to $N-1$.

$$H(I) = -\sum_{i} p(i) \log_2 p(i). \tag{3}$$

Likewise, the entropy of the talk time is given by Equation (4) where $p(t)$ is a ratio of the talk time whose value is in the interval $[t-1, t)$ to N.

$$H(T) = -\sum_{t} p(t) \log_2 p(t). \tag{4}$$

By the same token, the randomness in the individual life's schedule (location), $H(L)$ can also be quantified using information entropy which is defined in Equation (1).

2.3. Result and Analysis

Based on my real-life call logs of 84 users, I infer the relationship between the randomness based on the underlying parameters by computing the correlation coefficient (83). Correlation coefficient is a number between -1 and 1 which measures the degree to which two random variables are linearly related. A correlation coefficient of 1 implies that there is perfect linear relationship between the two random variables. A correlation coefficient of -1 implies that there is inversely proportional relationship between the two random variables. A correlation coefficient of zero implies that there is no linear relationship between the variables. As a preliminary result shown in Table 2.1, it can be observed that the randomness based on location($H(L)$) and calling time($H(C)$) show high correlation as well as the $H(I)$ and $H(T)$ pair.

TABLE 2.1. Result of correlation coefficient

	$H(L)$	$H(C)$	$H(I)$	$H(T)$
$H(L)$	1.0000	0.4651	-0.4695	-0.4642
$H(C)$	0.4651	1.0000	-0.2218	-0.3502
$H(I)$	-0.4695	-0.2218	1.0000	0.2197
$H(T)$	-0.4642	-0.3502	0.2197	1.0000

Next, I perform factor analysis in order to further study the relationship of the randomness levels (entropy) based on the underlying parameters. The main application of factor analysis is: (i) to reduce the number of variables and (ii) to detect structure in the relationship between variables, that is to classify variables (84). In my analysis I use it for both the purposes. The flow diagram of the principal factor analysis is shown in Fig. 2.1.

Two principal factors are selected based on the Scree plot (85). The principal factor plot of the entropy based on four parameters lying on the first and second factor is shown in Fig. 2.1. It can be observed that the $H(L)$ and $H(C)$ are positively lying on the first factor whereas the $H(I)$ and $H(T)$ are positively lying on the second factor. Since the first and second factor are orthogonal *i.e.*, uncorrelated, one can notice two established relations; one is between $H(L)$ and $H(C)$, and the other one is between $H(I)$ and $H(T)$.

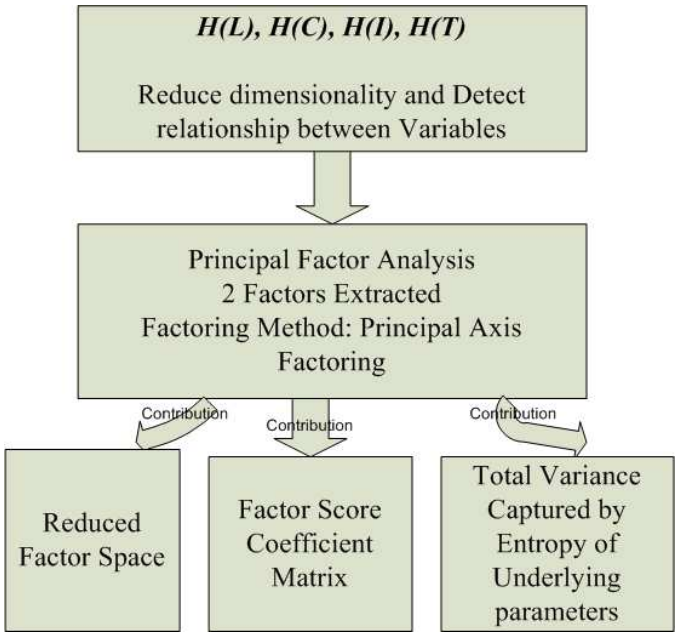


FIGURE 2.1. Flow diagram for principal factor analysis on calculated entropy.

Factor analysis generally is used to encompass both principal components and principal factor analysis. The Eigen values for a given factor measures the variance in all the variables which is accounted for by that factor as stated in Table 2.2. If a factor has a low eigen

value, then it is contributing little to the explanation of variances in the variables and may be ignored as redundant with more important factors.

TABLE 2.2. Total variance explained

Factor	Initial Eigen Values			Extraction		
	Total	Variance(%)	Cumulative(%)	Total	Variance(%)	Cumulative(%)
1	1.59	39.95	39.95	0.92	23.20	23.20
2	1.02	25.61	65.57	0.29	7.26	30.46
3	0.73	18.24	83.81	-	-	-
4	0.64	16.18	100.00	-	-	-

Eigen value is not the percent of variance explained but rather a measure of amount of variance in relation to total variance (since variables are standardized to have means of 0 and 1, total variance is equal to the number of variables).

Initial eigen values and eigen values after extraction (extracted sums of squared loadings) are same for Principal Component Analysis (PCA) extraction (86), but for factor analysis eigen values after extraction will be lower than their initial counterparts.

Scree plot was developed by Cattell (85) for selecting the number of factors to be retained in order to account for most of the variation. In my analysis, based on Kaiser's criterion (87) the first two factors whose eigen values are greater than 1 (as listed in Table 2.2) are selected based on the scree plot shown in Fig. 2.2.

The plot of the entropy based on four parameters lying on the first and second factor is shown in Fig. 2.3. It can be observed that the entropy based on location and calling time are positively lying on the first factor whereas the entropy based on inter-connected time and talk time are positively lying on the second factor. Since the first and second factor are orthogonal *i.e.*, uncorrelated, one can notice two established relations; (i) between entropy based on location and calling time and (ii) entropy based on inter-connected time and talk time.

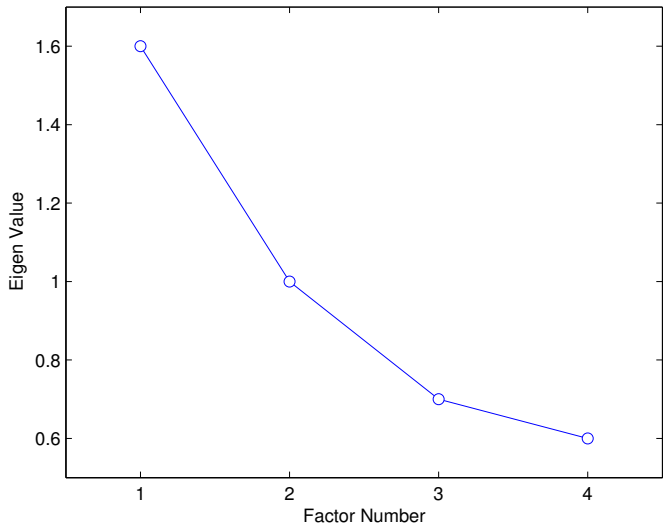


FIGURE 2.2. Scree plot.

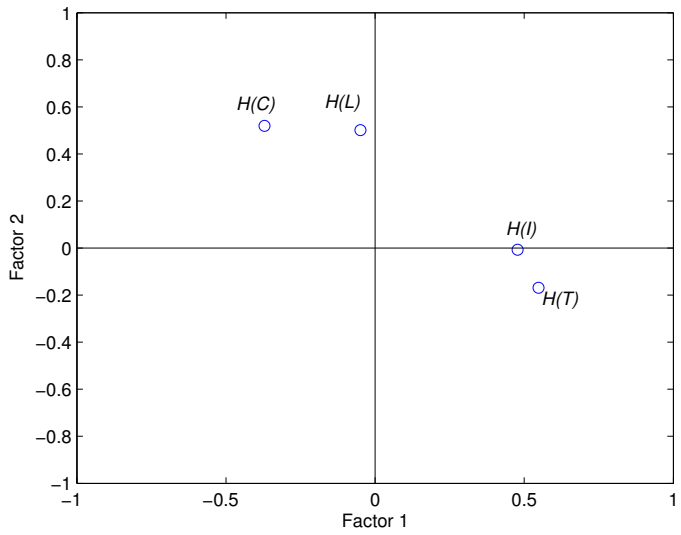


FIGURE 2.3. Principal factor plot.

The scatter plots in Fig. 2.4 also confirm my findings by showing the proportional relationships between pairs $(H(L), H(C))$ and $(H(I), H(T))$, and inversely proportional relationships among other pairs. The trend (linear-fitting) line is shown in red to emphasize the direction of the relationship, directly proportional (increasing) or inversely proportional (decreasing). Note that the linear fitting is obtained by the least square fitting method (88).

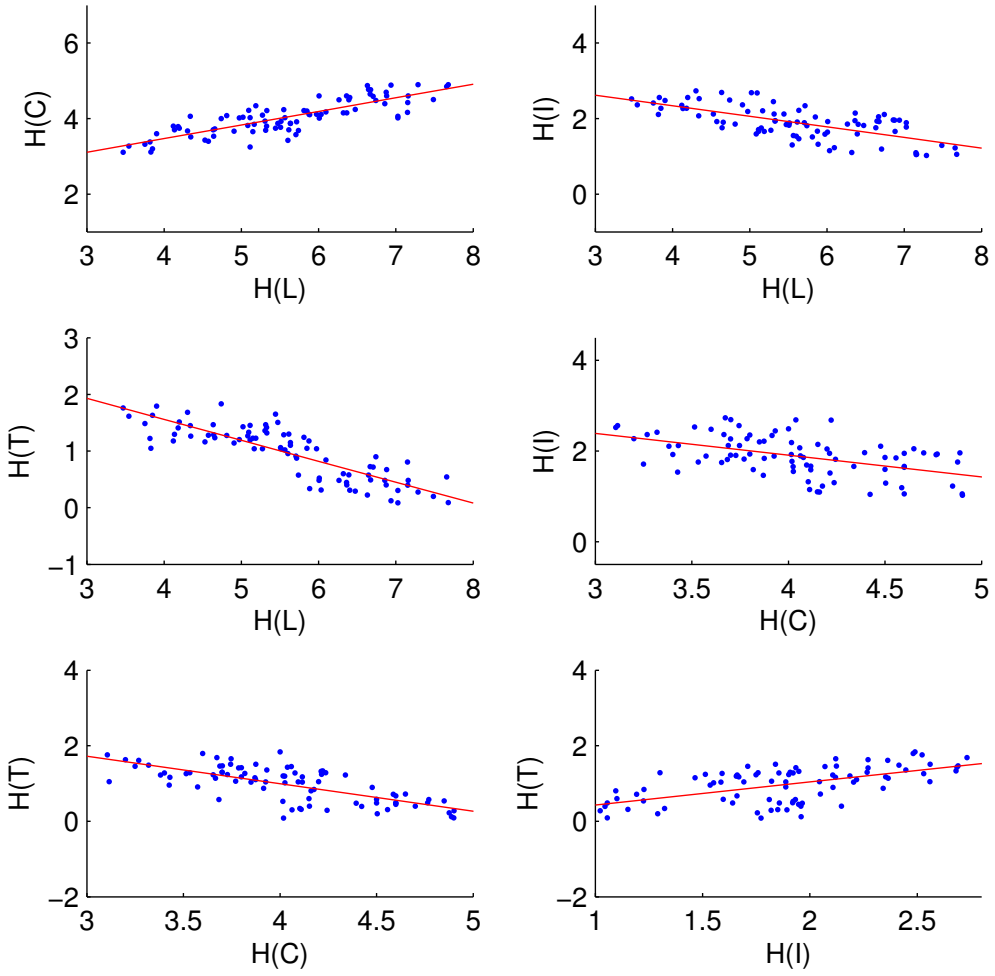


FIGURE 2.4. Scatter plots showing relationships among $H(L)$, $H(C)$, $H(I)$, and $H(T)$ with the linear trend lines.

The results based on the correlation coefficients, factor analysis, and scatter plots tell us that there is a high correlation in the randomness in phone user's location and calling time, as well as high correlation in the randomness in phone user's inter-connected time and talk time. This draws the conclusion of my study that phone users who have higher randomness in mobility tend to be more variable in time of making calls but less variable in time spent talking on the phone and the time between connection (idle time). By the same token, the phone users who spend higher random amount of time talking on the phone (connected

time) tend to also be more variable in idle time but not less random in mobility and time of initiating the calls.

I believe that this finding can also be useful for the phone service providers in offering right plans for the right customers based on customer's calling behavior, *e.g.*, suppose that a customer has increasingly high randomness in mobility, service provider might offer this customer a whenever-minute plan which would fit his calling pattern (high $H(L)$ implies high $H(C)$).

2.4. Conclusion

In this chapter, I have presented and analyzed cellular phone user behavior in forms of randomness level using information entropy based on user's location, time of call, inter-connected time, and duration of call. I am able to capture the relationship of the user's randomness level based on the underlying parameters by utilizing the correlation coefficient and factor analysis.

Based on my study, the user's randomness level based on location has high correlation to the randomness level in time of making phone calls and vice versa. My study also shows that the randomness level based on user's inter-connected time has a high correlation to the randomness level in time spent talking on each phone call.

A knowledge of the randomness levels of a phone user behavior and their relationships extends our understanding in the pattern of user behavior. I believe that this work can also be extended to predict what services that are suitable for the user. This study will also be useful for the future research in this area.

CHAPTER 3

INFERRING SOCIAL GROUPS USING CALL LOGS

3.1. Introduction

Social network describes a social structure mode of social entities and the pattern of inter-relationships among them. A social network can be either face-to-face or virtual network in which people primarily interact via communication media such as letters, telephone, email, or Usenet. Knowledge of social networks can be useful in many applications. In commerce, viral marketing can exploit the relationship between existing and potential customers to increase sales of products and services. In law enforcement, criminal investigation concerning organized crimes such drugs and money laundering or terrorism can use the knowledge of how the perpetrators are connected to one another to assist the effort in disrupting a criminal act or identifying additional suspects.

Social computing has emerged recently as an exciting research area which aims to develop better social software to facilitate interaction and communication among groups of people, to computerize aspects of human society, and to forecast the effects of changing technologies and policies on social and cultural behavior. One of the major challenges in social computing is obtaining real-world data. Quite often, analysis is based on simulations.

With rapidly increasing number of mobile phone users, mobile social networks have gained interests from several research communities. I also find it interesting to study the relationships between mobile phone users' calling behaviors and their social networks. With availability of real-life data of mobile phone users' call logs collected by Reality Mining project group (81), it allows us to carry out my analysis and experimental results in this paper in which I propose an end-to-end system for inferring social networks based solely on call logs. I believe that phone user's calling personality and habit has been unnoticeably concealed in the call logs from which it can be extracted to infer its user's social networks information.

To the best of my knowledge, no scientific research has been reported in classifying social networks/groups based solely on call logs.

3.2. System Overview

The system described here is intended to perform social networks/groups classification based on personal phone records. The input is phone records or call logs showing pertinent information (number dialed, duration, time of communications, etc.). The call logs is then transformed into knowledge useful for the classifier by extracting calling patterns and selecting useful features. The kernel-based naïve Bayesian classifier is used to perform supervised classification based on computed probability using kernel density estimator.

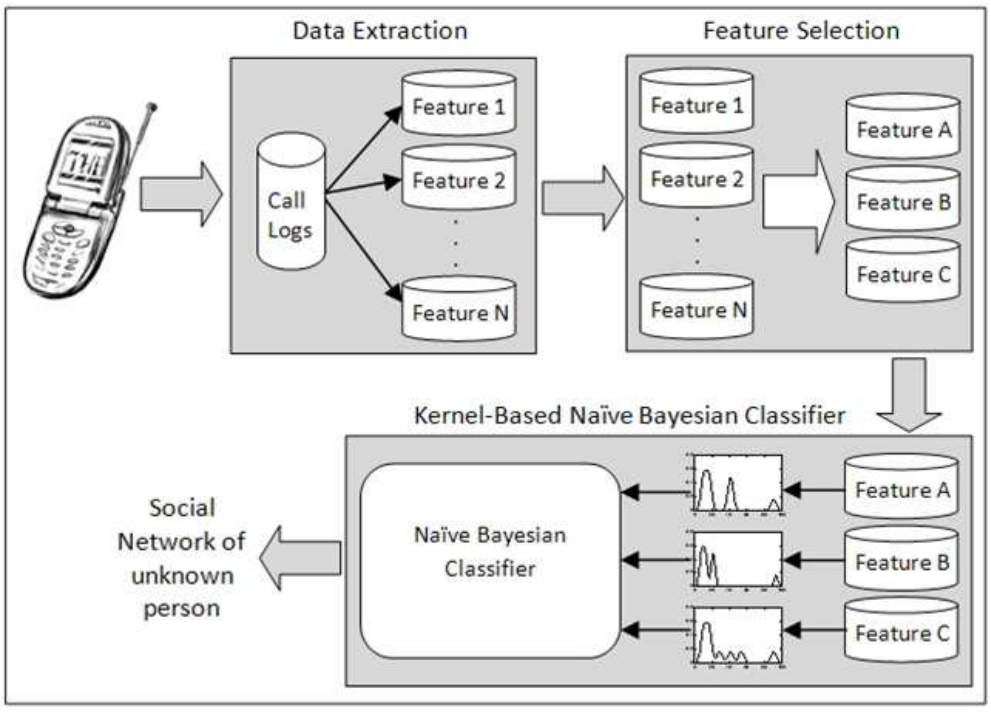


FIGURE 3.1. System overview.

3.3. Dataset

Every day phone calls on the cellular network include calls from/to different sections of our social life. We receive/make calls from/to family members, friends, supervisors, neighbors, and strangers. Every person exhibits a unique traffic pattern. Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which we believe that it can be extracted to infer its user's social networks information. To study this, I use the real-life call logs of 94 individual mobile phone users over the course of nine months which were collected at Massachusetts Institute of Technology (MIT) by the Reality Mining project group (81). Call logs were collected using Nokia 6600 smart phones loaded with software written both at MIT and the University of Helsinki to record minutely phone information including call logs, users in proximity, locations, and phone application currently being used. Of 94 phone users, 25 were incoming Sloan business students while the remaining 69 users were associated with the MIT Media Lab. According to MIT (81), this study represents the largest mobile phone study ever in academia and the data collected can be used in a variety of fields ranging from psychology to machine learning. There are some research works conducted by the Reality Mining project group using this dataset involves relationship inference, user behavior prediction, and organizational group dynamics.

As previously mentioned, my interest and the focus of this chapter is to extract the phone user's behavior concealed in the call logs and attempt to accurately classify user into belonging social networks. With MIT dataset, classification can be performed to differentiate phone users from the Media Lab and from Sloan. As described earlier, MIT dataset consists of more than just call logs but location information and others. In order to be more generalized, only call logs are considered for my study as currently call logs are only accessible feature from service providers (*e.g.*, billing, online account).

Due to missing information on the dataset which leaves us 84 users instead of 94 users, I then have 22 Sloan users and 62 Media Lab users. Of 62 Media Lab users, 20 users are clearly marked as students. I believe that even though all 62 users are with Media Lab, sub-social groups can be formed such as students, faculty, and staff, which exhibit slightly different

calling behavior. Therefore I choose to perform classification between clearly marked Media Lab students and Sloan students.

3.4. Feature Extraction

The main goal is to find some features from the call logs (raw data) that can solidly differentiate Media Lab students and Sloan students. For the data extraction process, I try to retrieve as much as possible useful features from the call logs. There might not be one dominate feature that captures entire calling behavior but combination of those characterize the core behavior structure. There are 11 features extracted and listed in Table 3.1 along with some statistical analysis (*i.e.*, averages (*Avg.*) and standard deviations (*Std.*)) where feature descriptions are listed in Table 3.2.

From the first glance of these features and their statistics, it is clear that there are differences between two social networks however the differences are not adequately large enough to differentiate them based on each individual feature.

TABLE 3.1. Extracted features

Features	Media Lab		Sloan	
	Avg.	Std.	Avg.	Std.
All_calls	9.670	6.902	14.168	7.264
Inc_calls	2.920	2.542	3.756	2.197
Out_calls	6.750	4.501	10.413	5.549
Missed_calls	8.708	6.167	12.810	6.718
All_talk	246.716	304.899	196.966	260.884
Inc_talk	140.906	109.479	172.518	111.427
Out_talk	272.599	367.231	207.328	320.731
All_call_time	12.934	1.671	14.571	2.120
Inc_call_time	13.164	1.775	14.591	2.226
Out_call_time	12.881	1.752	14.583	2.121
Ent_call_time	6.137	0.683	4.059	0.553

TABLE 3.2. Extracted feature descriptions

Features	Feature description
All_calls	The total number of all calls per day including incoming, outgoing, and missed calls.
Inc_calls	The number of incoming calls per day.
Out_calls	The number of outgoing calls per day.
Missed_calls	The number of missed calls per day.
Inc_talk	The total amount of time spent talking on the phone (call duration) per day (in seconds) including both incoming and outgoing calls.
Out_talk	The amount of time spent talking (in seconds) per day on the incoming calls.
All_call_time	The amount of time spent talking (in seconds) per day on the outgoing calls.
Inc_call_time	The time that calls either received or made, ranging between 0 and 24 (0AM – 12PM).
Out_call_time	The arrival time of incoming calls, 0-24 (0AM – 12PM).

The last feature in Table 3.1 and 3.2 is information entropy (82) which is a measure of the uncertainty of a random variable. In my case, this random variable is calling time. The entropy of a variable X is defined by Equation (5) where $x_i \in X$ and $P(x_i) = Pr(X = x_i)$.

$$H(X) = -\sum_i P(x_i) \log_2(P(x_i)). \tag{5}$$

Assessment based on these extracted features is that Sloan students tend to make more phone calls than Media Lab students, whereas Media Lab students like to talk (or spend time) on the phone longer on outgoing calls but talk less on incoming calls than Sloan students. Sloan students spend time on the phone later in the day (about 2:30PM) than Media Lab students (about 1PM). Lastly, the randomness in calling time of Media Lab students is higher than Sloan students.

3.5. Feature Selection

So far, I have extracted features from raw data (call logs) and I need to select the useful features for classification. This section discusses how to evaluate the usefulness of features for classification. In general, for classification task as I try to assign an unknown sample to different classes which have different characteristics. My goal is to find a character (*e.g.*, a set of features) of the unknown sample that mostly identifies its belonging class among other classes. This set of features need to have high degree of difference (or low degree of similarity) to other classes to be considered as a "good" set of features. If I adopt the correlation between two random variables as a goodness measure, the above definition becomes that a feature is good if it is highly correlated with the belonging class but not highly correlated with other classes.

There are two main approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory. The first approach is the well known *linear correlation coefficient*(r). For any pair of random variables (X, Y), r can be computed by (6).

$$r = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2}\sqrt{\sum_i (y_i - \bar{y}_i)^2}}, \tag{6}$$

where $\bar{x}_i$ is the mean of X, and $\bar{y}_i$ is the mean of Y. The value of r is between -1 and 1. A correlation coefficient of 1, -1, and zero implies perfect linear relationship, inversely proportional relationship, and no linear relationship between the two variables respectively. It is symmetrical measure for two variables. There also exists other measures in this category which are basically variations of r, such as *least square regression error* and *maximal information compression index* (89). There are several benefits of choosing linear correlation coefficient as a goodness measure for feature selection such as it helps remove features with correlation close to one from selection and retain other features with low correlation. However, in the reality it is not safe to always assume "linear" relationship between features. Linear correlation measures may not be able to capture the correlations that are not linear in nature.

Another approach to measure the correlation which is based on information theory can overcome this shortcoming. I adopt the concept of information entropy which is given in Equation (5) which measures the degree of uncertainty between two random variables. Information theory (90) defines conditional entropy of a random variable given another with a joint distribution $P(x_i, y_j)$ as follows.

$$H(X|Y) = -\sum_i \sum_j P(x_i, y_j) \log_2(P(x_i|y_j)). \tag{7}$$

Another important definition is *mutual information* which is a measure of the amount of information that one random variable contains about another random variable which is given by Equation (8).

$$I(X;Y) = -\sum_i \sum_j P(x_i, y_j) \log_2(\frac{P(x_i, y_j)}{P(x_i)P(y_j)}). \tag{8}$$

Given Equation (5) and Equation (7), it is straightforward to derive Equation (9).

$$I(X;Y) = H(X) - H(X|Y). \tag{9}$$

Mutual information is also referred to as *information gain*(91) which can be interpreted as a measure of the amount by which the entropy of X decreases reflects additional information about X provided by Y.

Theorem: The mutual information is symmetrical for two random variables X and Y which can be proved as follows.

Proof: To show that $I(X;Y) = I(X;Y)$.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - (H(X,Y) - H(X)) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \quad \square \end{aligned}$$

For fairness in comparisons, normalization is needed. Therefore the *normalized mutual information* can be derived as

(10) $$I_{Nor}(X;Y) = \frac{H(X) - H(X|Y)}{H(X)},$$

where denominator *H(X)* is a scale factor to normalize it to [0, 1].

3.6. Kernel-Based Naïve Bayesian Classifier

The approach to classification taken here is based on Bayes rule (92) of conditional probability which is given by Equation (11).

(11) $$P(Y|X) = P(Y)\frac{P(X|Y)}{P(X)},$$

where $P(Y|X)$ is the *a posteriori* probability which is the probability of the state of nature being Y given that feature value X has been measured. The *likelihood* of Y with respect to X is $P(X|Y)$ which indicates that other things being equal, the category Y for which $P(Y|X)$ is large is more "likely" to be the true category. $P(Y)$ is called *a priori* probability. The *evidence* factor, $P(X)$, can be viewed as a scale factor to guarantee that the posterior probabilities sum to one.

Suppose now that I have N input features, $X = \{x_1, x_2, .., x_N\}$, which can be considered independent both unconditionally and conditionally given y. This means that the probability of the joint outcome x can be written as a product,

(12) $$P(X) = P(x_1) \cdot P(x_2) \cdots P(x_N)$$

and so can the probability of X within each class y_j,

(13) $$P(X|y_j) = P(x_1|y_j) \cdot P(x_2|y_j) \cdots P(x_N|y_j).$$

With the help of these it is possible to derive the basis for the *naïve Bayesian classifier*(93) as follows,

(14) $$P(y_j|X) = P(y_j)\frac{P(X|y_j)}{P(X)} = P(y_j)\prod_{i=1}^{N}\frac{P(x_i|y_j)}{P(x_i)}.$$

The designation *naïve* is due to simplistic assumption that different input attributes are independent.

From Equation (14), the classification is then based on the likelihood function given by Equation (15).

$$L(y_j|X) = \prod_{i=1}^{N} P(x_i|y_j). \tag{15}$$

Most applications that apply naïve Bayesian classifier derive likelihood function from the actual data or assumed parametric density function (*e.g.*, Gaussian, Poisson). Another approach to derive likelihood function is by using non-parametric density estimation. The most popular method is the kernel estimation which is also known as the Parzen window estimator (94) as follows,

$$f(z) = \frac{1}{Mh} \sum_{k=1}^{M} K\left(\frac{z - z_k}{h}\right), \tag{16}$$

where $K(u)$ is kernel function, M is the number of training points, and h is the bandwidth or smoothing parameter. The most widely used kernel is Gaussian of zero mean and unit variance ($N(0,1)$) which is defined by Equation (17).

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{17}$$

The choice of the bandwidth h is crucial. Several optimal bandwidth selection techniques have been proposed ((95)). In this study, I use AMISE optional bandwidth selection using the *Sheather Jones Solve-the-equation plug-in* method which was proposed in (96).

Kernel density estimator provides smoothness to likelihood function with continuous attributes rather than relying on discrete ones. Now the likelihood function in (15) becomes

$$L(y_j|X) = \frac{1}{Mh} \prod_{i=1}^{N} \left(\sum_{k=1}^{M} K\left(\frac{y_j - z_k^i}{h}\right) \right), \tag{18}$$

where z_k^i is training point k of feature i.

3.7. Implementation and Results

To evaluate my proposed system, I continue my implementation from data extraction process in section 3.4. Recall that I have 11 extracted features from the call logs. Now I need to select useful features based on normalized mutual information as discussed in section

3.5. Based on Equation (10), normalized mutual information is computed for each feature and plotted in Fig. 3.2 for comparison. If normalized mutual information of 0.5 is chosen as a threshold, then I have six featured selected with the highest degree of discriminancy.

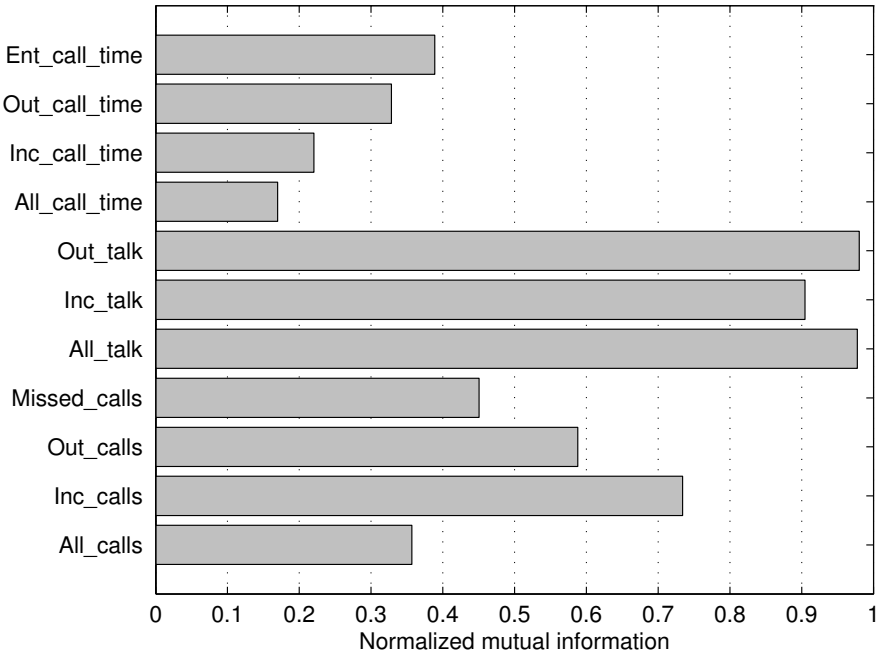


FIGURE 3.2. Result of normalized mutual information.

The six selected features with their corresponding normalized mutual information are listed in ascending order in Table 3.3. Recall that less normalized mutual information implies higher order of discriminancy or most useful feature for classification.

The useful features have been selected, before reaching classifier feature normalization is needed. The reason for normalization is to reduce the noisiness of features since non-normalized features have different ranges and are measured in different units. Thus, selected features are normalized to [0, 1].

Features are now ready to be fed to classifier which operates in two modes; training and testing. I use 50% of my feature set as training data and the other 50% as testing data. I implement my proposed method of using kernel-based naïve Bayesian classifier with selected six features based on normalized mutual information. The performance of my proposed

TABLE 3.3. Selected features based on normalized mutual information

Features	Normalized Mutual Information
All_call_time	0.169
Inc_call_time	0.220
Out_call_time	0.328
All_calls	0.357
Ent_call_time	0.388
Missed_calls	0.450

method is measured by the accuracy rate which is a ratio of correct classified users to the total testing users.

For performance comparison purposes, I also implement naïve Bayesian classifier using all 11 extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features to compare with my method. The result is shown in Table 3.4, among four approaches, my approach has the best performance with accuracy rate of 81.82%. Naïve Bayesian classifier using all 11 extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features perform at accuracy rates of 59.09%, 68.18%, and 77.27% respectively.

TABLE 3.4. Performance comparison

Methods	Accuracy Rate (%)
Naïve Bayes with all features	59.09
Naïve Bayes with six selected features	68.18
Kernel-based naïve Bayes with all features	77.27
Kernel-based naïve Bayes with six selected features	81.82

In addition, to evaluate the effectiveness of the six selected features based on normalized mutual information, I sort all 11 features based on normalized mutual information in

ascending order and monitor the changes in accuracy rate as more ascending sorted features taken into account. I monitor both kernel-based naïve Bayesian and classical naïve Bayesian approach which are shown in Fig. 3.3. Accuracy rate of both methods continue to increase up to when six features are taken into account, then accuracy rate decreases. The accuracy rate continues to decrease after more than six features taken for naïve Bayesian classifier whereas the accuracy decreases from six to seven features and stays constant until all 11 features are taken into account for kernel-based naïve Bayesian classifier.

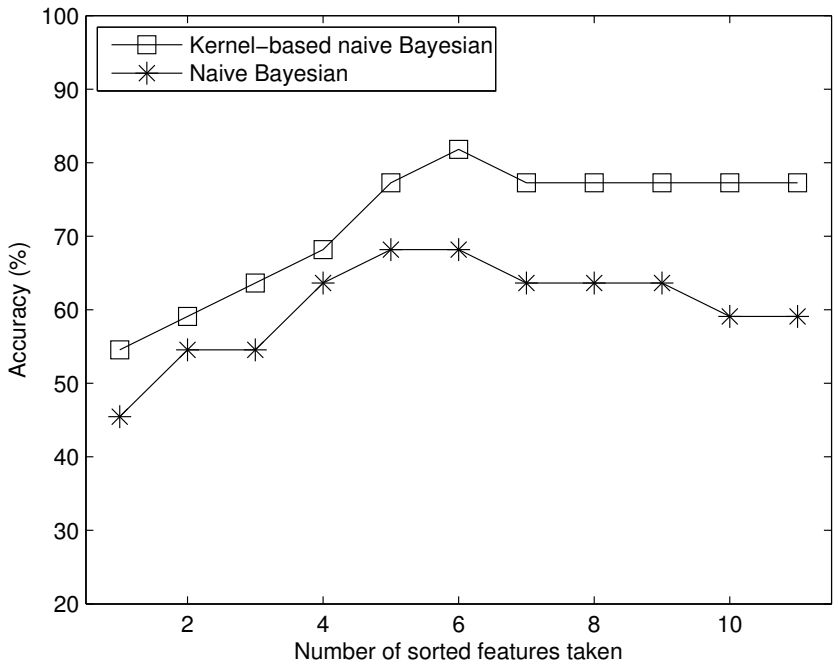


FIGURE 3.3. Change of accuracy according to number of features selected.

Figure 3.3 tells us that the selected six features listed in Table 3.3 are indeed useful features for classification. Including more features for classifier does not mean better performance. In fact, it may degrade the performance of classifier with its noisiness and low degree of discriminancy.

3.8. Conclusion

According to the CTIA (97), there are currently 243 million mobile phone subscribers in the US. With a current population of around 300 million and assuming that the CTIA figure

implies unique subscribers, about two in every three Americans own a mobile phone. With this widespread use of mobile phones, it becomes valuable source of information for social networks analysis. In this section, I analyze social networks based on mobile phone's call logs, and propose a model for inferring groups. I describe data pre-processing process which consists of data extraction and feature selection in which I introduce a technique for selecting features using normalized mutual information that measures degree of disciminancy. With its symmetrical and linearity-invariance property, I show that it makes normalized mutual information suitable for my feature selection process. I adopt the classical naïve Bayesian learning and introduce kernel density estimator to estimate the likelihood function which improves accuracy of the classifier with its smoothness. My model is evaluated with real-life call logs from Reality Mining project group. The performance is measured by the accuracy rate. The results show that my model performs at accuracy rate of 81.82% which is highest among other models (Naïve Bayesian classifier using all extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all extracted features). I believe that my model can also be useful for other pattern recognition and classification tasks.

CHAPTER 4

MOBILE SOCIAL GROUP SIZES AND SCALING RATIO

4.1. Introduction

Humans are evolving as fundamentally social creatures. Our belief and behavior have been shaped by our social context. Understanding social context and its structure can help unfold the concealed patterns that assemble our behavior. As our technology advances, we have created different ways of social networking. Besides the conventional face-to-face social networking, we are now interacting with people on online and mobile networks, which inherit some face-to-face social networking fundamentals and also introduce some new elements and concepts. As mobile networks expand rapidly to facilitate the rising number of mobile phone population, the more mobile social services are being developed and offered. Understanding the mobile social network is the first and an essential step towards creating an intelligent functionality that indeed enhances quality of life with a system that comprehends behavior and context of its user(s).

Human social grouping patterns have been studied extensively in both sociology (98) and social anthropology (99) (100). Dunbar (101) proposed that humans had a cognitive limit of about 150 on the number of individuals with whom coherent personal relationships could be maintained. Later, Zhou et al. (102) identified a social group scaling ratio of "3" as social network members were divided into six groups based on social connectivity with group sizes of about 3-5, 12-20, 30-50, 150, 500, and 1,000-2,000 people.

To the best of our knowledge, no scientific research has been reported in identifying mobile social group sizes and its scaling ratio, thus it is very interesting and important to investigate it for a better understanding of the mobile social network and a useful comparison to the face-to-face social network. The result of the investigation can also be related to behavioral grouping signatures, cognitive process of human brains for social closeness, and mechanisms

governing the human grouping dynamics. In this section, we mine mobile social network data by presenting a simple but efficient method to quantitatively define social closeness and social grouping, which are then used to identify social sizes and scaling ratio.

4.2. Mobile Social Closeness and Grouping

Literatures in social science ((103),(104)) discussed the social closeness of people based on amount of time and intensity of communication. Granovetter et al. (103) found that the time spent in a relationship and the intensity along with the intimacy and reciprocal services formed a set of indicators for social tie. The paper predicted that the strength of an interpersonal tie was a linear combination of amount of time, the emotional intensity, the intimacy (mutual confiding) and the reciprocal services in a relationship. Marsden et al. (104) evaluated the indicators and predicators of strength (tie) described by Granovetter et al. (103), and concluded that "social closeness" or "intensity" provided the best indicator of strength or tie. Marsden *etl al.*'s conclusion was derived based on the survey study of 2,329 human subjects who were drawn from three cross-sectional surveys conducted at two American cities (Detroit, Aurora) and a small city in the Federal Republic of Germany. Responders were asked to identify their three closest friends and report characteristics of these persons such as age, occupation, religion, and so on.

In mobile social network, amount of time and intensity of communication can be measured by call duration (talk time) and call frequency (number of phone calls). In our daily life, we communicate with people in the mobile network at different instances. These people constitute our mobile social network. Based on amount of time and intensity of communication with these people, our mobile social network can be divided into three broad groups:

Group 1: Socially Closest Members

These are the people with whom we maintain the highest socially connectivity. Most of the calls we receive, come from individuals within this category. We receive more calls from them and we tend to talk with them for longer periods. Typically, the face-to-face social tie of these people is family member, friend, and colleagues.

Group 2: Socially Near Members

People in this group are not as highly connected as family members and friends, but when we connect to them, we talk to them for considerably longer periods. Mostly, we observe intermittent frequency of calls from these people. These people are typically neighbors and distant relatives.

Group 3: Socially Distant Members

These individuals have less connection with our social life. These people call us with less frequency. We acknowledge them rarely. Among these would be, for example, a newsletter group or a private organization with whom we have previously subscribed. This group also includes individuals who have no previous interaction or communication with us. We have the least tolerance for calls from them e.g., strangers, telemarketers, fund raisers.

We quantitatively define the social closeness between user i to user j from perception of user i ($S(i,j)$) by Equation (19).

$$S(i,j) = \sqrt{(1 - F(i,j))^2 + (1 - T(i,j))^2}, \tag{19}$$

where $F(i,j)$ is the normalized call frequency (normalized to the maximum call frequency among all users with whom user i communicate) between user i and user j which is given by Equation (20), and $T(i,j)$ is the normalized call duration or talk time (normalized to the maximum talk time among all users with whom user i communicate) between user i and user j, which is given by Equation (21). The reason for normalization here is to align all associated users (callers and calles) of the user i onto a reference scale ranging from zero to one where zero means the minimum and one means the maximum. As they are on the same scale, it is thus convenient to compare in a more systematic way.

$$F(i,j) = \frac{f(i,j)}{\max\limits_{k \in U_i}\{f(i,k)\}}, \tag{20}$$

$$T(i,j) = \frac{t(i,j)}{\max\limits_{k \in U_i}\{t(i,k)\}}, \tag{21}$$

where $f(i,j)$ is the total number of calls or call frequency between user i and user j, $t(i,j)$ is the total call duration or talk time between user i and user j, and $U_i = \{1, 2, ..., N\}$ is the

set of all users associated with user i (*i.e.,* all users who have made/received calls to/from user i with total of N users).

As $F(i,j)$ and T(i,j) are normalized values that lie between zero and one, thereby $S(i,j)$ has values in the range $[0, \sqrt{2}]$, which indicates the mobile social closeness between user i and user j from user i's perspective where 0 implies the closest and $\sqrt{2}$ implies the farthest relation. Based on this quantity, we can categorize all users associated with the user i into three social groups using a simple algorithm given in Fig. 4.1. (where Fig. 4.2 shows graphical illustration).

Algorithm 1. Social Grouping

Input: Social Closeness ($S(i,j)$), $F(i,k)$, and $T(i,k)$ for $k \in U_i$

Output: Social Group that user j belongs to, according to user i ($G(i,j)$)

1. Compute μ_F and μ_T, the means of $F(i,k)$ and $T(i,k)$ respectively for $k \in U_i$
2. Compute R, the Euclidean distance from (μ_F, μ_T) to (1,1).
3. IF $S(i,j) \leq R/2$
4. $\quad G(i,j) = 1$;
5. ELSE IF $R \geq S(i,j) > R/2$
6. $\quad G(i,j) = 2$;
7. ELSE IF $S(i,j) > R$
8. $\quad G(i,j) = 3$;
9. END IF
10. Return $G(i,j)$;

FIGURE 4.1. Algorithm for social grouping.

As social closeness and social group are defined according to the perception of user i, therefore using analogy of the circle, user i can be referred to as a *center user*, where the distance from the center of the circle (center user) represents the closeness of social relationship to other associated users.

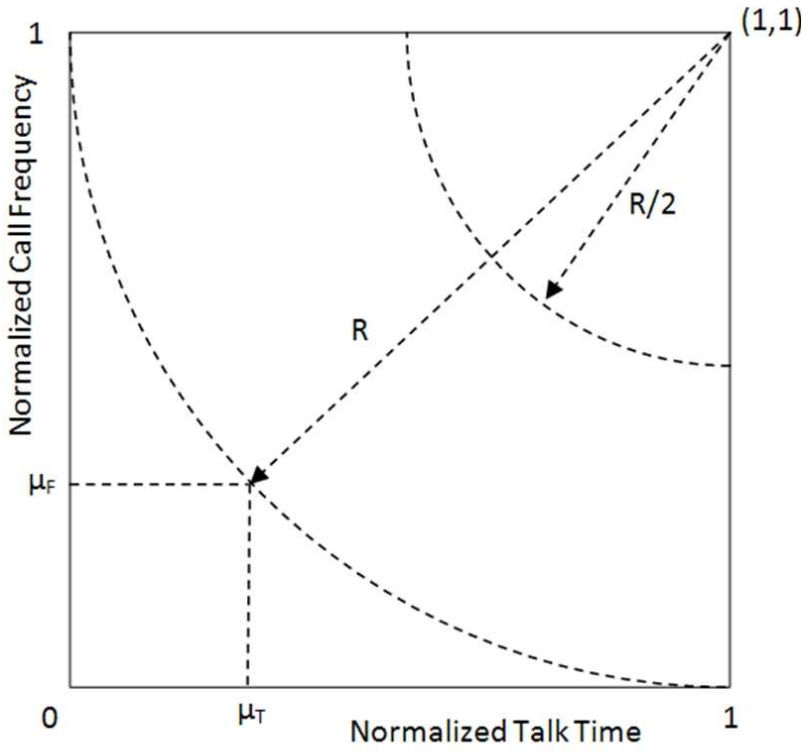


FIGURE 4.2. Graphical illustration for identifying boundaries of mobile social groups.

Property 1. *Social closeness is typically not symmetric but can be symmetric under a specific condition.*

Social group is based on social closeness which is measured by amount of time and intensity of communication between the center user and associated user. Social closeness is computed according to the center user's perception of each associated user compared to all other associated users. Since different center users may have different associated users with different amount of time and intensity of communication, thus social closeness is not symmetric. For example, user j is perceived by user i as a member of group 1, however user i is perceived as a member of group 2 of user j since user j has other associated users to whom user j communicate more than user i.

From Equations (19), (20), and (21); $S(i,j)$ can be defined as

$$S(i,j) = \sqrt{(1 - \frac{f(i,j)}{\max\limits_{k \in U_i}\{f(i,k)\}})^2 + (1 - \frac{t(i,j)}{\max\limits_{k \in U_i}\{t(i,k)\}})^2}, \tag{22}$$

and $S(j,i)$ can be defined as

$$S(j,i) = \sqrt{(1 - \frac{f(j,i)}{\max\limits_{m \in U_j}\{f(j,m)\}})^2 + (1 - \frac{t(j,i)}{\max\limits_{m \in U_j}\{t(j,m)\}})^2}. \tag{23}$$

Since $f(i,j) = f(j,i)$ and $t(i,j) = t(j,i)$, Equation (23) can be rewritten as

$$S(j,i) = \sqrt{(1 - \frac{f(i,j)}{\max\limits_{m \in U_j}\{f(j,m)\}})^2 + (1 - \frac{t(i,j)}{\max\limits_{m \in U_j}\{t(j,m)\}})^2}. \tag{24}$$

If social closeness is symmetric, *i.e.,* $S(i,j) = S(j,i)$, then

$$\sqrt{(1 - \frac{f(i,j)}{\max\limits_{k \in U_i}\{f(i,k)\}})^2 + (1 - \frac{t(i,j)}{\max\limits_{k \in U_i}\{t(i,k)\}})^2} = \sqrt{(1 - \frac{f(i,j)}{\max\limits_{m \in U_j}\{f(j,m)\}})^2 + (1 - \frac{t(i,j)}{\max\limits_{m \in U_j}\{t(j,m)\}})^2}, \tag{25}$$

where equality holds if and only if $\max\limits_{k \in U_i}\{f(i,k)\} = \max\limits_{m \in U_j}\{f(j,m)\}$ and $\max\limits_{k \in U_i}\{t(i,k)\} = \max\limits_{m \in U_j}\{t(j,m)\}$.

Note: Symmetry of the social closeness here means that the social closeness between the user i and j perceived by the user i ($S(i,j)$) is the same as the social closeness perceived between the user i and j perceived by user j ($S(j,i)$). For example, suppose there are Subj. #1 and Subj. #2 who have been communicating with each other via mobile phones so that they have established a mobile social relationship. In other words, Subj. #1 is an Associated User of Subj. #2 and Subj. #2 is also an Associated User of Subj. #2. Suppose we ask each subject (independently) to quantitatively identify the social closeness between them with a value from 0 to $\sqrt{2}$(where 0 implies the closest and $\sqrt{2}$ implies the furthest). If Subj. #1 thinks that the social closeness between him and Subj. #2 is $S(1,2)$, and Subj. #2 thinks that the social closeness between him and Subj. #1 is $S(2,1)$. We say that social closeness between Subj. #1 and Subj. #2 is "symmetric" if $S(1,2) = S(2,1)$. According to Eq. 22, the social closeness ($S(i,j)$) depends on four parameters: call frequency between user i and user j ($f(i,j)$), call duration between user i and user j ($t(i,j)$),

the maximum call frequency among all Associated Users with whom user i communicate ($\max_{k \in U_i}\{f(i,k)\}$), and the maximum call duration among all Associated Users with whom user i communicate ($\max_{k \in U_i}\{t(i,k)\}$). Likewise, the social closeness ($S(j,i)$) from the user j's perception depends on the same aforementioned parameters but from user j's perspective: $f(j,i)$, $t(j,i)$, $\max_{m \in U_j}\{f(j,m)\}$, and $\max_{m \in U_j}\{t(j,m)\}$– as given by Eq. 23. Since the call frequency between user i and user j ($f(i,j)$) is always equal to the call frequency between user j and user i ($f(j,i)$) *i.e.*, $f(i,j) = f(j,i)$, and the call duration between user i and user j ($t(i,j)$) is always equal to the call duration between user j and user i ($t(j,i)$) *i.e.*, $t(i,j) = t(j,i)$ – as shown in Eq. 24, therefore this symmetry can only occur when $\max_{k \in U_i}\{f(i,k)\} = \max_{m \in U_j}\{f(j,m)\}$ and $\max_{k \in U_i}\{t(i,k)\} = \max_{m \in U_j}\{t(j,m)\}$ – as shown in Eq. (25) . A symmetric social closeness is rare beacuse it can only happen under the mentioned condition ($\max_{k \in U_i}\{f(i,k)\} = \max_{m \in U_j}\{f(j,m)\}$ and $\max_{k \in U_i}\{t(i,k)\} = \max_{m \in U_j}\{t(j,m)\}$).

Property 2. *Social closeness and social group change over time.*

In our daily life, relationships inevitably change over time. Meeting new people with whom the closer relationships established and not keeping in touch with whom the relationships become further are part of our social life. It is inherently true in mobile social network that social closeness changes over time. Situations bring people together and take them apart. These situations can be work, school, hobby, or any event in life. As soon as the phone numbers have been exchanged or given, a new social member may arise and possibly gain closer relationship as time progresses. Thus social closeness and social group change over time.

4.2.1. Datasets

In this study, we use two sets of real-life call logs of 30 combined users with nearly 3,000 associated callers/calees and over 46,000 call activities. Our first dataset consists of three-month call logs of 20 individual mobile phone users, which were collected at University of North Texas (UNT) during summer of 2006. These 20 individuals were faculty, staff, and students. These call logs were collected as part of the Nuisance Project (105), where

Kolan et al. (105) studied the nuisance level associated with each phone call. Our second dataset consists of three-month call logs of ten mobile phone users, which were collected during summer of 2008 at UNT. These ten subjects were also faculty, staff, and student. In addition, during our second dataset collecting process, we interviewed the subjects about the social closeness for all of his/her associated users by having the subjects identifying the perceived social group for each associated user. As the result, our second dataset includes an additional information on social group corresponding to each associated user. The details of the data collecting process are described in (106). The survey is included in the Appendix B.

As part of the data collecting process for both datasets, each user downloaded three months of detail telephone call records from his/her online accounts on the mobile phone service provider's website. Each call record in the dataset had 5-tuple information as follows (an example call record is shown in Fig. 4.3):

Call record: {Date, Start time, Type, User ID, Talk time} where

- Date – date of call
- Start time – start time of call
- Type – type of call i.e., "Incoming" or "Outgoing"
- User ID – caller/callee identifier
- Talk time – duration of call (in minutes)

4.2.2. Validation of Social Grouping

To validate the accuracy of our social closeness/group computation, we use the second set of our data, which contains social group information. *We are able identify social groups correctly with the overall accuracy rate of 93.8%.* The detailed result is shown in Table 4.1, which presents the number of correct classification (Hit), the number of incorrect classification ($Miss$), and the accuracy rate ($\frac{Hit}{Hit+Miss}$) for each center user.

Based on the follow-up interviews with these ten subjects, most of "$Miss$" are caused by confusion between the face-to-face social closeness and mobile social closeness. For example, one of the subjects indentifies his roommate as a group 1 member but since the subject sees

Date	Start time	Type	User ID	Talk time
3/11/2007	2:28PM	Outgoing	123-4567890	2
3/11/2007	5:31PM	Incoming	888-8888888	11
3/11/2007	8:12PM	Incoming	999-9999999	6
…	…	…	…	…

FIGURE 4.3. An example of call record. Note that User IDs have been modified to protect privacy.

his roommate quite often thus the subject does not make/receive many phone calls to/from him. As the result, his roommate is classified to group 2 based on our calculation (Equation (19)) but identified as group 1 member by the subject. To avoid the biased feedbacks from the subjects, we do not provide any information about our social closeness computation or much more details about the three social groups than the description provided earlier in this section. Nevertheless, we believe that we have a good result in accuracy rate and, in addition, we do not have a single incorrect classification that misses more than one level of social group.

Furthermore, as stated by Property 2 that social relationships change over time. With our real-life datasets, we thus further experimentally validate Property 2 by showing an example of an actual social-group plot of a randomly selected center user from our datasets in Fig. 4.4, from which we can see that the associated user 8 used to be a member in group 1 (Fig. 4.4(a)) but as time progresses, he/she has changed calling behavior towards the center user (or the center user changes his/her calling behavior towards the associated user 8) by which furthers relationship apart and leads the user 8 to become a member of group 2 at 30 days later (Fig. 4.4(b)).

As texting becomes a popular mobile means of communication, one may be curious about how to apply the proposed computational framework to the coexistence of the texting

TABLE 4.1. The result of validation of social group calculation, which includes the number of correct/incorrect classification ($Hit/Miss$) based on our social closeness calculation and group classification, and the accuracy rate for each user

User	Hit	$Miss$	Accuracy Rate (%)
1	60	5	92.31
2	57	6	90.48
3	48	5	90.57
4	141	13	91.56
5	127	8	94.07
6	188	11	94.47
7	88	3	96.70
8	80	6	93.02
9	62	1	98.41
10	87	4	95.60
Overall	938	62	93.80
Mean	93.80	6.20	93.72
Std. Dev.	44.82	3.61	2.64

information in the call logs. According to our definition of social closeness (Eq. 19), the social closenss can be estimated based on the intensity of communication, which can be measured by call duration and call frequency. With texting, the frequency can simply be measured by the number of communications via texting (both incoming and outgoing textings). Even though there is no explicit form of duration of the texting, the length of the texting (number of characters typed) can be the counterpart of call duration. Such that the social closeness computation given in Eq. 19 can be rewritten as

$$S(i,j) = W_V\sqrt{(1-F_V(i,j))^2+(1-T_V(i,j))^2} + W_T\sqrt{(1-F_T(i,j))^2+(1-T_T(i,j))^2},$$

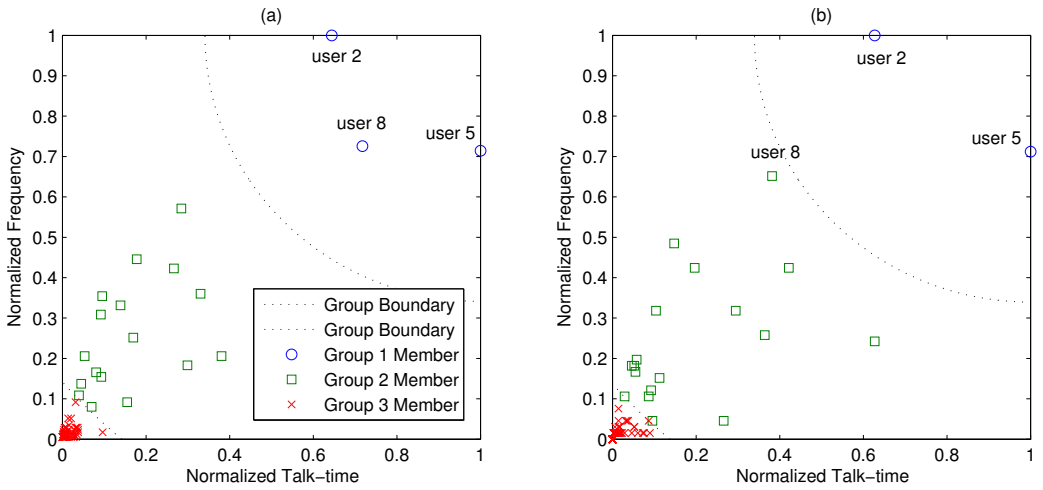


FIGURE 4.4. (a) Social relationship at time T and (b) social relationship at 30 days later $(T + 30)$.

where subscripts V and T represent Voice and Text respectively. The varaibles W_V and W_T are the the weights of voice and texting communication means, social closeness is thus estimated based on both mobile means with either same ($W_V = W_T = 0.5$) or different contributing load $W_V \neq W_T$ and $W_V + W_T = 1$) to the social closeness. These weights can be further studied and determined with the most suitable values.

One may also raise an issue of communication intensity being a subjective judgement. Clearly, it is true. This means that the definition of high and low intensity in communication completely depends on the perception of each individual subject. If we were to acquire the feedback about the communication intensity from two different subjects: Subj. #1 and Subj. #2 given that these two subjects know each other and have been communicating via mobile phones. The feedbacks from both subjects may be different because "the intensity is a subjective judgement" *e.g.*, Subj. #1 may say that he/she has high communication intensity with Subj. #2 but on the other hand, Subj. #2 may think that he/she has low communication intensity with Subj. #1. The perceived intensity levels are different because each subject makes decision on the intensity after comparing the intensity level of the other subject with other associated contacts (other persons with whom the subject has been communicating via mobile phone). Therefore, the perceived intensity is estimated

based on the subject's past communications with all contacts. The subject is thus the center (reference) point of perception. According to our study, the feedback and computation are done based on "one" reference point of view – the Center User who gives the feedback from his/her perception about the communication intensity between him/her and each of his/her Associated Users (callers/callees). We do not compare one subject's perceived intensity against another but based on one subject's perceived intensity (and computed intensity), we classify the social groups. Thereby, it is true that intensity is a subjective judgement, however it does not after our results as we take this perceived intensity as a ground truth for verifying our computation, not to argue that one subject's perceived intensity is the same as or different from another's. We consider each individual subject independently (from the others).

For a possibility of some comfusion in feedback-based evaluation of our social grouping scheme and its accuracy calculation, we note the following. The goal of our study is to construct a computational model that estimates the human's perceived mobile social tie, and then use the verified model to infer other useful characteristics of mobile social group structure. The feedback from the human subjects is the actual perception or the ground truth or the reality (see Response of Comment #4) that is used to evaluate our model. We further investigate about the group sizes and their successive ratio upon the validated social closeness and social grouping algorithm. To reemphasize on the correct and incorrect classification of our social grouping scheme, we revisit our conducted survey study process and the evaluation process. In our mobile social survey, we recruit ten mobile phone users who are faculty, staff, and students in computer science and engineering department at University of North Texas. We obtain three-month call logs from each subject who is then asked to identify his/her perceived social tie of each Associated User in the call logs. Information about definition of the social tie is given to the subject as described in Appendix C. In our validation process, for each subject (Center User), we use our social closeness and grouping model to compute the social tie for each Associated User, then compare this computed value against the actual feedback from the subject. The accuracy rate of our model is computed for each subject

as the ratio of the number of correct classified Associated Users to the total number of Associated Users. For example, suppose a given subject has five Associated Users and our model computes the social tie (group) as 1, 2, 3, 3, 3 for each Associated User respectively. Then we check these computed values against the actual perceived social tie from the subject, suppose the subject's feedback shows the social tie as 1, 2, 2, 3, 3 for each Associated User respectively. Such that the accuracy rate can be computed as (Number of corrected social tie base on our model)/(Total number of Associated Users) = 4/5 = 80%

4.3. Social Group Sizes and Scaling Ratio

Based on the social closeness and group inference in the previous section, it is straightforward to find social group sizes for any given center user.

In our social world, people who know a lot of people and have many friends are typically socially active. On the other hand, people who are socially less active tend to have smaller social network. It is inherently the case for mobile social network. Since activeness of a phone user (center user) is related to social group sizes, we define activeness of a center user by number of outgoing calls per day. Based on this definition, center users can be divided into three categories:

(1) *Low active users*: center users who have less than six outgoing calls per day.

(2) *Medium active users*: center users who have between six to ten outgoing calls per day.

(3) *High active users*: center users who have more than ten outgoing calls per day.

Table 4.2 summarizes the result of social group sizes based on our entire datasets (30 mobile phone users) by listing the mean group sizes for each social group and each category of the center users based on the activeness. It can be observed that the mean group sizes have scaling ratio of 8.

For low active users, group 1 has mean size of 1.00 ($S_1^L = 2^0$), group 2 has mean size of $8.67 \approx 8$ ($S_2^L = 2^3$), and group 3 has mean size of $63.33 \approx 64$ ($S_3^L = 2^6$). Thus, scaling ratio for low active users is approximately $\frac{S_{i+1}^L}{S_i^L} = 2^3 = 8$.

TABLE 4.2. The mean group sizes of each social group for low, medium, and high socially active center users

Social Group	Mean Group Sizes		
	Low Active Users	Medium Active Users	High Active Users
1	1.00	1.50	2.00
2	8.67	11.83	16.91
3	63.33	90.83	126.64

For medium active users, group 1 has mean size of $1.50 \approx 2^{0.5}$ ($S_1^M = 2^{0.5}$), group 2 has mean size of $8.67 \approx 2^{3.5}$ ($S_2^M = 2^{3.5}$), and group 3 has mean size of $90.83 \approx 2^{6.5}$ ($S_3^M = 2^{6.5}$). Hence scaling ratio is medium active users is approximately $\frac{S_{i+1}^M}{S_i^M} = 2^3 = 8$.

For high active users, group 1 has mean size of 2.00 ($S_1^H = 2^1$), group 2 has mean size of $16.91 \approx 16$ ($S_2^H = 2^4$), and group 3 has mean size of $126.64 \approx 128$ ($S_3^H = 2^7$). Similarly, scaling ratio for high active users is approximately $\frac{S_{i+1}^H}{S_i^H} = 2^3 = 8$.

From the results of all three categories of center users, it is very interesting to see that activeness of center user indeed reflects the social group sizes and the same scaling ratio is found for every category, that is

$$\frac{S_{i+1}^L}{S_i^L} = \frac{S_{i+1}^M}{S_i^M} = \frac{S_{i+1}^H}{S_i^H} = 8. \tag{26}$$

Besides a simple analysis based on the mean group sizes, we further employ a more systematic method of analysis that uses raw group sizes. We thus consider all 90 grouping clusters in our dataset, which are shown in Fig. 4.5 (in semi log scale) where the sample distribution can be represented as a sequence of Dirac's delta functions given by Eq. (27).

$$f(s) = \sum_{i=1}^{N} \delta\left(s - s_i\right), \tag{27}$$

where δ is Dirac's delta function and N is the number of grouping clusters.

Note that the main idea is to instead of considering the mean group sizes, take into account each individual group size such that the scaling ratio is derived from the raw data. To do so, we need to lay out our raw data and extract the pattern from which the scaling

ratio can be obtained. To lay out our data, each group size of all 90 data points (three social groups of 30 Center Users) is plotted onto a simple Center User-versus-Group Size plot (Fig. 4.5), which provides us a graphical representation of the distribution of the data. The plot is in semi log scale because the clusters of soical group sizes appear to be separated by some exponential constant. Thereby a semi log scale better represents the data distribution than a linear scale that would have depicted a non-periodic signal or less periodic signal. To extract the pattern from this data distribution plot, we choose to estimate this raw distribution with a Gaussian kernel density estimator such that the data distribution can be transformed to a probability density function (pdf) from which the scaling ratio can be obtained by extracting the periodicity of the signal (pdf).

Figure 4.6 shows the probability density function (pdf) $f(s)$ estimated by a Gaussian kernel estimator (107) with zero mean, unit variance, and AMISE optimal bandwidth selection using the Sheather Jones Solve-the-equation plug-in method (96). From Fig. 4.6, it can be observed that there are three main clusters of the local peaks of f(s) around 2, 10, and 80. These clusters represent the cummulative frquency of raw data distribution in Fig. 4 of three social groups. Eventhough the peaks seem to spread out, the three clusters can still be observed. With the obtained pdf, the challenge here is to extract a possible periodicity in the $\ln s$ variable, which is called "log-periodicity" (108), *e.g.,* if the previous scaling ratio in Equation (**??**) is true, then the periodic oscillation of $f(s)$ can be expressed in the variable $\ln s$ with the expected mean period of $\ln 8 = 2.08$. We use generalized q-analysis or (H, q)-analysis (109), which has been shown to be very efficient for finding periodicity (102). The q-analysis is a natural tool for describing discrete scale invariance (110) (111). The (H, q)-analysis consists in constructing the (H, q)-derivative, which is given by Equation (28).

$$D_q^H f(s) \underline{\underline{\Delta}} \frac{f(s) - f(qs)}{[(1-q)s]^H}. \tag{28}$$

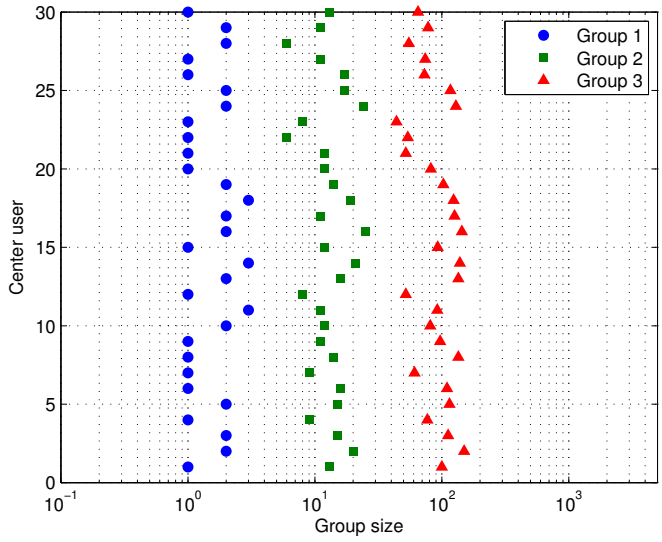


FIGURE 4.5. Distribution of group sizes in our dataset.

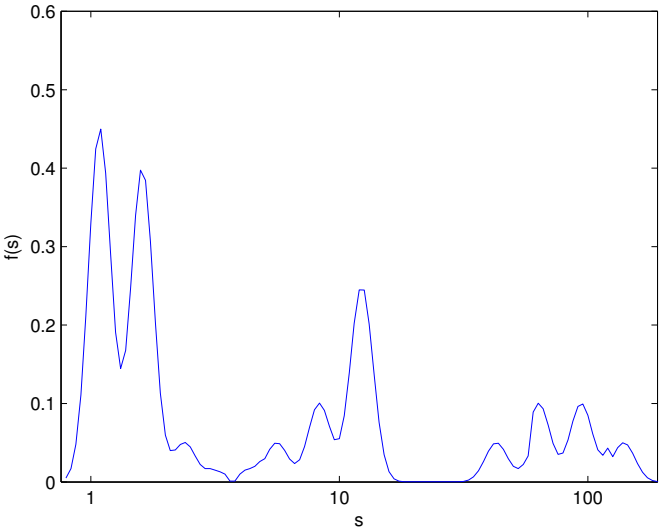


FIGURE 4.6. The pdf ($f(s)$) obtained from Gaussian kernel density estimation of group size s.

The (H, q)-derivative has two control parameters; the discrete scale factor q derived to characterize the log-periodic structure and the exponent H introduced to allow us to detrend $f(s)$ in an adaptive way.

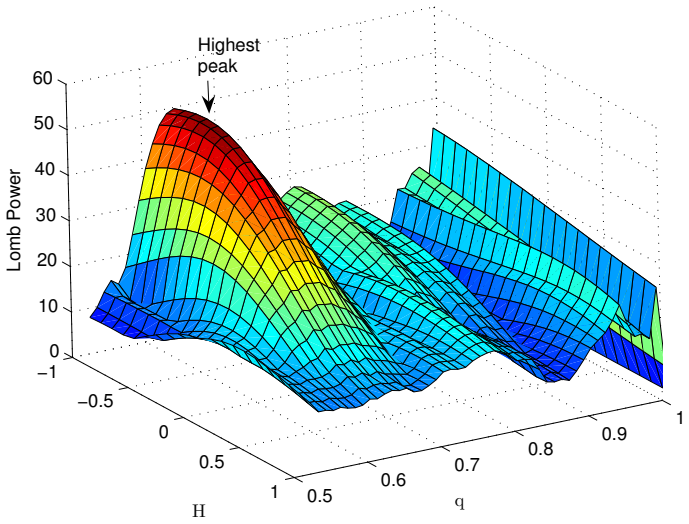


FIGURE 4.7. The highest peak of Lomb power is found at $H = -0.7$ and $q = 0.62$.

To extract the log-periodicity in $f(s)$, we then use a Lomb periodogram analysis (112). The Lomb periodogram or Lomb power $P(\omega)$ is given by Equation (29).

$$P(\omega) = \frac{1}{2\sigma^2}\left\{\frac{\left[\sum_s f(s)\cos\omega(s-\tau(\omega))\right]^2}{\sum_s \cos^2\omega(s-\tau(\omega))} + \frac{\left[\sum_s f(s)\sin\omega(s-\tau(\omega))\right]^2}{\sum_s \sin^2\omega(s-\tau(\omega))}\right\}, \tag{29}$$

where σ^2 is the variance of $f(s)$ and $\tau(\omega)$ is given by Equation (30).

$$\tau(\omega) = \frac{1}{2\omega}\arctan\left\{\frac{\sum_s \sin 2\omega s}{\sum_s \cos 2\omega s}\right\}. \tag{30}$$

We test for the statistical significance of possible log-periodic oscillations. For each (H, q) pair, the highest peak $P(H, q)$ and its associated angular log-frequency $\omega(H, q)$ in the Lomb periodogram are obtained. The basic criterion used to identify a log-periodic signal is the strength of the Lomb periodogram analysis, *i.e.,* the height of the spectral peaks. Figure 4.9 presents the Lomb periodograms of the (H, q)-derivative $D_q^H f(s)$ for different pairs of (H, q) with $-1.0 \leq H \leq 1.0$ and $0.5 \leq q \leq 1.0$. The highest Lomb power is found at $H = -0.7$ and $q = 0.62$ (shown Fig. 4.7) where its $D_q^H f(s)$ is shown in Fig. 4.8. The highest peak is at $\omega = 2.99$ with Lomb power of 53.25. *The preferred scaling ratio is thus* $\lambda = e^{2\pi/\omega} = 8.17 \approx 8$*, which is consistent with the previous result using mean group sizes.*

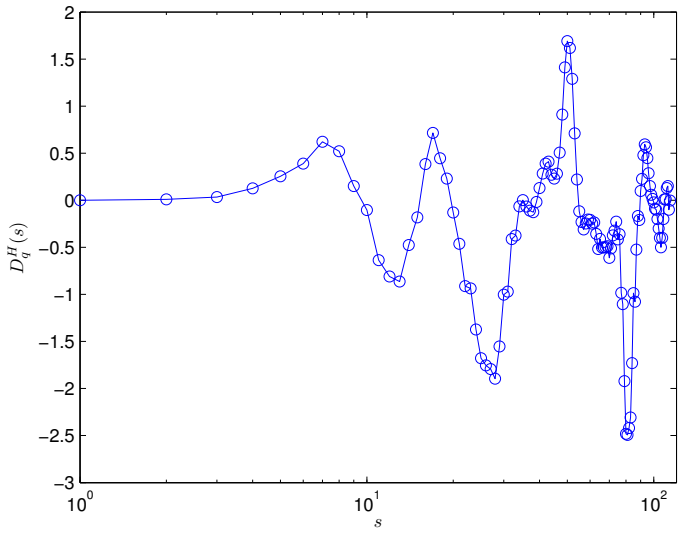


FIGURE 4.8. The (H, q)-derivative $D_q^H f(s)$ as a function of group size s with $H = -0.7$ and $q = 0.62$.

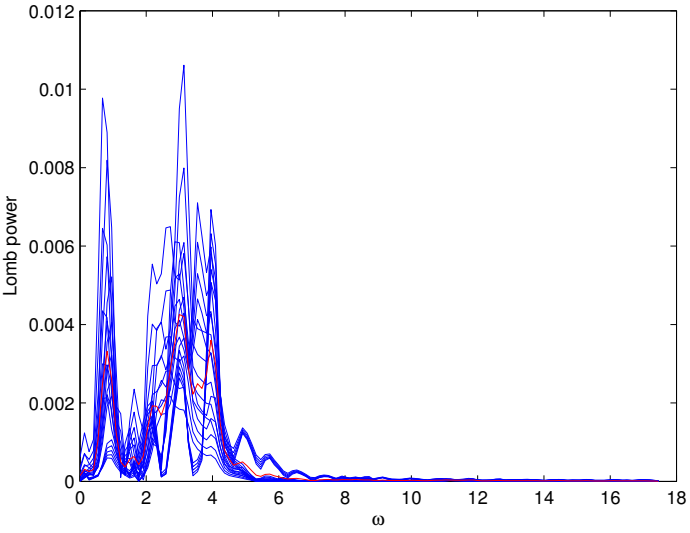


FIGURE 4.9. Lomb power as a function of angular log-frequency ω of the (H, q)-derivative $D_q^H f(s)$ for different pairs of (H, q) where the red line indicates the average of Lomb power.

Note for the readers who are not familiar with the generalized q-analysis: The goal of applying the generalized q-analysis here is to extract the (most probable) periodicity of the signal ($f(s)$, shown in Fig. 4.6) obtained from the raw data of social group sizes (shown

in Fig. 4.5). The criterion used to identify the most probable periodicity is the strength of the Lomb power (given by Eq. 29). It appears in Fig. 4.7 that the highest Lomb power is found at H = -0.7 and q = 0.62 where its corresponding angular log-frequency is $\omega(H = -0.7, q = 0.62) = 2.99$(shown in Fig. 4.9). Therefore, the log-periodicity is $\ln \lambda = 2\pi\omega \rightarrow \lambda = e^{2\pi\omega} = e^{2\pi(2.99)} = 8.17 \approx 8$.

4.4. Related Work

Closeness in face-to-face social networks has been studied in psychology, from which various definitions (113) (114) (115), components (116) (117), classifications of closeness (118) (119), and social support (120) have been defined.

As online social networking is gaining popularity, online social analysis has also been extensively studied and the results have been reported in several literatures, among which discussed about social closeness in online communities (121) (122) (123). To our knowledge, no scientific research has been reported in quantifying closeness in mobile social networks.

Mobile social closeness has been mentioned to be an important component of interaction syntax for mobile social software in (124) but never once defined. A literature that has come close to defining mobile social closeness is (125), in which the authors measured the closeness centrality for mobile phone users based on the definition proposed by Freeman in (126).

There have been research studies in social group sizes and scaling ratio in sociology (98), social anthropology (99) (100), and psychology (127) (102) in face-to-face social networks but not in mobile social networks.

4.5. Discussion

Social networking is a process of initiating, developing, and maintaining the relationships. With the advance in our technology, we are now interacting with people in online and mobile networks besides the conventional face-to-face social networks. Despite the different setups, these three networks share common members *e.g.,* we often have friends with whom we contact in the face-to-face network as well as in online and mobile network. Figure 4.10 shows a Venn diagram of these social networks that shares common members in the

overlapping areas. Since almost all of the mobile members are initiated through the face-to-face networks, the overlapping area between the mobile and face-to-face network is relatively larger than the overlapping area between the online and the face-to-face network where several online social members are people with whom we have never met in person. However, we occasionally communicate with people on the mobile phone with whom we have never met (*e.g.,* job interviews by phone, customer service calls, etc.), which thus results in a small non-overlapping area between the mobile and face-to-face network.

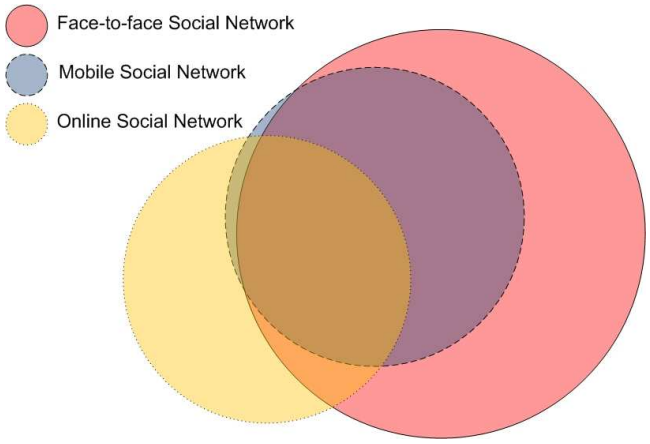


FIGURE 4.10. Venn diagram of three social networks

Mobile social relationships (mostly) are initiated through the face-to-face social networks. Mobile social relationships are developed and maintained with the intensity of communication, which also strengthen the face-to-face relationships. Thus, despite a small non-overlapping area between the mobile and face-to-face social network, mobile social network is (roughly) a subset of the face-to-face social network.

For face-to-face social networks, Zhou *et al.* (102) found a scaling ratio close to "3" based on the results of the previous studies of social grouping (127)(128)(129), which divided social members into six groups with different group sizes as described in Table 4.3.

According to Dunbar's number (101) ("150"), which indicates the number of individuals with whom a stable inter-personal relationship can be maintained, we thus restrict our

TABLE 4.3. Face-to-face social grouping

Group	Name	Group size	Description
1	Support clique ((128)(129))	3-5	A group of individuals from whom the subject would ask personal advice or help in times of severe emotional and financial distress.
2	Sympathy group ((128)(129))	12-20	A group of individuals with whom the subject has special tie; these individuals are typically contacted at least once a month.
3	Overnight camp or Band ((127))	30-50	A group of individuals from whom the subject feels a personal allegiance at a given time.
4	Clan or Regional group ((127))	150	A group of individuals with whom the subject maintains a coherent personal relationship.
5	Megaband ((127))	500	A group of individuals with whom the subject maintains distant relationship.
6	Tribe ((127))	1,000-2,000	A group of individuals with whom the subject maintains the furthest distant relationship.

attention to only the face-to-face social group 1 to group 4 for comparison with our findings in this study.

Let F_g and M_g denote the face-to-face social group g and the mobile social group g, respectively. Figure 4.11 shows the group sizes of the face-to-face network comparing to the mobile social network. From this comparison, it is straightforward to see that

(1) $M_1 \subset F_1$

(2) $M_2 \subset F_1 \cup F_2$

(3) $M_3 \subset F_2 \cup F_3 \cup F_4$

Therefore, we conclude our discussion that the mobile social network is a subset of the face-to-face social network, and both groupings are not necessary the same (but relationships can still be drawn, as shown above), hence the scaling ratios are distinct ("3" for face-to-face and "8" for mobile social grouping).

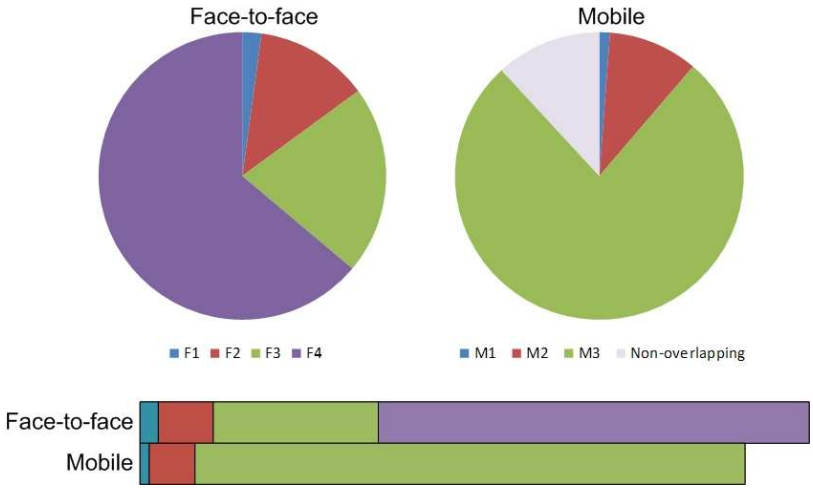


FIGURE 4.11. Comparison of group sizes between face-to-face and mobile social network

4.6. Societal Context

In this study, mobile social networking has been structured into three discrete hierarchies. Group sizes and their successive ratios are presented. Group sizes vary depending on the activeness of the users, but the ratio is nearly constant (close to eight). The findings of this study can be beneficial to:

(1) *Mobile phone service providers*: With the increase of mobile phone user population, mobile phone service providers are competing to offer better services and plans for their existing and especially potential customers. The success of "T-Mobile myFaves" (130) (plan that allows the user to make unlimited calling to his/her five favorite people) suggests that the emphasis of the future mobile phone services will be on social context. Thereby ability to recognize mobile social groups and sizes can indeed enrich the services *e.g.,* personalized plans, per-social-group rates, active/non-active social plans, etc.

(2) *Privacy settings*: Privacy concerns are rising as today's telecommunication technologies allow people to be connected pervasively (131). Mobile phone becomes more than just a voice communication device but camera, book, Internet browser, and so on. With the user unaware, information being shared on a connected network can be sensitive and private. Thus, an ability to recognize the user's social context can facilitate a context-aware mobile phone (132) to configure privacy level appropriately.

(3) *Anomaly detection*: Mobile phone calls form a communication network. Anomaly detection is to identify abnormal behavior occurring in the network. Anomalies in the network usually mean frauds, congestion, or even terrorism. Social context can be used to enhance anomaly detecting methods such as link-based (133) and rule-based (134) mechanism.

(4) *Phone call filtering*: With a flexibility of comfort and ease of use, mobile telephony is widely preferred mode over other communication modes e.g., e-mail, face-to-face interaction. However, this ease of use in real-time communication brings challenges that are not really pertinent in e-mail communications and face-to-face interactions. One problem that mobile users experience is spam and unwanted calls (135). These spam and unwanted calls can be mitigated by using social grouping scheme to develop a protocol to allow/block phone calls based on social context (136)(105).

(5) *Epidemiology*: Today's mobile phones provide convenience by integrating traditional telephony with handheld computing devices. However, the flexibility of running third-party software also leaves the phone open to malicious viruses. In fact, in the past few years hundreds of mobile phone viruses have emerged and spread through various means such as SMS/MMS, Bluetooth, and traditional IP-based applications (137). Integrating social group scaling ratio into the epidemiologic model can help predict and estimate the spread of virus outbreaks. Recognition of social context can also alleviate vulnerability of becoming infected.

(6) *Business marketing*: Acquiring new clients is one of the top priority in a business. Marketing is a process to communicate to individuals and communities about the existing and new products and services. To increase its effectiveness, social context of the existing clients can be used to guide the direction of the marketing while maintaining the marketing cost-efficiency. Social context has shown its positive impact on marketing in previous studies in psychology and marketing research (138) (139) (140) (141).

4.7. Limitations of the Study

Nevertheless, there are some limitations of this study, which can be pointed as follows:

(1) *Diversity of the subjects*: Our subjects were faculty, staff, and students, which present homogeneous subjects. The result would be more generalized with more diverse subjects (*e.g.,* subjects with different backgrounds and life styles).

(2) *Amount of data for analysis*: Our analysis is based on the mobile phone's call logs over the course of three months. With the amount of call logs grows to four months, five months, six months, and so on, the number of associated users also increases as new social relationships are initiated. This limited amount of call logs does not allow us to further study the impact of increase of the new social relationships to the social group sizes and scaling ratio. On the other hand, as stated by Property 2 that social closeness and social tie change over time, it is very interesting to investigate

on what indicates the current relationships and with these current relationships, would the group sizes and scaling ratio remain unchanged?

(3) *Sample size*: It is difficult to collect these call logs due to the privacy issues and the subject's unwillingness to participate in the survey due to the time-consuming process. With our 30 mobile users, it might not completely represent the actual mobile social networks but we believe that it is the first step towards further analysis in this research study. Nonetheless, we will continue to collect more datasets for our future studies.

(4) *Characteristic of Associated Users*: The call duration may depend on the characteristic of the Associated Users (callers or callees) *e.g.*, discursive, talkative, or cryptic. Even though the social closeness computation is from the Center Users perspective that means the duration of each call is influenced by the willingness of the Center User, there are also call durations that are quite extensive and exceeding the Centere User's willingness. These calls are typical as we all may have experienced in our daily lives. Undoubtedly we believe that such calls exist in our dataset. We also believe that the amount of these calls are relatively small because the Center User usually learns from one or few of these calls and would try to avoid the similar situation (spending undesired long period of time on the phone talking with (listening to) the persons). We are aware of these calls and we take them as noises or outliers in our dataset. Since the characteristic of the Associated Users is not included for analysis in our survey study, it thus notes another limitation of our study that we would like to address in the future work.

(5) *Diversity of situaional contexts*: Sine our data are drawn from typical normal mobile phone users (ordinary lives under no significant political situations or any extraordinary circumstances e.g., natural disasters, big social/economical influences, etc.), thereby the impact of these extraordinary circumstances is not evidenced and analyzed in our present study. For example, suppose there is a subject who is under an unusual situation such as political movements. He could be communicating with

several new people (out of his usual daily life style). Intuitively, he forms new relationships. According to our model, he establishes social ties. These new established ties would become his current social ties within his current situational context. Therefore, instead of what it should have been in his normal situational context of 3, 24, and 192 as number of members in his group 1, 2, and 3 respectively, it may be 6, 24, and 96 or else in this current abnormal situational context. We speculate that with a wide range of diverse situational contexts, our model can still be applied. However, the result in group sizes and scaling ratio might be different. The grouping scheme could also be diffident as well. This is a very interesting issue to be further investigated in our future work.

4.8. Conclusion

With the rapidly growing population of mobile device users, more new mobile social services are being offered. Research and development in mobile social computing are thus intensified. In this article, we present a simple but efficient method to quantitatively define mobile social closeness, which is then used to categorize mobile social network into three groups. Our social grouping approach has been validated with the real-life datasets with a high accuracy rate. With our mobile social grouping results, we identify a group sizes' scaling ratio of close to "8" based on two different analyses where one is based on mean group sizes and the other is based on all raw group clusters. We carry out a discussion on social networks. We point out the overlapping area (common members) between face-to-face, online, and mobie social networks and draw a conclussion based on our findings that the mobile social network is a subset of the face-to-face social network, where both have distinct groupings and constant group sizes' scaling ratios. In societal context, we show that our findings can be benefitial to mobile phone service providers, privacy setting, anomaly detection, phone call filtering, epidemiology, and business marketing.

Nevertheless, there are limitations in our study. Diversity of the study subject's background is one the limitations since all of our subjects are faculty, staff, and students in the department of computer science and engineering. Their similar backgrounds thus bound the

generalization of our results. The amount of data for analysis is also crucial as described in Property 2, thereby larger data (longer call logs e.g., six months, one year, etc.) would allow us to study the impact of the increase of the new social relationships to the social group sizes and scaling ratio. The characteristic of the Associated Users (callers/callees) also plays an important role in social closeness computation as it does influence some call durations. Without a survey study of the characteristic of the Associated Users, our findings are lacking in this aspect. The privacy issue is the biggest obstacle in our survey study. We find that it is very difficult to find a subject who is willing to share his/her calll logs and provide a feeback about social relationships. Our study is therefore limited to 30 subjects who might not completely be a representive of the entire mobile social networks but we believe that they are a part of the first step towards further analysis in this research study.

As our future direction, we will continue to investigate on the correlation between the mobile social group sizes and scaling ratio, and the increase of initiation of new social relationship as time progresses. We will investigate on what and how to characterize the current social relationships, in which we believe to play an important role in identifying the underlying correlation.

CHAPTER 5

MOBILE SOCIAL CONTEXT AND COMMUNICATION PATTERNS

5.1. Introduction

Concepts and techniques have been recently proposed to analyze complex systems to provide new insight into the structure of social networks. Uncovering the communication pattern in such networks is a key issue to characterize them. I investigate communication patterns in mobile social networks based on my previous findings (142) of mobile social closeness and grouping scheme.

5.2. Datasets

In this study, I use two sets of real-life call logs of 30 combined users with nearly 3,00 associated callers/calees and over 46,000 call activities. My dataset consists of three-month call logs of 30 individual mobile phone users, which were collected at University of North Texas (UNT) during summer of 2006 and summer of 2008. As part of the data collecting process, each user downloaded three months of detail telephone call records from his/her online accounts on the mobile phone service provider's website. Each call record in the dataset had 5-tuple information; Date, Start time, Type (Incoming or Outgoing), User ID, and Talk time.

5.3. Mobile Social Closeness and Grouping

In our daily life, we communicate with people in the mobile network at different instances. These people constitute our mobile social network. In my previous work (142), I showed that based on amount of time and intensity of communication with these people, our mobile social network could be divided into three broad groups:

Group 1: Socially Closest Members – These are the people with whom we maintain the highest socially connectivity. Most of the calls we receive, come from individuals within this

category. We receive more calls from them and we tend to talk with them for longer periods. Typically, the face-to-face social tie of these people is family member, friend, and colleagues.

Group 2: Socially Near Members – People in this group are not as highly connected as family members and friends, but when we connect to them, we talk to them for considerably longer periods. Mostly, we observe intermittent frequency of calls from these people. These people are typically neighbors and distant relatives.

Group 3: Socially Distant Members – These individuals have less connection with our social life. These people call us with less frequency. We acknowledge them rarely. Among these would be, for example, a newsletter group or a private organization with whom we have previously subscribed. This group also includes individuals who have no previous interaction or communication with us. We have the least tolerance for calls from them *e.g.,* strangers, telemarketers, fund raisers.

Social closeness between user i and user j from the user i's perception ($S(i,j)$) can be computed by Eq. 31.

$$S(i,j) = \sqrt{(1 - F(i,j))^2 + (1 - T(i,j))^2}, \tag{31}$$

where $F(i,j)$ is the normalized call frequency (normalized to the maximum call frequency among all users with whom user i communicate) between user i and user j, which is given by Eq. 32, and $T(i,j)$ is the normalized call duration or talk time (normalized to the maximum talk time among all users with whom user i communicate) between user i and user j, which is given by Eq. 33.

$$F(i,j) = \frac{f(i,j)}{\max\limits_{k \in U_i}\{f(i,k)\}}, \tag{32}$$

$$T(i,j) = \frac{t(i,j)}{\max\limits_{k \in U_i}\{t(i,k)\}}, \tag{33}$$

where $f(i,j)$ is the total number of calls or call frequency between user i and user j, $t(i,j)$ is the total call duration or talk time between user i and user j, and $U_i = \{1, 2, ..., N\}$ is the set of all users associated with user i (*i.e.,* all users who have made/received calls to/from user i with total of N users).

5.4. Similarity in Calling Patterns

Daily, we receive calls from our social group members. Every per exhibits a unique calling pattern. I have shown in my previous work (143) that the calling patterns (of Associated Users) can be characterized by arrival time of the calls using kernel density estimator. I believe that not only the calling pattern of each Associated User is unique but the calling pattern from Center User to each Associated User is also unique. I also believe that there exists some similarity in calling patterns between an Associated User and the Center User, which may lead to a correlation between similarity in calling patterns and social closeness.

Calling pattern from user i to user j can be represented by Gaussian kernel estimation as

$$C_{i,j}(t) = f(h_{i,j}[n]), \tag{34}$$

where $f(\cdot)$ is the Gaussian kernel estimator, and $h_{i,j}[n]$ is a histogram function of arrival time of calls from user i to user j where $n = \{1, 2, 3, \ldots, 24\}$ is the hour slot.

For a given Center User i, Fig. 5.1 and Fig. 5.2 show three different calling pattern pairs; Fig. 5.2(a) shows calling pattern from Center User i to Associated User a (outgoing calls to a), $C_{i,a}(t)$ and calling pattern from Associated User a to Center User i (incoming calls from a), $C_{a,i}(t)$ where user a is member of social group 1, Fig. 5.2(b) shows $C_{i,b}(t)$ and $C_{b,i}(t)$ where user b is a member of group 2, and Fig. 5.2(c) shows $C_{i,c}(t)$ and $C_{c,i}(t)$ where user c belongs to group 3. By visual inspection, one can observe that there is more similarity in calling patterns between Center User i and member of group 1 than Center User i and member of group 2, and even more as to compare to similarity between user i and member of group 3.

I compute the similarity in calling patterns between user i and user j ($Sim(i, j)$) based on Hellinger distance as follows.

$$Sim(i, j) = 1 - d_H^2(C_{i,j}(t), C_{j,i}(t)). \tag{35}$$

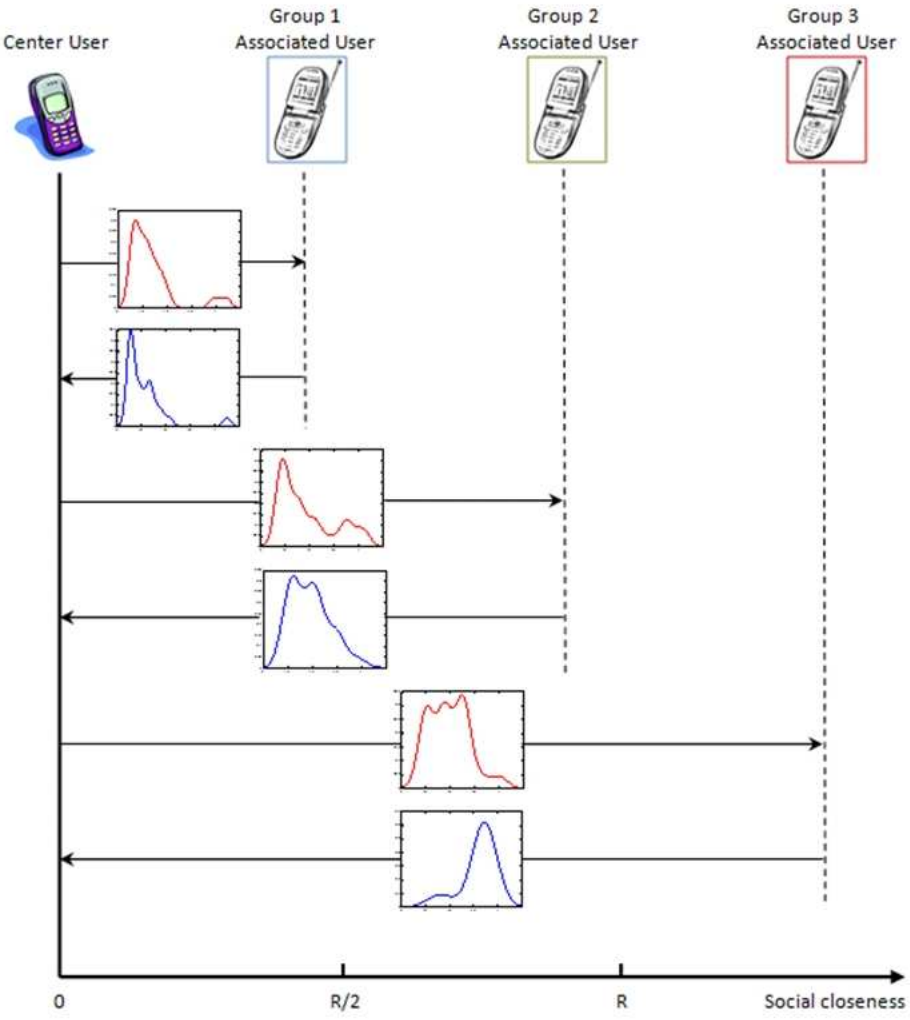


FIGURE 5.1. Calling patterns (outgoing patterns are in red and incoming patterns are in blue, the direction of the calling pattern can also be determined by the arrow) between a Center User i and three different Associated Users who are members of social group 1, 2, and 3.

Based on Eq. 35, I find that $Sim(i, a) = 0.766$, $Sim(i, b) = 0.452$, and $Sim(i, c) = 0.125$, which confirm my observation.

For each Center User in my datasets, I compute similarity in calling patterns ($Sim(i, j)$) and social closeness ($S(i, j)$) for all Associated Users and then find their averages for each

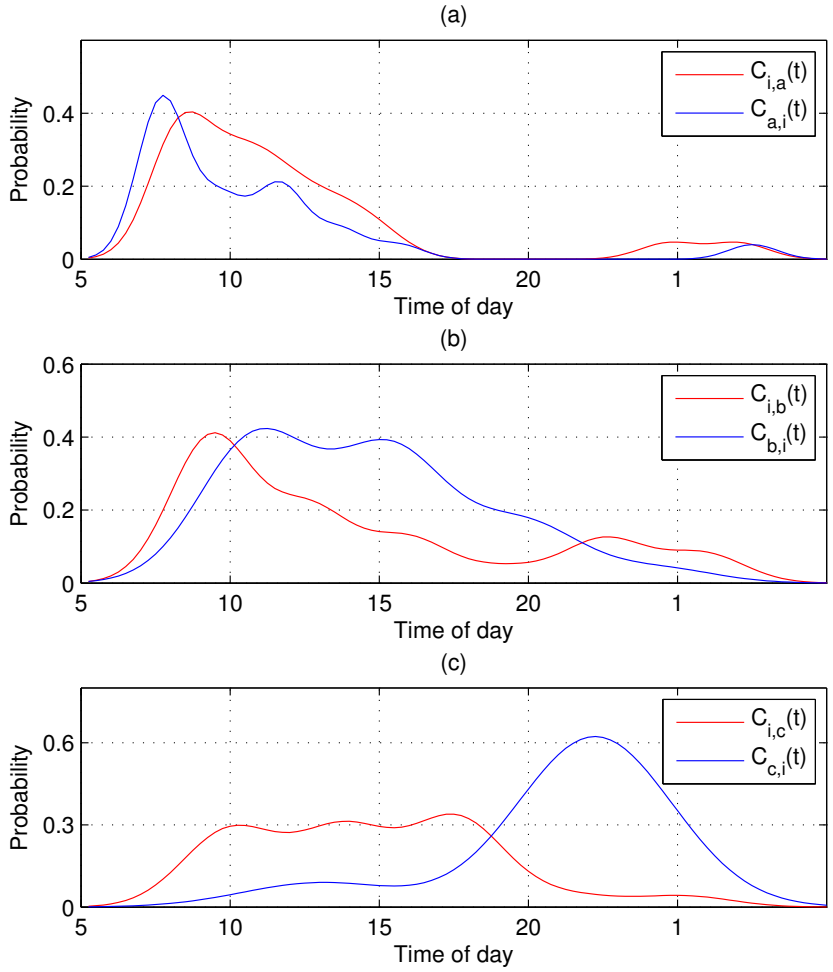


FIGURE 5.2. Calling pattern comparisons between Center User i to (a) member of social group 1, (b) member of social group 2, and (c) member of social group 3; where $C_{i,j}(t)$ is the calling pattern from user i to user j.

social group. The result is shown in Fig. 5.3(a) where I can observe that as social closeness becomes more distant, the similarity level in calling patterns decreases. This relationship can also be estimated by a fitting curve of 5th degree polynomial. In addition, Fig. 5.3(b) shows the average similarity level in calling patterns for each social group. This result is consistent with result shown in Fig. 5.3(a) where group 1 is clustered around similarity level of 0.8, group 2 is clustered around 0.4, and group 3 clustered around 0.1.

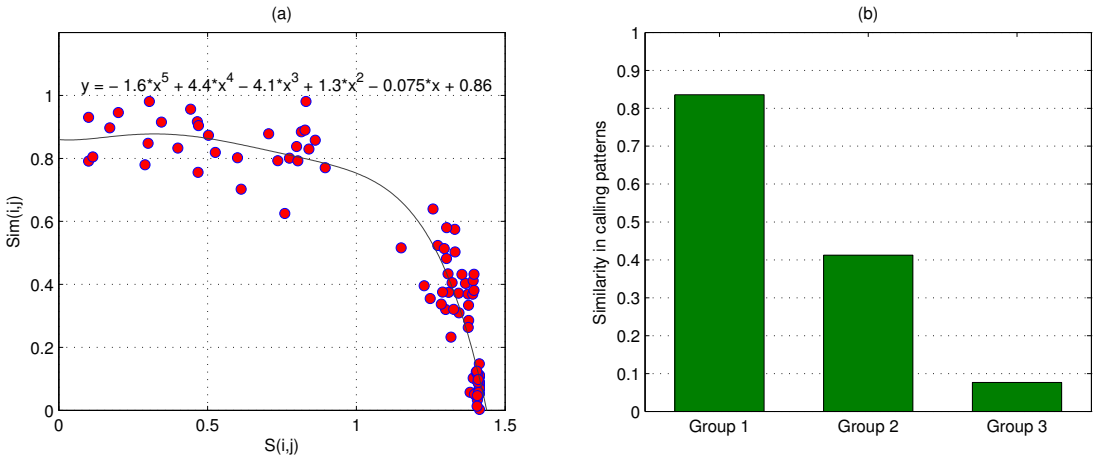


FIGURE 5.3. (a) Similarity level in calling patterns and the corresponding social closeness, (b) Similarity level in calling patterns and the corresponding social groups.

In addition to my analysis of correlation between similarity in calling patterns and social closeness/groups, I quantify *Reciprocity* ($R(i,j)$) as a level of interaction between the Center User i and an Associated User j as follows.

$$R(i,j) = F(i,j) \cdot I(i,j), \tag{36}$$

where

$$I(i,j) = -\frac{f_{in}(i,j)}{f(i,j)} \log_2 \left(\frac{f_{in}(i,j)}{f(i,j)} \right) - \frac{f_{out}(i,j)}{f(i,j)} \log_2 \left(\frac{f_{out}(i,j)}{f(i,j)} \right), \tag{37}$$

$F(i,j)$ is the normalized call frequency defined earlier using Eq. 32, $f_{in}(i,j)$ is the total number of incoming calls from Associated User j to Center User i, $f_{out}(i,j)$ is the total number of outgoing calls from Center User i to Associated User j, and $f(i,j) = f_{in}(i,j) + f_{out}(i,j)$. $R(i,j)$ has value in the range of zero to one, where $R(i,j) = 1$ implies the highest reciprocity (level of interaction) between user i and user j, and $R(i,j) = 0$ implies no reciprocity.

$R(i,j)$ is a product of a normalized call frequency ($F(i,j)$) and an *Interaction Ratio* ($I(i,j)$). The $F(i,j)$ indicates a level of interaction based on total number of calls between Center User i and Associated User j with respect to all other Associated Users, $I(i,j)$

quantifies the interaction level based on the number of exchanged calls between the two users. This value lies between zero and one ([0, 1]). Fig. 5.4(a) depicts the graph of the function $I(i, j)$.

After computing $R(i, j)$ for all Center Users in my datasets, I find hat the closer the relationship, the higher the reciprocity between the Center User and the Associated User, Fig. 5.4(b) indicates that reciprocity increases as social closeness becomes stronger. The average reciprocity is 0.8442, 0.1008, and 0.0035 for groups 1, 2, and 3, respectively. These results also imply that similarity in calling patterns increases with reciprocity.

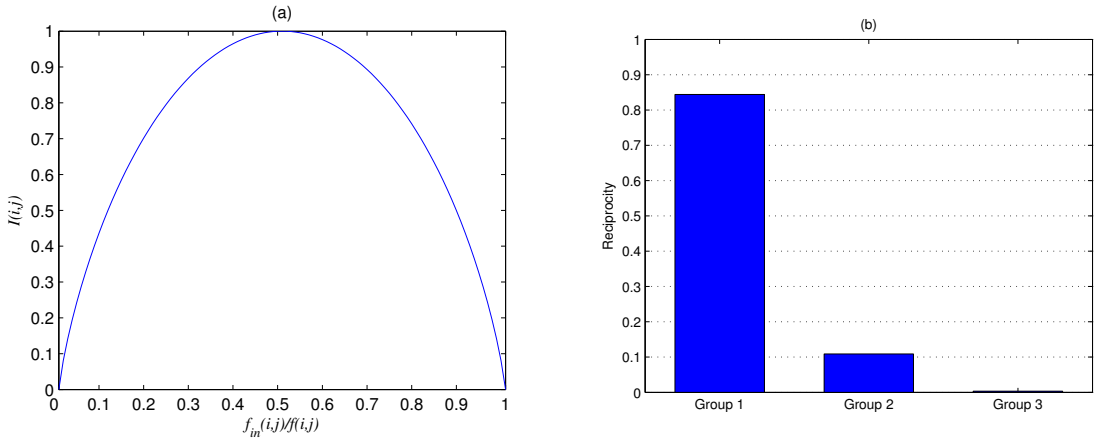


FIGURE 5.4. (a) Graph of function $I(i, j)$ versus $\frac{f_{in}(i,j)}{f(i,j)}$, (b) Integration ratio and the social groups.

5.5. Talk Time and Inter-Contact Time

Mobile phones have become indispensable within our social lives. We may spend differing amounts of time talking during each phone call depending on situation, conversation, person, and mood; however, in each instance, making or taking calls provides ways to establish, maintain, and enrich our social ties with associated persons (based on Eq. 31). Making and taking calls provides us ways to persist in our social ties (144).

I believe a correlation exits between talk time and inter-contact time. Making or taking phone call is influenced by both recent talk time and social ties. To investigate this, I generated histograms of the talk time (in minutes) versus time until the next call (incoming

or outgoing) (in hours) for my entire dataset and Fig. 5.5, which is the average histogram of all Center Users, presents my results for each of the three social group members. .

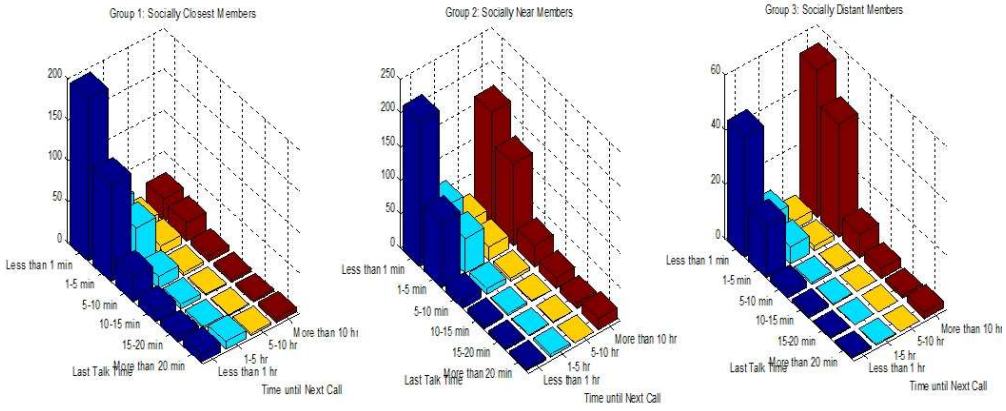


FIGURE 5.5. The histograms of the last talk time (minutes) versus time until the next call (hours), averaged over all Center Users.

For the three social groups, there are more calls for shorter talk time, which is consistent with my previous finding that talk time has exponential distribution (145). In addition, it is clear that social group plays an important role in determining inter-contact time. My results indicate that the inter-contact time increases as social closeness becomes distant. To account for this, I developed a naive Bayesian classifier conditioned on social group. Using two months of data from all Center Users to train the model, I were able to determine the time until the next contact (less than 1 hour, within the next 1-5 hours, within the next 5-10 hours, or more than 10 hours) based on the last talk time (less than 1 minute, between 1-5 minutes, between 5-10 minutes, between 15-20 minutes, and more than 20 minutes) with close to 70% accuracy over a following month.

5.6. Mobile Social Tie Prediction

Social closeness has been defined as an amount of communication intensity between the Center User and Associated User. This intensity indicates social strength and tie. As I showed in my previous work (142) that this social tie changes over time. Based on my results, I believe that I can also predict this social tie. Social ties can be indicated by social

group, which can be thought of as a state of social closeness at a given time. Change of belonging social group of an Associated User can also be thought of as a change of state of social closeness to the Center User. In my experiments, to predict social tie (group), I developd a discrete Markov chain model (146) (illustrated in Fig. 5.6), where the social group corresponds to the state of the process ($S = \{1, 2, 3\}$). The transition probability matrix (P) is given by Eq. 38, where p_{ij} is the transition probability from state i to j. The fundamental property of Markov model is the dependency on the previous state (*i.e.* future state only depends on the present state) that is $P[S(t+1) = s_{t+1} j S(t), S(t-1), S(t-2), \ldots] = P[S(t+1) = s_{t+1} j S(t)]$, where $S(t)$ corresponds to the state at time t.

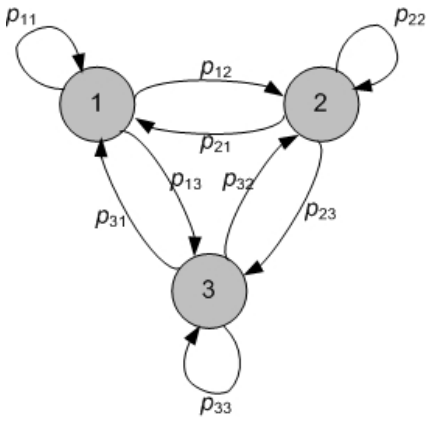


FIGURE 5.6. Discrete Markov chain model for social tie prediction with Markov states represent social groups.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}. \tag{38}$$

Eq. 39 gives the steady state probability matrix (Π) where π_i is the steady state probability of state i, which is given by Eq. 40.

$$\Pi = \{\pi_1, \pi_2, \pi_3\}. \tag{39}$$

$$\pi_i = p_{1i}\pi_1 + p_{2i}\pi_2 + p_{3i}\pi_3. \tag{40}$$

I can estimate the matrix P using many methods. Without loss of generality, the maximum likelihood principle is applied to estimate P and Π. Each element of P and Π can be estimated by Eq. 41 and Eq. 42, respectively.

$$p_{ij} = \frac{C(i,j)}{\sum_k C(i,k)}, \tag{41}$$

$$\pi_i = \frac{C(i)}{\sum_k C(k)}, \tag{42}$$

where $C(i,j)$ is the count of the number of times state j follows state i in the training data, and $C(i)$ is the count of the number of times at state i. After using the first six weeks of data for training and the following two weeks for testing with two-week time unit (matrix P and Π are sequentially recomputed every two weeks), I correctly determined social groups for the following two weeks for all Associated Users for each Center User with 96.09% overall average accuracy rate. Note that only Associated Users who established social ties with the Center User within the first two weeks were considered in this experiment This ensured that there was the same amount of training data for each Associated User (six weeks). The result for each user is shown in Table 5.1.

5.7. Mobile Life Pattern Prediction

Much like web-based social networking, people virtually meet other people in mobile world . Mobile life, then, exists as long as the user stays connected by either talking or listening. I argue that each mobile phone user exhibits a unique mobile life pattern, which reveals collective character, behavioral tendencies, and temperamental traits of that person in the mobile world. This pattern can also hint at the real-life schedule and availability of that person. I also argue that I can predict this mobile life pattern.

Like other human behavioral patterns, mobile life pattern are periodic and changing. Eagle and Pentland (81) uncovered two fundamental frequencies of human social behavior, onebeing strongest at 24 hours (1 day) and the second strongest at 7 days. Although a 1-day period does not provide an ample sample size, I can detect mobile life patterns for 7-day period and then use these patterns as a training data to predict the mobile life pattern over a subsequent week. For each week, I constructed a histogram of *on-air time* over an hour-of-the-day scale (24-hour time slots). Then, I obtained a mobile life pattern by applying Gaussian kernel estimation (see Fig. 5.7).

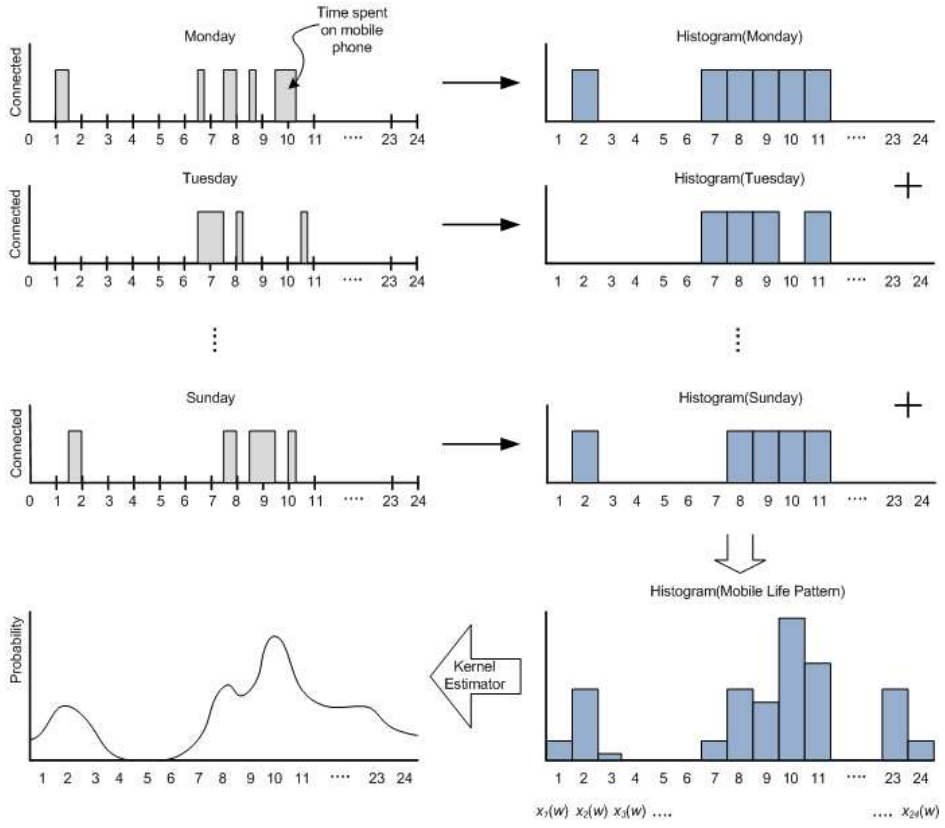


FIGURE 5.7. Process of obtaining a mobile life pattern.

Let a matrix $X(w)$ contain histogram values $\{x_1(w), x_2(w), x_3(w), \ldots, x_{24}(w)\}$, where $x_t(w)$ denotes a histogram value of t^{th} hour slot of w^{th} week, $t = \{1, 2, 3, \ldots, 24\}$, and $w = \{1, 2, 3, \ldots, n\}$ where n is the total number of weeks of training data, as follows.

$$X(w) = \begin{bmatrix} x_1(w) \\ x_2(w) \\ x_3(w) \\ \vdots \\ x_{24}(w) \end{bmatrix}. \tag{43}$$

Let Z_n be a matrix that contains $\{X(w)\}$ as follows.

$$Z_n = \begin{bmatrix} X(1) & X(2) & X(3) & \cdots & X(n) \end{bmatrix}. \tag{44}$$

My goal here is to predict a matrix $X(n+1)$ based on Z_n. A normal distribution can be fairly assumed for $x_t(w)$ for each time slot t, such that a Moving Average (MA) model is suitable for the task. Eq. 45 provides a simple MA model.

$$x_t(w) = c_t + \varepsilon_t(w), \tag{45}$$

where $\{\varepsilon_t(w)\}$ are uncorrelated random variables with mean 0 and variance $\sigma\varepsilon^2$ and c_t is an unknown parameter, which can be estimated using the least-squares criterion by choosing c_t such that minimizes $\sum_{w=1}^{n}(x_t(w) - c_t)^2$ and, hence, $\hat{c}_t = \frac{1}{n}\sum_{w=1}^{n} x_t(w)$.

Once I computed the estimated matrix $\hat{X}(w)$, I could obtain the predicted mobile life pattern by applying a kernel estimation function. I used the first two months of data as the initial data for training the model I then made a prediction for each week of a following month for each Center User. My prediction's accuracy was measured using a similarity measure based on Hellinger distance to determine similarity between the predicted mobile life pattern and the actual pattern. similarity My approach accurately predicted mobile life patterns with an average of 71.05% overall, I provide my results for each user in Table 5.1.

5.8. Conclusion

Based on my mobile social grouping framework (142), I reveal the significant role of social tie on similarity in calling patterns and inter-connect time. My results show that (i) the closer the social tie, the higher the similarity, (ii) a closer tie implies higher reciprocity, and (iii) the inter-contact time increases as social closeness becomes distant. I also show that social tie and mobile life pattern can be predicted accurately. With a discrete Markov model and a Moving Average model, social tie and mobile life pattern can be predicted with 96% and 71% accuracy rate, respectively.

TABLE 5.1. The overall result of accuracy rate of social group and mobile life pattern prediction.

User	Accuracy Rate (%)	
	Social Group Prediction	Mobile Life Pattern
1	96.00	77.44
2	100	82.23
3	100	70.77
4	96.55	62.74
5	88.24	65.82
6	100	73.41
7	84.62	65.18
8	96.22	85.86
9	87.88	63.76
10	95.74	66.43
11	96.30	77.91
12	92.31	51.89
13	98.00	66.93
14	100	80.25
15	94.44	70.56
16	96.00	67.06
17	97.78	68.90
18	98.00	88.37
19	97.50	81.82
20	91.67	76.38
21	94.74	61.13
22	90.00	64.96
23	100	61.90
24	100	70.12
25	100	60.97
26	97.72	89.05
27	96.55	72.67
28	100	73.10
29	100	57.88
30	96.30	75.89
Mean	96.09	71.05
Std. Dev.	4.11	9.15

CHAPTER 6

CALL PREDICTOR: PHONE CALL-BASED DAILY PLANNER

6.1. Introduction

Prediction plays an important role in various applications. The prediction is widely applied in the areas such as weather, environmental, economic, stock, disaster (earthquake, flooding), network traffic, and call center forecasting (147)(148)(149)(150). Companies use predictions of demands for making investment and efficient resource allocation. The call centers predict workload so that they can get the right number of staff in place to handle it. Network traffic prediction is used to assess future network capacity requirement and to plan network development so as to better use of network resources and to provide better quality of services. Prediction is also applied in the human behavior study by combining the computer technology and social networks (86)(81)(151)(152). There is also some work reported on telephone telepathy based on psychology (153).

Predicting the expected calls for a busy business executive can be very useful for scheduling a day. Match making services can use calling patterns for the compatibility studies (154)(155). Moreover, the prediction of incoming calls can be used to avoid unwanted calls and schedule a time for wanted calls. For example, the problem of spam in VoIP networks has to be solved in real time compared to email systems. Compare receiving an email spam at 2:00 AM that sits in the inbox until you open it the next morning to receiving a junk phone call that must be answered immediately.

Over the past few years, there has been a rapid development and deployment of new advanced phone features, including internet access, e-mail access, scheduling software, built-in camera, contact management, accelerometers, and navigation software as well as the ability to read documents in variety of format such as PDF and Microsoft Office. In 2005, Google filed a patent including detail about the Google Phone (GPhone) that could predict what a

user is searching for or the words they are typing in a text messages by taking into account the user's location, previous searching/messaging history, and time of the day. However, none of these features offers ability to predict future calls.

Let us consider a simple caller-callee scenario shown in Fig. **??**. In order to have an efficient scheduling of transactions, the caller wants to know the willingness of taking calls of the callee, which could be determined by the callee's presence. At the same time, the callee wants to know (predict) the incoming calling pattern (calling schedule) of the caller. This raises two interesting problems; (i) predicting the incoming calling pattern of the caller, and (ii) determining the presence of the callee. In this chapter, I attempt to solve the first problem. The second problem and the proposed solution have been addressed in my other work (156).

FIGURE 6.1. A simple caller-callee scenario.

To the best of my knowledge, no scientific research has been reported in predicting the incoming calls for phone services. Predicting of incoming calls using just the call history is a challenging task. I believe that this is a new area of research. One way of predicting incoming calls from specified callers is to compute the probability of receiving calls associated with them. In this chapter, I present a model for predicting the next-day calls based on caller and user's past history.

6.2. Call Predictor

The Call Predictor (CP) for computing the probability of receiving calls from a specified caller and making next-day call prediction can be deployed either in conjunction with perimeter controllers such as voice spam filters or firewalls, or in end systems such as multimedia phones. The basic architecture of the CP is shown Fig. 6.2.

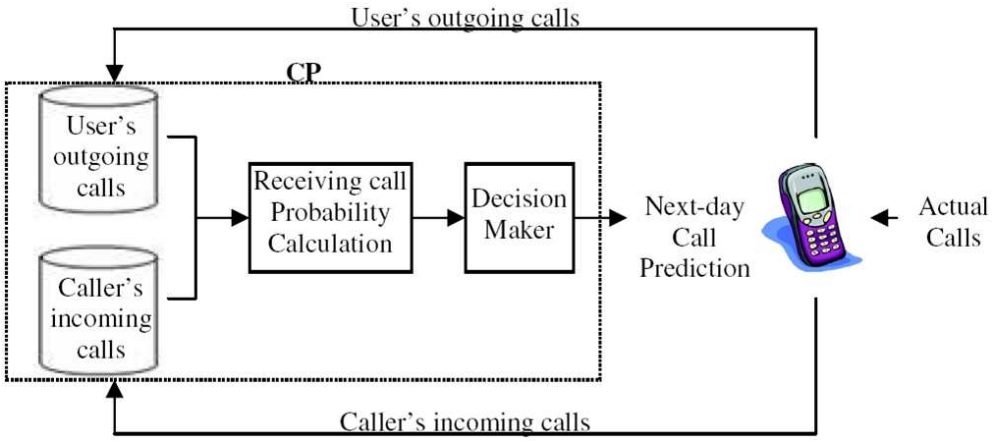


FIGURE 6.2. Architecture of Call Predictor (CP). The CP calculates the probability of receiving next-day calls from specified callers based on the past call history (incoming and outgoing calls) and makes next-day call prediction. The call database is updated with the actual call activities.

For any time that phone user requests for a call prediction of a particular caller, the CP computes the probability of receiving calls of the next 24 hours based on the caller's past history (*Caller's incoming calls*) and the previous outgoing calls from the phone user to the caller (*User's outgoing calls*). Both of these histories are maintained by the CP by logging the call specific information for every call received and mode by the user. The computed receiving call probability is checked with a preconfigured threshold value to make a decision as to predict "call" or "no call" for each of the next 24 hours.

6.3. Call Prediction Framework

To predict the future incoming calls, the behavior learning models must be used. These models should incorporate mechanisms for capturing the caller's behavior (based on call arrival time and inter-arrival time), the user's behavior (based on call departure time), reciprocity (based on call inter-arrival/departure time), the probability model of receiving calls from caller, and finally, the next-day call prediction.

6.3.1. Dataset

Every day calls on the cellular network include calls from different sections of our social life. We receive calls from family members, friends, supervisors, neighbors, and strangers. Every person exhibits a unique calling pattern. These calling patterns can be analyzed for predicting the future calls to the callee.

To study calling pattern, I collected the actual call logs of 20 individuals at my university. These 20 individuals are faculties, staffs, and students. I am in process of collecting many more call logs. The details of the data collecting process are given in (154). I found it difficult to collect the data set because many people are unwilling to give their call logs due to privacy issues. Nevertheless, the collected datasets include people with different types of calling patterns and call distributions.

As part of the data collecting process, each individual downloaded three months of detail telephone call records from his/her online accounts on the cellular service provider's web-site. Each call record in the dataset had the 5-tuple information: Date, Start time, Type (Incoming or Outgoing), Caller ID, and Talk time (call duration in minutes).

I then used the collected data for deriving the traffic profiles for each caller who called the individuals. To derive the profile, I inferred the arrival time (time of receiving a call), inter-arrival time (elapsed time between adjacent incoming calls), and inter-arrival/departure time (elapsed time between adjacent incoming and going calls).

6.3.2. Probability Computation

In our daily life, when we receive a phone call, at the moment of the first phone ring before we look at the caller ID, we often guess who the caller might be. We base this estimation on:

- *Caller's behavior*: Each caller tends to have a unique calling pattern. These patterns can be observed through history of calling time (we normally expect a call from a caller who has history of making several calls at some particular time, for example, your spouse likes to call you while you drive to work in the morning and after work in the evening therefore when your phone rings while you are on the way to work or back home, you likely to guess that it is a phone call from your spouse), periodicity of call history (we can expect that a caller who calls periodically will repeat the same pattern, for example, your friend calls you at about 2:00 PM every Tuesday therefore you expect a call from him/her at about 2:00 PM for next Tuesday).
- *Reciprocity*: The communication activity patterns between the caller and the user in the past. These patterns can be observed in terms of number of user's outgoing calls per caller's incoming call and call inter-arrival/departure time.

The calling pattern based on caller's call arrival time can be captured by using nonparametric density estimation. The most popular method for density estimation is the kernel density estimation (also known as the Parzen window estimator (94)) which is given by Eq. 46.

$$a(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right). \tag{46}$$

$K(u)$ is kernel function and h is the bandwidth or smoothing parameter. The most widely used kernel is the Gaussian of zero mean and unit variance which is defined by Eq. 47.

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{47}$$

The choice of the function and h is crucial. Several optimal bandwidth selection techniques have been proposed (157)(158). In this paper, I use AMISE optimal bandwidth selection using the Sheather Jones Solve-the-equation plug-in method (96). Fig. 6.3(a) shows an example histogram of call arrival time. It should be noted that the widow of observation is shifted to start at 5:00 AM and end at 4:59 AM in order to capture the entire calling pattern in the middle. The corresponding estimated probability density function (pdf) using kernel density estimation is shown in Fig. 6.3(b).

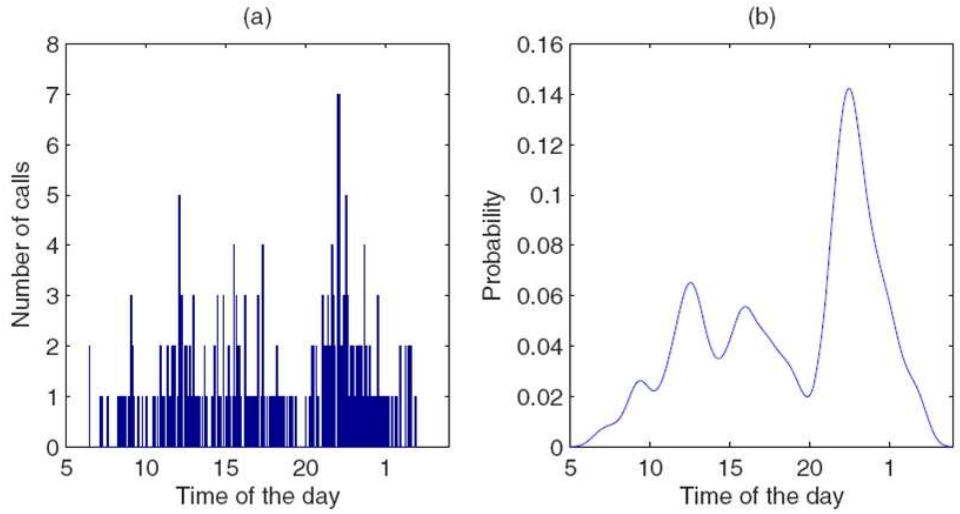


FIGURE 6.3. (a) An example histogram of call arrival time. (b) The estimated probability density function using kernel density estimation of the example histogram of call arrival time shown in Fig. 3(a). Note that observation window is 5:00 AM to 4:59 AM.

I define a *Call Matrix* as a matrix whose entries are call indicators where rows are hours of the day and columns are days of observation. The call indicator (CI) indicates if there is at least one incoming call or going call or both incoming and outgoing call or no call. CI's values and its indications are given in Eq. **??** and an example Call Matrix of 15 days of observation is shown in Fig. 6.4.

(48)
$$CI = \begin{cases} 0, \textit{ no call} \\ 1, \textit{ at least one incoming call} \\ 2, \textit{ at least one outgoing call} \\ 3, \textit{ at least one incoming and one outgoing call} \end{cases}$$

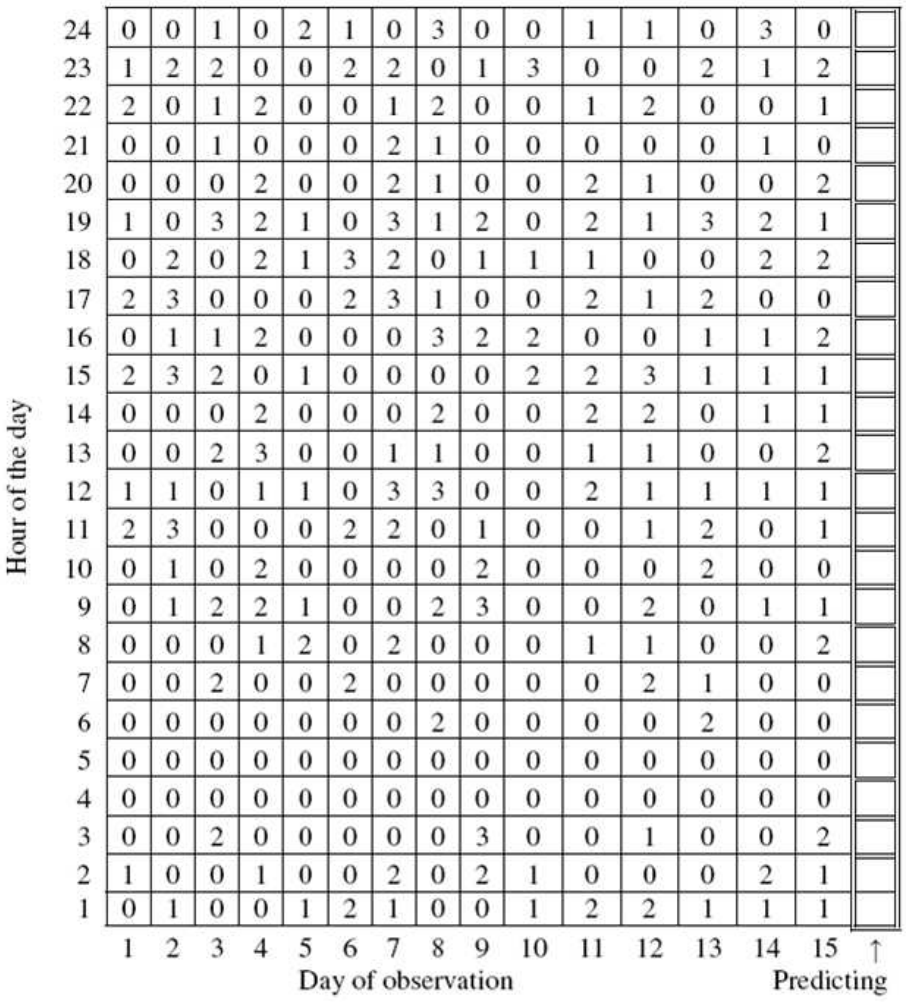


FIGURE 6.4. An example Call Matrix of 15 days of observation.

The caller's behavior can also be observed through the call inter-arrival time. However, the inter-arrival time in our normal sense is the elapsed time between temporally adjacent calls, which I believe that it does not accurately represent the caller's behavior based on

inter-arrival time. Due to the human nature that requires state of natural rest, sleeping time causes the inaccuracy in the average inter-arrival time. In fact, it increases the average inter-arrival time from the true value. Therefore, I believe that the more accurate angle to observe calling pattern based on inter-arrival time is to scan over each hour of the day through days of observation, *i.e.* capturing inter-arrival time patterns by observing each row of the Call Matrix.

Let a random variable X_k be inter-arrival time of k^{th} hour where $k = 1, 2, 3, \ldots, 24$. A Normal distribution $N(\mu_k, \sigma_k^2)$ is assumed for the call inter-arrival time since no information is available that $Pr(X_k = \mu_k - c) < Pr(X_k = \mu_k + c)$ or vice versa therefore it can be safely assumed that $Pr(X_k = \mu_k - c) = Pr(Xk = \mu_k + c)$ where μ_k is the mean and σ_k^2 is the variance of inter-arrival time of k^{th} hour, which can be calculated by Eq. 49 and Eq. 50 respectively.

$$\mu_k = \frac{1}{N-1} \sum_{n=1}^{N-1} x_k(n). \tag{49}$$

$$\sigma_k^2 = \frac{1}{N-1} \sum_{n=1}^{N-1} \left(x_k(n) - \mu_k \right)^2. \tag{50}$$

N is the total number of calls and $x_k(n)$ is the n^{th} inter-arrival time. The inter-arrival time is now treated as a random variable X_k that consists of number of small random variables $\{x_k(1), x_k(2), x_k(3), ..., x_k(N-1)\}$, is normal random variable, which has probability density function (pdf) given by Eq. 51.

$$i_k(x_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(x_k - \mu_k)^2 / 2\sigma_k^2}. \tag{51}$$

For example, if a caller calls on average every 3 days, the chances of receiving a call one day earlier (day 2) or day one later (day 4) are the same.

As previously mentioned that receiving a call is influenced by not just caller's behavior but also reciprocity, one way to observe the calling patterns based on reciprocity is to monitor

the number of outgoing calls per incoming call. This can give us a good approximation of when the next incoming call can be expected. A normal distribution $N(\mu_k, \sigma_k^2)$ is also assumed for the same reason as in the inter-arrival time case, where Y_k is a random variable representing the number of outgoing calls per incoming call of the k^{th} hour where μ_k is the mean and σ_k^2 is the variance of which can be calculated by Eq. 52 and Eq. 53 respectively.

$$\mu_k = \frac{1}{M} \sum_{n=1}^{M-1} y_k(n). \tag{52}$$

$$\sigma_k^2 = \frac{1}{M} \sum_{n=1}^{M-1} \left(y_k(n) - \mu_k \right)^2. \tag{53}$$

M is the total number of incoming calls of k^{th} hour and $y_k(n)$ is the number of outgoing calls beween the n^{th} and $(n+1)^{th}$ incoming call. Therefore, the pdf is given by Eq. 54.

$$n_k(y_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(y_k-\mu_k)^2/2\sigma_k^2}. \tag{54}$$

An example of calculating $n_k(y_k)$ is shown in Fig. 6.5.

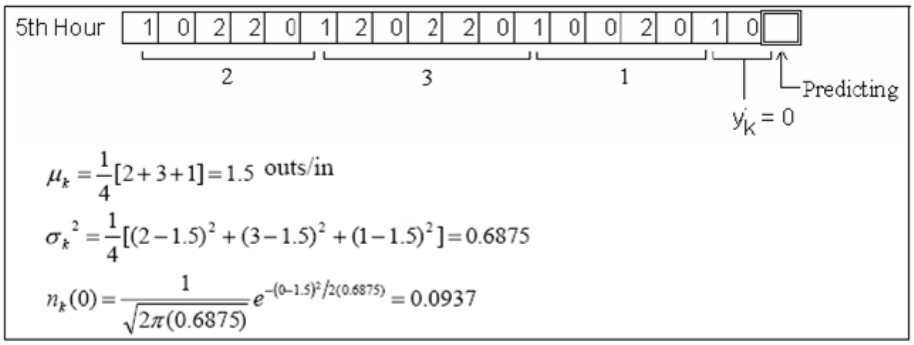


FIGURE 6.5. An example of calculating $n_k(y_k)$ for one hour slot (5th hour) of 18 days of observation.

Another angle to observe the calling patterns based on reciprocity is to monitor the inter-arrival/departure time. This gives us the chance (probability) of receiving a call from the caller given the time of the last outgoing call to the caller.

Let Z_k be a random variable mapping to the inter-arrival/departure time of the k^{th} hour. A normal distribution $N(\mu_k, \sigma_k^2)$ is also assumed for the same reason previously mentioned. The mean (μ_k) and variance (σ_k^2) are given by Eq. 55 and Eq. 56 respectively.

$$\mu_k = \frac{1}{L-1}\sum_{n=1}^{L-1} z_k(n). \tag{55}$$

$$\sigma_k^2 = \frac{1}{L-1}\sum_{n=1}^{L-1} \left(z_k(n) - \mu_k\right)^2. \tag{56}$$

L is total number of incoming calls of k^{th} hour and $z_k(n)$ is the average inter-arrival/departure time of the nth incoming call to all right-hand-side outgoing calls (in the Call Matrix's row) before reaching the $(n+1)^{th}$ incoming call (an example is illustrated in Fig. 6.6). The pdf of inter-arrival/departure time is given in Eq. 57.

$$t_k(z_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(z_k-\mu_k)^2/2\sigma_k^2}. \tag{57}$$

An example of calculating $t_k(z_k)$ is shown in Fig. 6.6.

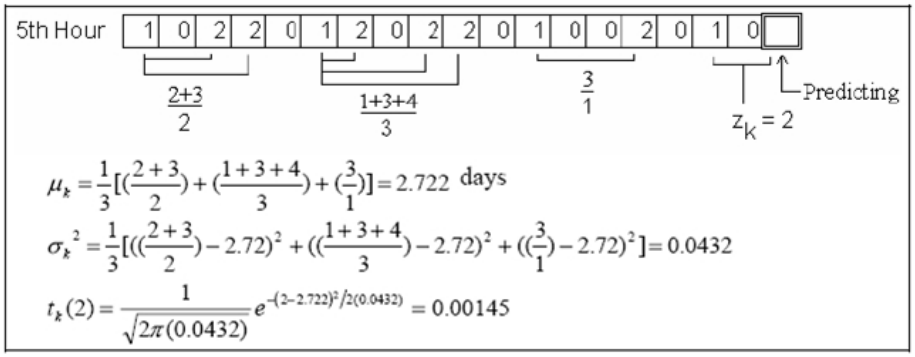


FIGURE 6.6. An example of calculating $t_k(z_k)$ for one hour slot (5th hour) of 18 days of observation.

From Eq.46, 51, 54, and 57, I can infer the probability of receiving a call from "*Caller A*" of k^{th} hour ($P_A(k)$) as the average of the probability of receiving a call based on the

caller's behavior (arrival time and inter-arrival time) and the reciprocity (number of outgoing calls per incoming call and inter-arrival/departure time), which is given by Eq. 58 where $k = 1, 2, 3, \ldots, 24$.

$$P_A(k) = \frac{1}{4}\left[a_k(k) + i_k(x_k) + n_k(y_k) + t_k(z_k)\right]. \tag{58}$$

There is another group of callers who never receive any calls back from the user, i.e. no reciprocity. More likely these callers are telemarketers or voice spammers. Since there is no history of call interaction between the callers and the user, the Eq. 58 reduces to the averaging over the probability based on only the caller's behavior, which is given by Eq. 59. Likewise, for the regular callers where some hour slots (rows of Call Matrix) have no reciprocity, Eq. 58 also reduces to Eq. 59.

$$P_A(k) = \frac{1}{2}\left[a_k(k) + i_k(x_k)\right]. \tag{59}$$

To present the accuracy of the receiving call probability model, a phone user is randomly selected to represent all the individuals. Fig. 6.7 shows 30 consecutive days of receiving call probability calculation for an arbitrary caller where the receiving call probability is represented with a green surface and the actual calls during these 30 days of observation are represented with vertical black pulses.

It can be observed from Fig. 6.7 that most of the calls are received when the computed receiving-call probability is high. At the same time, no call is received during 0:00 AM to 9:00 AM period where the probability of receiving call is low.

6.4. Performance Analysis

The CP is tested with the actual call logs. Its performance is then measured by false positives, false negative, and error rate. A false positive is considered when a call is predicted but no call is received during that hour. A false negative is considered when no call is

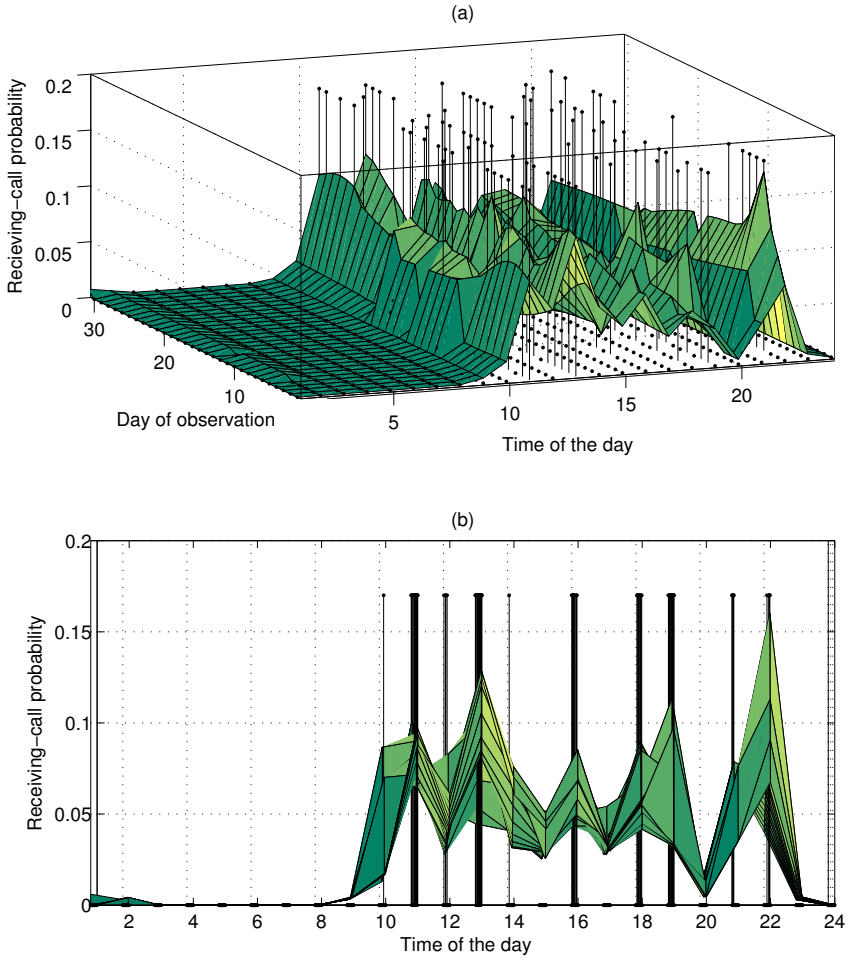


FIGURE 6.7. A randomly selected phone user with 30 consecutive days of computed receiving-call probability of an arbitrary caller plotted with the actual received calls represented with vertical pulses. Top figure is the 3-dimensional view. Bottom figure is the front view (looking from the first day of observation).

predicted but a call is received during that hour. Error rate is defined as a ratio of the number of fault predictions to the total number of predictions.

An experiment is conducted with 20 phone users (as mentioned in Section 3.1). The call logs of the first 2 months are used to train the CP, which is then tested with the call logs of the following month (next 30 days). Each of the 30 days of testing, the new prediction is

consequently made by the CP at midnight (0 AM) with all available call history (up to that day) taken into account. The computed receiving call probability is checked with a threshold value to make a decision as to predict "call" or "no call" for each of the next 24 hours. The average number of calls per day is computed and rounded to the next largest integer M. The threshold is dynamically set as M hour slots are selected to make "Call" prediction and the rest of the $(24 - M)$ time slots are predicted "No Call." The experimental results are shown in Table 6.1.

There are total of 151,008 predictions made with 8,095 total fault predictions. The average false positive is 2.4416%, the average false negative is 2.9191%, and the average error rate is 5.3606%. Therefore the overall average number of fault predictions per day (24 predictions) is 1.2866 and the average tolerance is 2.1246 hours. The average tolerance is a measure of how far off (in hours) the predicted call from the actual call when fault prediction occurs.

6.5. Conclusion

In this chapter, I propose a Call Predictor that computes receiving call probability and makes the next-24-hour call prediction. The receiving call probability is based the caller's behavior and reciprocity. The caller's behavior is measured by the caller's call arrival time and inter-arrival time. The reciprocity is measured by the number of outgoing calls per incoming call and the inter-arrival/departure time.

The kernel density estimation is used to estimate the probability model for the calling pattern based on caller's arrival time. The normal distributions are assumed for the inter-arrival time, number of outgoing calls per incoming call, and the inter-arrival/departure time. The final receiving call probability model is the average of the receiving call probabilities based on these four parameters.

To validate the model, the cell phone call records of real-life individuals at my university are used to test the call predictor. The results show that the call predictor exhibits a reasonably good performance with low false positives, false negatives, and error rate.

TABLE 6.1. The experimental results of 20 phone users.

Phone user	Number of predictions	Number of fault predictions	False positive (%)	False negative (%)	Error rate (%)	Number of fault predictions per day	Average tolerance (hours)
1	6,432	332	2.5683	2.9214	5.4896	1.3175	1.9070
2	14,472	503	1.4486	2.2042	3.6528	0.8767	1.4618
3	1,968	133	4.0278	2.7183	6.746	1.6190	2.5676
4	13,512	609	1.9651	2.5916	4.5568	1.0936	1.8395
5	8,136	547	3.9371	4.2595	8.1966	1.9672	2.8694
6	5,616	579	6.4966	5.6342	12.1308	2.9114	2.7092
7	6,000	211	2.4096	1.6697	4.0793	0.9790	1.9995
8	10,178	178	1.1033	0.8860	1.9893	0.4774	1.3011
9	1,776	224	6.4342	8.1774	14.6117	3.5068	2.1220
10	8,352	659	3.4785	4.7221	8.2005	1.9681	2.8823
11	17,400	870	2.6798	2.5612	5.2409	1.2578	1.8337
12	2,088	67	2.6235	0.6944	3.3179	0.7963	1.5622
13	7,416	374	2.6365	2.9602	5.5968	1.3432	2.0133
14	3,720	167	2.6730	3.1831	5.8561	1.4054	2.3980
15	7,632	357	2.2900	3.0674	5.3574	1.2857	2.4452
16	19,416	1,090	2.7982	3.3569	6.1551	1.4772	2.7210
17	6,840	652	3.8854	5.6129	9.4984	2.2796	1.6221
18	2,808	216	3.5417	4.2014	7.7431	1.8583	1.8703
19	2,208	181	3.5779	4.6196	8.1975	1.9674	2.7001
20	5,040	146	1.0417	1.8750	2.9167	0.7000	1.6675

Clearly, there are still many parameters that need to be identified to capture the calling patterns. This work is intended to be the first piece of many more to come in this new area

of predicting future calls which can be useful to many applications such as planning a daily schedule and preventing unwanted communications (e.g. voice spam). Also, the prediction technique proposed here is preliminary and other approaches need to be considered in order to minimize the number of false positives and negatives. I will continue to investigate other parameters to characterize the behaviors of the phone users and explore other prediction techniques to improve the performance of the call predictor as my future direction.

CHAPTER 7

CALL PREDICTED LIST: LIST OF POTENTIAL CALLERS AND CALLEES

7.1. Introduction

With the rapid development of telecommunication technologies and the fast-growing number of users on the networks, the cellular phone has moved beyond being a simple phone and has become a mobile workstation and integrated into many parts of people's lives. The mobile phone is gradually becoming a ubiquitous computing device at this early stage of the pervasive-computing era where handheld devices are precursors to a phase of ambient computing that is always on, personalized, context-sensitive, and highly interactive.

Mobile (personal) phones record the history of our lives in the form of the call logs. By utilizing call logs in computing human (user) behavior, I can enhance the usability of the phone as it is becoming more than just a voice communication device and evolving into an intelligent assistant to its user.

In this chapter, I design and evaluate a model that makes use of the call logs to predict incoming as well as outgoing calls. With my model, the personal phone will become even more personal as it learns and recognizes its user's calling behavior as well as the associated users' (callers' and callees') in order to provide the most accurate prediction of the future caller and callee for the user. In this way, the mobile phone becomes more personalized and sensitive to the user's context and needs.

7.2. Call Prediction

Predicting incoming calls can be very useful for planning and scheduling (e.g., it can be used to avoid unwanted calls and schedule time for wanted calls). People normally check weather forecast before leaving homes and watch for signs of approaching storms to prepare and schedule their days accordingly. Knowing what is coming next gives us supplemental time to think, prepare, and optimize our solutions. I believe that incoming call prediction

can be useful for daily planning and it may become an important element as an initiative decision support for our daily life scheduling.

Quite often in our daily lives, we find ourselves in a situation where we wish to know who will be calling in the next hour so we could schedule (plan) things out accordingly. In many occasions, we know for certain that we will not be available to accept any incoming calls over the next hour (e.g., having a flight, attending a class, having a meeting) thus we wish to know who will be calling during the next hour so we could perhaps make a call to the persons to inform of our next-hour schedule as we do not wish to miss any important future calls, which could be too important calls to miss.

Likewise, predicting outgoing calls can be useful for many applications such as enhancing mobile phone's usability by providing a list of the most likely contacts/numbers to be dialed when user wants to make a call. Such that it reduces the searching time as well as enable better life synchronization for the user.

Our call predictor makes use of the user's call history e.g., call identifications, time of calls, day of calls, frequency of calls, and last received/made numbers, to build a probabilistic model of calling behavior. The calling behavior model is then used to generate a list of numbers/contacts that are the most likely to be the callers for the next hour (as an incoming call predictor) or a list of numbers/contacts to be dialed (as an outgoing call predictor).

The list can be presented to the user in a number of different ways for different purposes. I envisage the predictor as a "Call Predicted List (CPL)," *i.e.,* a list that anticipates the most likely callers/callees and gives these numbers/contacts higher precedence on the list. Figure 7.1 shows an example of the envisaged CPL where the most likely callers/callees are listed higher on the list.

7.3. Call Prediction Framework

When that cell phone rings, how often do we make a guess on who the caller might be? More often than not, but if we do make a guess, we are usually right. We often base this estimation on the caller's call history as well as our call history with the caller.

FIGURE 7.1. CPL user interface.

Each caller exhibits a unique calling pattern which can be observed through history of "time of the calls" *i.e.,* we normally expect a call from someone who has a history of making several calls during a particular time period of the day. For example, your spouse likes to call you while you are driving to work in the morning therefore when your phone rings while you are on the way to work you are likely to guess that it is a phone call from your spouse. The pattern can also be observed from "day of the calls," for example, your friend, John, has made several calls to you on every Tuesday because it is his day off, therefore when your phone rings on Tuesday, the first person that comes to mind is John. Likewise, the person who has made the most "number of calls" to you (regardless of time and day) among other callers is also the person whom you most anticipate the calls from. Receiving a call is also influenced by the "reciprocity" or call interaction between the user and the

caller. For example, you may anticipate a phone call from a specific person based on your last phone conversation with the person (e.g., "call me when you get home" or "call me same time tomorrow" or "I'm busy right now, call me back in an hour"). This reciprocity may sequentially lead to a later call received from the person caused by your initiative. For example, you decide to make a call to an old friend to whom you have not called for a long time, and later you start to receive calls from this old friend. Another example, you make a call to your mother to get some advice during the night (assume that normally you do not make or receive calls from her during this time), and then you receive calls from your mother later on during that night. These are the examples that actually happen in our everyday lives as a phone user. Understanding the actual human behavior towards phone usage gives the CPL an intelligence to assist its user effectively and in the same time makes the smart phone smarter.

7.3.1. Datasets

Predicting future calls is a challenging task. It requires a design of model that should incorporate mechanism for capturing and learning the caller/callee's calling patterns. Calling patterns can be extracted from the call logs, which can be obtained from a variety of sources. For example, they may be collected by a network or service operator for billing purposes or they may be captured directly on device such as a mobile phone or on a software application such as a VoIP softphone. In my current implementation, I use two sets of real-life call logs of 30 combined users with nearly 3,000 callers/calees and over 46,000 call activities. my first dataset consists of three-month call logs of 20 individual mobile phone users, which were collected at University of North Texas (UNT), Denton, during summer of 2006. These 20 individuals were faculty, staff, and students. These call logs were collected as part of the Nuisance Project, where Kolan et al. (105) studied the nuisance level associated with each phone call. The details of the data collecting process are given in (106). my second dataset consists of three-month call logs of ten mobile phone users, which were collected during summer of 2008 at UNT. These ten subjects were also faculty, staff, and students.

As part of the data collecting process (for both datasets), each user downloaded three months of detail telephone call records from his/her online accounts on the mobile phone service provider's website. Each call record in the dataset had 5-tuple information as follows where an example call record is shown in Fig. 7.2.

- Date: date of the call
- Start time: start time of the call
- Type: type of the call *i.e.,* "Incoming" or "Outgoing"
- Call ID: caller/callee identification
- Talk Time: duration of the call (in minutes)

Date	Start time	Type	Call ID	Talk time
3/11/2007	2:28PM	Outgoing	123-4567890	2
3/11/2007	5:31PM	Incoming	888-8888888	11
3/11/2007	8:12PM	Incoming	999-9999999	6
...	...	...	...	...

FIGURE 7.2. An example of a call record. Note that Call ID's have been modified for privacy reason.

7.3.2. System Overview

The call record shown in Fig. 2 is subject to pre-processing to extract features or information about "time of the calls" (day and hour), "total call count," and "reciprocity". The pre-processed call records are eventually fed into the classifier to be ingested. Classifier then outputs a list of phone numbers ordered by the likelihood of the number being the next-hour caller (as an incoming call predictor) or the dialing number (as an outgoing call predictor). The basic system overview is shown in Fig. 7.3.

7.3.3. Inference Engine

With the same framework, the CPL can function as an incoming call predictor and an outgoing call predictor with just a simple modification in the direction of the calls (*i.e.,*

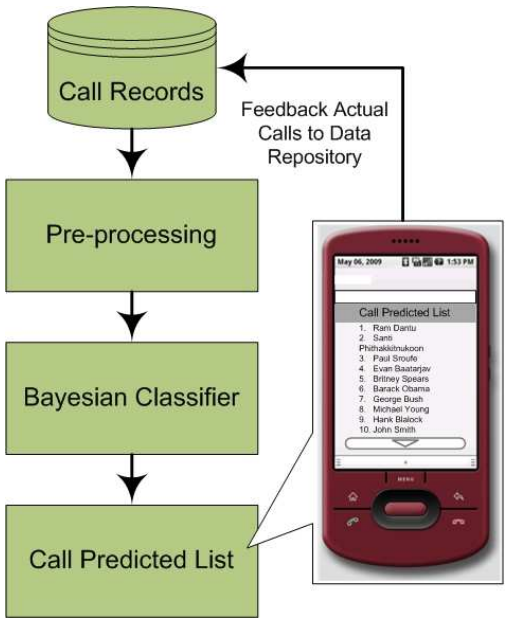


FIGURE 7.3. Basic system overview.

incoming and outgoing) in the analysis. Therefore, let us consider the CPL first as an incoming call predictor.

Our inference engine is driven by a Bayesian classifier, which has two modes of operation; training and predicting. During the training, classifier ingests the pre-processed call logs and constructs four hash tables that primarily contain call counts of the corresponding features. The first table maps each unique telephone number (or caller identifier) to a count of calls received for each day of the week as shown in Fig. 7.4.

Caller ID	Day of week						
	1	2	3	4	5	6	7
123-4567890	5	23	6	2	11	0	1
888-8888888	6	0	0	1	0	33	4
999-9999999	0	0	11	8	21	7	8
...	...	...	...	...	...	...	...

FIGURE 7.4. An example of a hash table for day of the week.

The second table maps each unique telephone number (or caller identifier) to a count of calls received for each hour of the week as shown in Fig. 7.5.

Caller ID	Hour of day						
	0	1	2	...	21	22	23
123-4567890	0	0	0	...	9	3	1
888-8888888	2	0	0	...	15	8	2
999-9999999	0	0	0	...	27	9	0
...	...	...	...	...	...	...	...

FIGURE 7.5. An example of a hash table for hour of the day.

The third table maps each unique telephone number (or caller identifier) to the total number of calls received as shown in Fig. 7.6.

Caller ID	Call count
123-4567890	118
888-8888888	121
999-9999999	157
...	...

FIGURE 7.6. An example of a hash table for cumulative frequency of calls.

Quantify the "reciprocity" is not quite trivial. Having no knowledge about the context of the previous phone calls of the user, it is difficult to identify which outgoing calls would influence the future incoming calls. Nevertheless, the recent received calls can be linked to the user's calling behavior. These recent received calls are typically stored in the "last dialed calls" list (normally a list of last 20 outgoing calls) where the lower order corresponds to more recent dialed number (e.g., "1" is the most recent dialed number, "20" is the least recent dialed number). Thus the same number/contact can occupy in more than one position on the list. Clearly the numbers/contacts on the list are pushed down one position when a new call is received. Based on the position on this list and its corresponding number of times that

actual incoming caller was listed on that position, the likelihood of receiving a call can be estimated. For example, suppose currently statistic (hash table) shows that position "3" of the list has the most counts, it implies that the number/contact that is on position "3" of the current "last dialed calls" list has the highest likelihood of being the next caller. Therefore, the fourth hash table maps each position on the "last dialed calls" list to the count of the calls received as shown in Fig. 7.7.

Once the input call records have been ingested and the hash tables generated, the classifier is considered trained. With the classifier trained on a set of representative call records, it is then ready to be used in predicting mode. The classifier is given a target day of week, hour of day, total call count, and current last-20-dialed-calls list, and uses the calling behavior model to estimate the likelihood of the user receiving each of the telephone numbers (or caller identifiers) seen in the training data. Clearly, the classifier can only make predictions for numbers that it has already seen.

Position on last-20-dialed-calls	Number of times when a call is received and its caller is listed on corresponding position on last-20-dialed-calls list
1	69
2	45
3	71
...	...
...	...
19	3
20	8

FIGURE 7.7. An example of a hash table for caller's position on the last-20-dialed-calls list.

A likelihood metric then is calculated for each number seen by the classifier and the numbers are then sorted in descending order of likelihood of being received. If the caller's behavior has a high degree of predictability (*i.e.,* they tend to make calls consistently to user at this certain time of the day, or in this particular day of the week, or after some number of calls from the user), then it is expected that the number is likely to be listed towards the top of the list. If there is a tie *i.e.,* several numbers end up with the same value of likelihood, then the classifier list them in the alphanumerical order.

Our inference engine is based on the Naïve Bayesian Classifier, which is a simple probabilistic classifier based on Bayes' theorem with independence assumptions. In my case, I want to compute the likelihood of each number (T_n) being received given that the day of the week (D_x), hour of the day (H_y), the current last-20-dialed-calls list (L_z), and total call count (F_n). Bayes rule (92) of conditional probability is given by Eq. 60.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{60}$$

where $P(A|B)$ is the *posterior* probability, which is the probability of the state of nature being A given that feature value B has been measured. The *likelihood* of A with respect to B is $P(B|A)$, which indicates that other things being equal, the category A for which $P(A|B)$ is large is more "likely" to be the true category. $P(A)$ is called *prior* probability. The *evidence* factor, *P(B)*, can be viewed as a scale factor to guarantee that the posterior probabilities sum to one.

I use this rule to obtain the probability of a number being received given a specific hour of the day, day of the week, current last-20-dialed-calls list, and total call count, as given by Eq. 61.

$$P(T_n|D_x, H_y, L_z, F_n) = \frac{P(D_x|T_n)P(H_y|T_n)P(L_z|T_n)P(F_n|T_n)P(T_n)}{P(D_x, H_y, L_z, F_n)}, \tag{61}$$

With the Naïve Bayesian classifier, a well known issue occurs when a particular attribute value doesn't occur in conjunction with every class value in the training data. In my case,

the attributes are D_x, H_y, and L_z. The class values are the incoming telephone numbers (callers). The computed probability of a number being received at a particular time will be zero if the training data has no instance of that number being received during either the specified hour or the specified day.

A solution to this problem is to start all the call counts in the Hash tables for day-of-week and hour-of-day at one instead of zero and defining some normalizing factors in the resulting computations. This is not an issue for the F_n since there must be at least one call count for any seen incoming call. For L_z, this is sort of an issue since only those numbers/contacts that are on the current last-20-dialed-calls list are considered. A solution for this case is to assign the lowest call count of the position on the last-20-dialed-calls list (hash table) to those phone numbers that are not on the current last-20-dialed-calls list. Therefore, those numbers that are not on the current last-20-dialed-calls list will have the same probability of being received as the lowest probability of the number on the current list being received. There is also a possibility of one telephone number occupies more than one position on the current last-20-dialed-calls list. In this situation, the highest call count among all positions occupied by that telephone number is assigned to it.

Adopting this approach, I compute the likelihood of the caller T_n being received, given D_x, H_y, L_z, and F_n, by Eq. 62.

$$(62)\quad L(T_n|D_x, H_y, L_z, F_n) = \left(\frac{C(T_nD_x)+1}{C(T_n)+7}\right)\cdot\left(\frac{C(T_nH_y)+1}{C(T_n)+24}\right)\cdot\left(\frac{C(T_nL_z)}{C(L)}\right)\cdot\left(\frac{C(T_nF_n)}{C(T_n)}\right),$$

where $C(T_nD_x)$ is the call count from the caller T_n on day D_x ($x = 1, 2, 3, \ldots, 7$), $C(T_nH_y)$ is the call count from the caller T_n during hour H_y ($y = 0, 1, 2, \ldots, 23$), $C(T_nL_z)$ is the call count from the caller T_n when T_n's position on the current last-20-dialed-calls list is L_z ($z = 1, 2, 3, \ldots, 20$), $C(T_nF_n)$ is the total call count from caller T_n ($n = 1, 2, 3, \ldots, N$, where N is the total number of callers that have made at least one call to the user), $C(L)$ is the total call count of all position on the list (sum of the second column of hash table in Fig. 7.7), and $C(T_n)$ is the total call count from caller T_n over the entire training data.

7.4. Performance Analysis

In this section, the CPL is evaluated with the actual call logs of 30 mobile phone users as described in Section 3. The first two months (approximately 60 days) of call logs are used to train the CPL and the rest of the call logs are assumed to be the future observed call activities to test the performance of the CPL by observing for each call received what position that actual caller has in the predicted list. If the CPL performed perfectly, one clearly would expect the actual caller to be at the top of the predicted list. Generally, such performance is not achievable, but one might expect that the actual caller would tend to appear earlier rather than later in the list.

7.4.1. Improvement over Conventional Last-Received-Calls List

The overall performance of the CPL based on these 30 mobile phone users is shown in Fig. 8 where the its accuracy is measured by the average percentage of the actual callers listed within the predicted list as the length of the list varies from 1 to 20. One may be curious to find out that if the conventional last-20-received-calls list, which already exists in today's mobile phone, is used as a call predicted list. How well can it perform? Will it performs better than my CPL? *The comparison is illustrated in Fig. 7.8 where it can be seen clearly that my CPL outperforms the last-20-received-calls list (if used as predictor) with nearly 30% better accuracy.*

7.4.2. Impact of Caller Population

The CPL would always predict the caller correctly, if there was only one caller. In general, the population of the callers increases e.g., meeting new friends, signing up with a new group, being on telemarketers' list, etc. This increasing number of caller population may affect the accuracy of the CPL *i.e.,* it becomes harder to guess the correct number from a larger callers pool.

To illustrate the impact of the increase of the caller population on the CPL, I randomly select one user in my datasets as an example shown in Fig. 7.9 where the vertical axis represents the accuracy of the CPL, and horizontal axis represents the cumulative caller

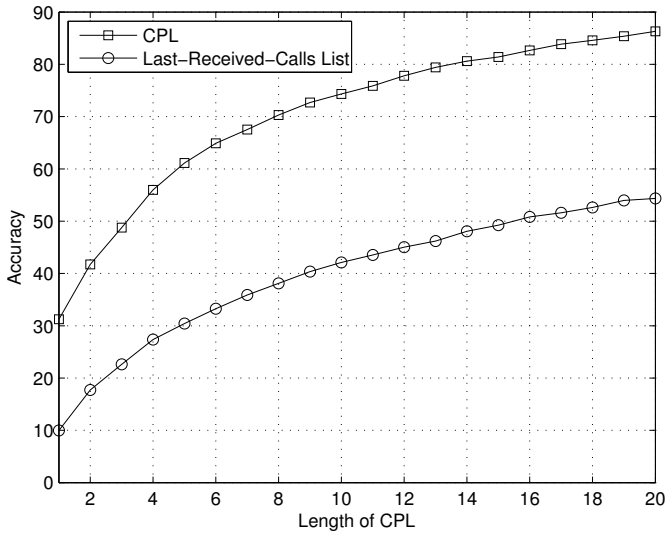


FIGURE 7.8. Overall performance of the CPL comparing to the conventional Lat-20-Received-Calls list.

population that continues to increase from 41 callers to 70 callers. It shows that the accuracy decreases dramatically as the caller population becomes larger for different length of the list ($L = 1, 5, 10, 15, 20$). The accuracy drops with relatively higher rate for the shorter length of the list as one may expect.

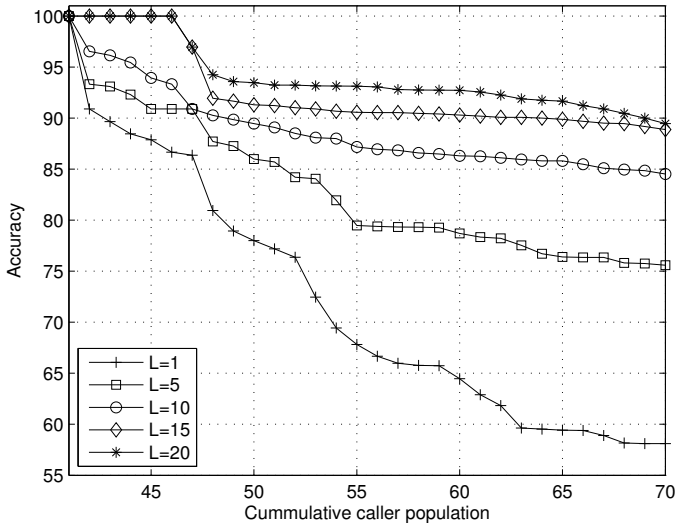


FIGURE 7.9. A demonstration of the impact of the increasing cumulative caller population on the accuracy of the CPL.

7.4.3. Impact of New Callers

In the meanwhile, the new callers or the first-time callers (whose call received for the first time) also have a negative impact on the performance of the CPL. This may be a bigger issue for those users who are more social and those who are unfortunately on numerous telemarketers' lists. This is a voice spam problem, which is expected to increase severely, especially in the VoIP networks where the cost of communication is extremely low with the absurdly large IPv6 address (can supports 2128 addresses). To demonstrate the impact of the new callers, I examine the accuracy of the CPL without considering the new callers *i.e.,* if the caller is the first-time caller then it is not taken into account for the accuracy computation. After the first call however the caller will be recognized and taken into account for accuracy computation as normal. *It can be seen from Fig. 7.10 that the accuracy of the CPL is indeed improved about 10% as the new callers are not considered.*

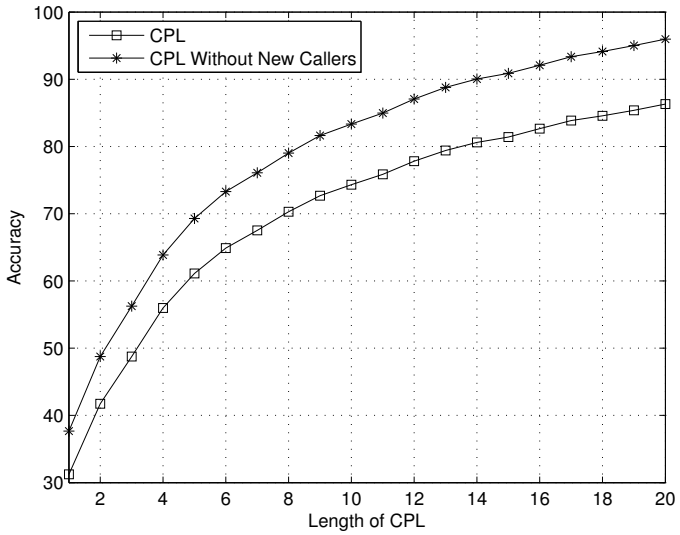


FIGURE 7.10. Overall performance of the CPL with and without considering the new callers.

If I modify my definition or criterion for the new callers by redefining the new caller to be the caller who has called C times in the past, then I observe that as variable C increases the accuracy of CPL also increases accordingly (shown in Fig. 7.11). This unsurprising result

implies that the CPL can predict more accurately for the callers whose behaviors have been learned for a longer period of time.

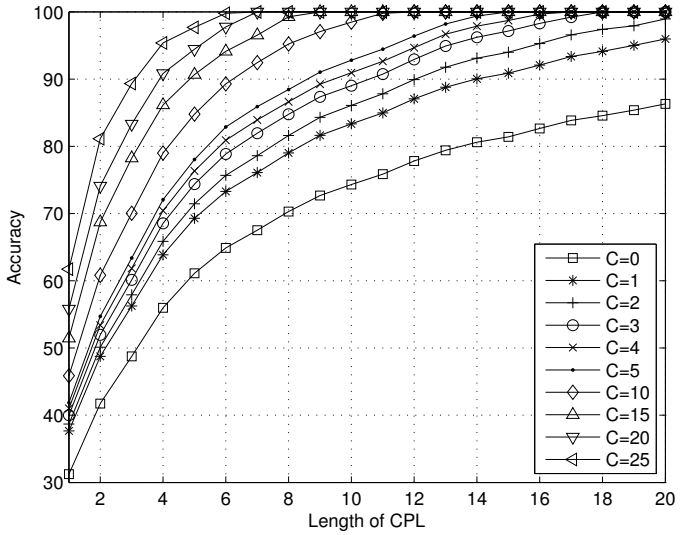


FIGURE 7.11. The impact of the new callers to the accuracy as the criterion of new caller (C) varies from 0 to 25.

7.4.4. Impact of Mobile Social Closeness

I can further extend the concept of the new callers to infer the "social closeness". The number of incoming calls alone can only be used to quantify the social closeness to some extent. In social science, the social closeness of people has been discussed and found that it can be based on the amount of time and the intensity (frequency) of communication (103)(104). Granovetter (103) suggests that the time spent in a relationship and the intensity along with the intimacy and reciprocal services form a set of indicators for social tie. Marsden and Cambell (104) evaluate the indicators and predicators of strength (tie) described by Granovetter (103) and conclude that "social closeness" or "intensity" provides the best indicator of strength or tie.

In mobile social network, the amount of time and the intensity of communication can be measured by the call duration (talk time) and the call frequency (number of phone calls).

In our daily life, we communicate with people in the mobile network at different instances. These people constitute our mobile social network. Based on amount of time and intensity

of communication with these people, our mobile social network can be divided into three broad groups:

Group 1: Socially Closest Members – These are the people with whom we maintain the highest socially connectivity. Most of the calls we receive, come from individuals within this category. We receive more calls from them and we tend to talk with them for longer periods. Typically, the face-to-face social tie of these people is family member, friend, and colleagues.

Group 2: Socially Near Members – People in this group are not as highly connected as family members and friends, but when we connect to them, we talk to them for considerably longer periods. Mostly, we observe intermittent frequency of calls from these people. These people are typically neighbors and distant relatives.

Group 3: Socially Distant Members – These individuals have less connection with our social life. These people call us with less frequency. We acknowledge them rarely. Among these would be, for example, a newsletter group or a private organization with whom we have previously subscribed. This group also includes individuals who have no previous interaction or communication with us. We have the least tolerance for calls from them e.g., strangers, telemarketers, fund raisers.

I quantitatively define the social closeness between user i and user j from the user i's perception ($S(i,j)$) by Eq. 63.

$$S(i,j) = \sqrt{(1 - F(i,j))^2 + (1 - T(i,j))^2}, \tag{63}$$

where $F(i,j)$ is the normalized call frequency (normalized to the maximum call frequency among all users with whom user i communicate) between user i and user j, which is given by Eq. 64, and $T(i,j)$ is the normalized call duration or talk time (normalized to the maximum talk time among all users with whom user i communicate) between user i and user j, which is given by Eq. 65.

$$F(i,j) = \frac{f(i,j)}{\max\limits_{k \in U_i}\{f(i,k)\}}, \tag{64}$$

$$T(i,j) = \frac{t(i,j)}{\max\limits_{k \in U_i}\{t(i,k)\}}, \tag{65}$$

where $f(i,j)$ is the total number of calls or call frequency between user i and user j, $t(i,j)$ is the total call duration or talk time between user i and user j, and $U_i = \{1, 2, ..., N\}$ is the set of all users associated with user i (*i.e.,* all users who have made/received calls to/from user i with total of N users).

Therefore, $S(i,j)$ has values in the range $[0, \sqrt{2}]$, which indicates the mobile social closeness between user i and user j from user i's perspective where 0 implies the closest and $\sqrt{2}$ implies the farthest relation. Based on this quantity, I can categorize all users associated with user i into three social groups using a simple grouping algorithm as follows.

Let R denote the Euclidean distance from coordinate (μ_F, μ_T) to (1,1) where μ_F and μ_T are the means of $F(i,j)$ and $T(i,j)$, respectively and $j \in U_i$. If $S(i,j) \leq R/2$, then user j belongs to Group 1, if $R \geq S(i,j) > R/2$, then user j belongs to Group 2, and if $S(i,j) > R$, then user j belongs to Group 3.

To validate the accuracy of my social closeness/grouping computation, I use the second set of my data described in Sect. 7.3.1. During my second dataset collecting process, I interviewed the subjects about the social closeness for all of his/her associated users by having the subjects identified for each associated user (caller/callee ID) the perceived social group. Each participant received $20 as compensation. As the result, my second dataset includes additional information of social group corresponding to each associated user.

After comparing my calculation against the user feedback, I am able identify social groups correctly with the overall accuracy rate of 93.8%. The detailed result is shown in Table 7.1, which presents number of correct classification (Hit), number of incorrect classification (Miss), and the accuracy rate (Hit/(Hit + Miss)) for each user. Based on the follow-up interviews with these ten subjects, most of "Miss" are caused by confusion between the face-to-face social closeness and mobile social closeness. For example, one of the subjects indentifies his roommate as a group 1 member since the subject sees and talks with his roommate on daily basis, the subject however does not make/receive many phone calls

to/from him. As the result, his roommate is classified to group 2 based on my calculation (Eq. 63) but identified as group 1 member by the subject. To avoid biased feedbacks from the subjects, I did not provide any information about my social closeness computation or much more details about the three social groups than the description provided earlier in this section. Nevertheless, I believe that I have a decent result in accuracy rate and, in addition, I do not have any incorrect classification that misses more than one level of social group.

TABLE 7.1. The result of social group calculation of each user.

User	Hit	Miss	Accuracy Rate (%)
1	60	5	92.31
2	57	6	90.48
3	48	5	90.57
4	141	13	91.56
5	127	8	94.07
6	188	11	94.47
7	88	3	96.70
8	80	6	93.02
9	62	1	98.41
10	87	4	95.60
Overall	938	62	93.80
Mean	93.80	6.20	93.72
Std. Dev.	44.82	3.61	2.64

To see the impact of the social closeness on the CPL, Fig. 7.12 shows the overall accuracy rate versus the length of the CPL for different social ties; group 1, 2, and 3. The CPL performs better in accuracy for the callers with closer social tie.

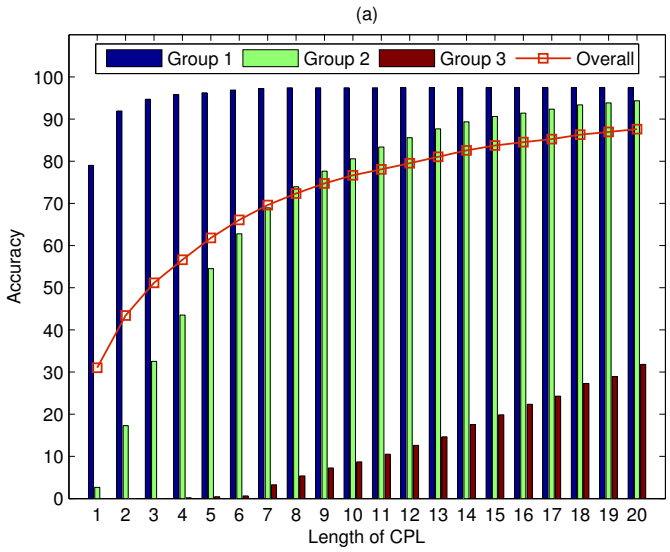


FIGURE 7.12. The overall accuracy of the CPL as an incoming call predictor for different lengths of the list as well as for different social groups.

7.4.5. Impact of Change of Life's Schedule

Since call logs represent human behavior associated with trends and changes of behavior over time, thus the accuracy of the CPL can also be impacted by the change of the caller's life schedule because it changes the calling pattern towards the user. For example, your friend changes job from working Monday through Thursday from 8AM to 5PM to working Friday through Sunday from 6PM to 3AM. This major change of your friend's life schedule may result in totally different calling pattern towards you, from receiving several calls at night and on weekends to several calls during the day and on weekdays, for instance. The change of calling pattern of several callers could degrade the performance of the CPL even more.

7.4.6. How fast can CPL become reliable?

How fast can the CPL learn to become a reliable predictor for its user? This is an important question to answer. In attempt to answer this question, I monitor the accuracy of the CPL as the learning time or usage time (*i.e.,* number of days since that user starts using CPL) increases. I find that the accuracy normally starts with a low value, fluctuates, then gradually increases, and eventually becomes more stable at some level. The answer to the

question of when the CPL will become a reliable predictor or when the accuracy will become stable, is not trivial. Of course, for CPL being reliable predictor does not necessarily mean that it has perfect accuracy (100%) but rather it has a stable accuracy (*i.e.,* small variation). The accuracy level when it becomes stable as well as the time that takes for the accuracy to become stable may depend on various factors such as number of incoming calls per day, structure of caller's calling pattern, and aforementioned factors that impact the accuracy (*i.e.,* increase of caller population, new callers, change of caller's calling pattern).

Nonetheless, I demonstrate the relationship between the learning time and the accuracy of CPL by plotting accuracy as learning time (number of days) increases for three different sample users (randomly selected from my dataset) with different incoming call rates in Fig. 7.13, Fig. 7.14, and Fig. 7.15 where their number of incoming calls per day are 15.65, 5.61, and 2.05 respectively for different length of the predicted list ($L = 1, 5, 10$, and 20).

As I previously speculated that one of many possible factors that may determine how fast the accuracy to become stable was the rate of incoming calls or number of calls received per day. Since other factors such as structure of caller's calling pattern and change of caller's calling pattern are harder to identify and are difficult to quantify for comparison among users, therefore I can restrict my attention to just incoming call rate and assume for a moment that other factors are approximate the same for all users.

In fact, it is evident in Fig. 7.13, 7.14, and 7.15 that the accuracy of CPL becomes stable faster for higher incoming call rate. *It is reasonable because the more calling information (higher incoming rate) that CPL learns, the quicker CPL recognizes caller's calling pattern.*

7.4.7. Unpredictability of Calls

The accuracy rate of the CPL can be impacted by many different factors as mentioned previously. One of the factors that has a high impact on the accuracy of the CPL is the "randomness" of the calling pattern of each caller.

Randomness or uncertainty associated with a random variable has been studied and defined as the information entropy by Claude E. Shannon (159) as follows.

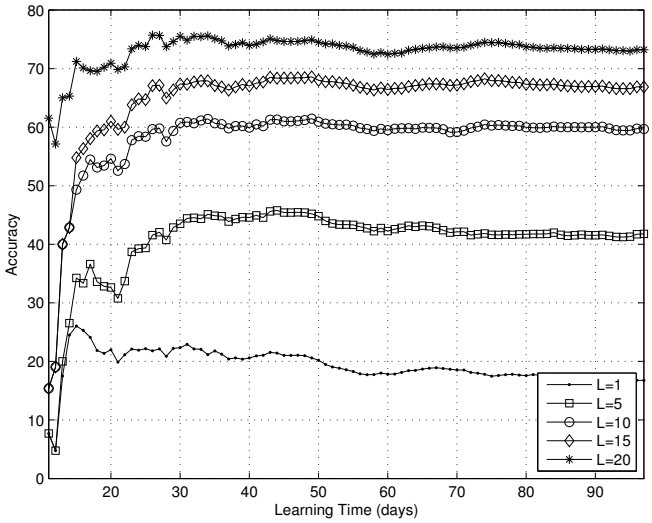


FIGURE 7.13. The accuracy of CPL as learning time increases for sample user who receives averagely 15.65 calls per day.

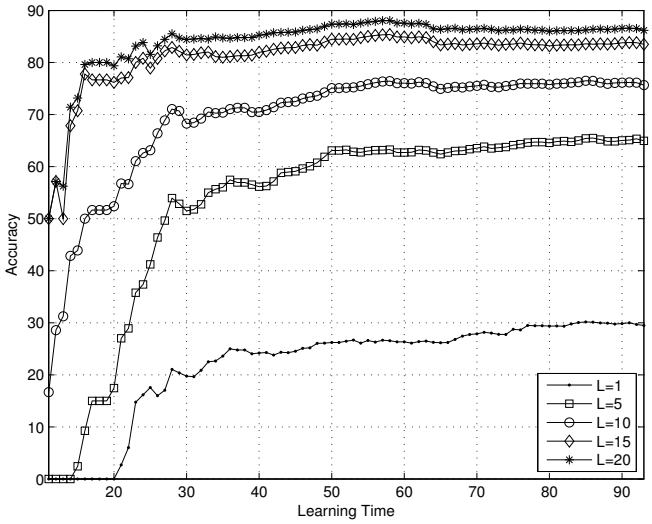


FIGURE 7.14. The accuracy of CPL as learning time increases for sample user who receives averagely 5.61 calls per day.

$$E(X) = -\sum_i p(x_i) \log_2 p(x_i), \tag{66}$$

where $E(X)$ is an entropy of random variable X where $x_i \in X$ and $p(x_i) = Pr(X = x_i)$.

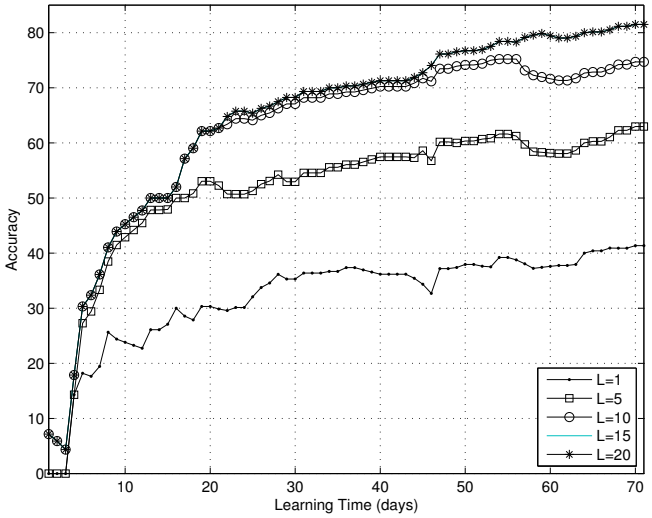


FIGURE 7.15. The accuracy of CPL as learning time increases for sample user who receives averagely 2.05 calls per day. Note that accuracy curve for $L = 15$ is equal to $L = 20$.

By adopting this information entropy, I define the *Unpredictability of Incoming calls* (UI) as the sum of the entropy of each caller such that UI increases with randomness of each caller as well as the number of possible callers. The unpredictability of incoming calls for the user k (UI_k) is given by Eq. 67.

$$UI_k = \sum_{k}^{N} \left(- \sum_{h}^{24} p_k(h) \log_2 p_k(h) \right), \tag{67}$$

where N is the total number of callers and

$$p_k(h) = \frac{C(T_k H_h)}{\sum_{h=1}^{24} C(T_k H_h)}. \tag{68}$$

I compute the UI_k for each user in my dataset ($k = 1, 2, 3, \ldots, 30$). Fig. 7.16 shows that the accuracy rate of the CPL at $L = 5$ decreases unsurprisingly with the unpredictability.

7.4.8. CPL as an Outgoing Call Predictor

With the same framework, the CPL can function as an outgoing call predictor. I find that the analyses that have been done so far for the incoming call predictor is also valid

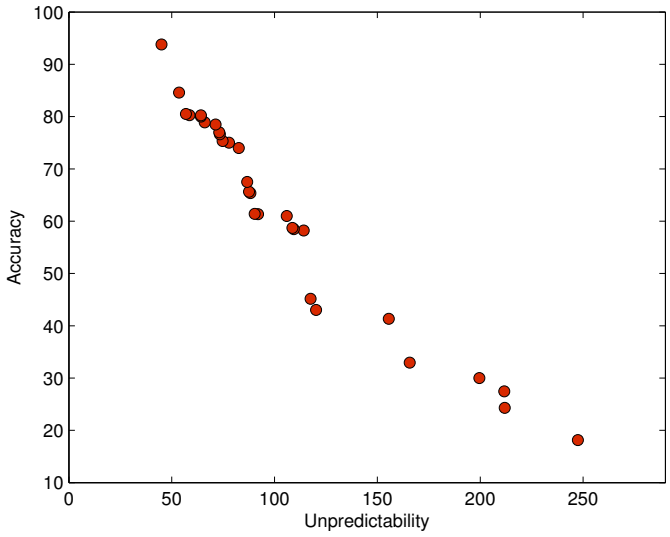


FIGURE 7.16. The overall accuracy rate of CPL as an incoming call predictor decreases with the unpredictability of incoming calling patterns.

for the outgoing call predictor. Figure 7.17 shows the overall accuracy rate of the CPL as an outgoing call predictor with and without considering the "new callees". About 10% improvement in accuracy is also evident. Figure 7.18 shows the accuracy rate of CPL as an outgoing call predictor for different social groups. A similar result to the incoming call predicted list's is also obtained here where the CPL predicts much more accurately for the callees who are within a closer social tie. Figure 7.19 shows the accuracy of CPL as an outgoing call predictor decreases with the unpredictability of the user's outgoing calling pattern. As expected, the accuracy rate decreases with the unpredictability of the outgoing calling pattern.

7.5. Applications of CPL

To demonstrate the usefulness of CPL besides its own features, I describe here two applications of CPL including Call Firewall and Call Reminder.

7.5.1. Call Firewall

By adopting the concept of firewall–the wall that keeps destructive forces away from our computer systems, Call Firewall basically monitors and handles incoming calls by keeping

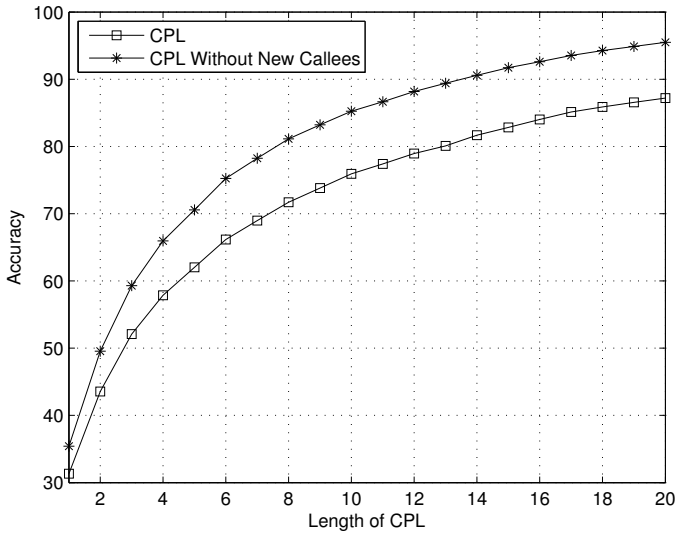


FIGURE 7.17. Overall performance of the CPL as an outgoing call predictor with and without considering the new callees.

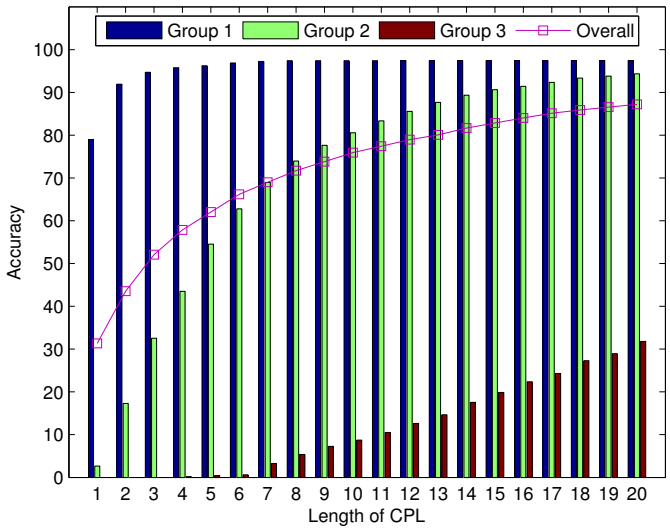


FIGURE 7.18. The overall accuracy of the CPL as an outgoing call predictor for different lengths of the list as well as for different social groups.

unsolicited and unwanted calls away while allowing desired calls to pass through. The problem of unwanted telemarketing calls or spam calls is expected to be a serious problem especially in VoIP networks due to its much lower communication cost than the circuit-switched telephone network system (it also becomes an attractive target for spammers). In

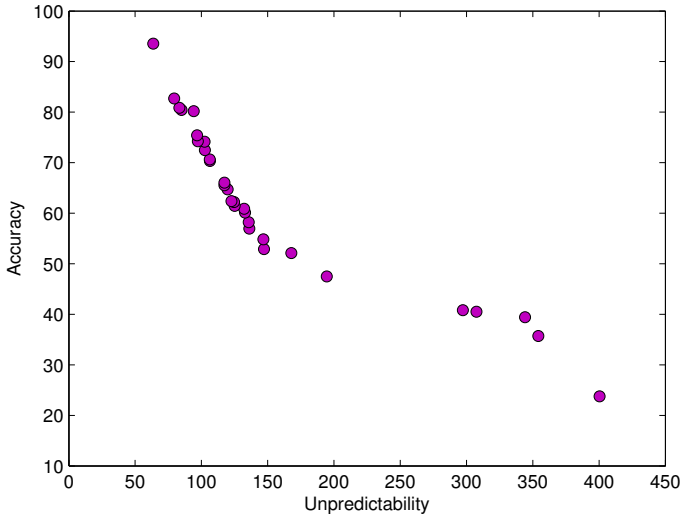


FIGURE 7.19. The overall accuracy rate of CPL as an outgoing call predictor decreases with the unpredictability of outgoing calling patterns.

fact, SPIT (Spam over Internet Telephony) is roughly three orders of magnitude cheaper to generate than traditional circuit-based telemarketing calls (160). Unlike e-mail spam, call spam is a real-time problem, which requires a real-time defense mechanism. The real challenge is thus to block the spam call before the phone rings. Not only these spam calls create nuisance for the user, Kolan et al. (105) showed that each incoming phone call created different level of nuisance depending on the user's presence (mood or state of mind) based on situational, spatial, and temporal contexts. Therefore, to address this problem of unwanted calls, the system for detecting voice spam and estimating spamminess level (known as VoIP Spam Detector or VSD) described by Kolan and Dantu (135) and Dantu and Kolan (136) and the nuisance computation model (known as Nuisance Detector or ND) proposed by Kolan et al. (105) can be integrated with the call prediction model proposed in this article (CPL) to proactively handle incoming calls before the phone rings. VSD, as described in (135) and (136), is a multi-stage adaptive spam filer based on presence (location, mood, time), trust, and reputation to spam in voice calls. It uses a close-loop feedback control between different stages to detect a spam call. As described in (105), ND is a model for computing nuisance

level of incoming calls based on the social closeness and other behavioral patterns such as periodicity of the caller and reciprocity.

As shown in Fig. 7.20, CPL generates a periodic 24-hour call prediction to be fed into VSD to learn behavior of callers (among which are spammers) and analyze the trustworthiness (VSD indicates the untrusted calls to be "dropped") and ND computes nuisance level associated with each predicted call (ND determines each call to be either sent directly to "voicemail" or "ringer" to ring the phone), then a set of firewall rules is generated *e.g.*, IF John calls between 10am-11am, THEN forward it to voicemail, IF Pizza House calls between 4pm-5pm, THEN drop the call. The firewall rules are updated periodically (can be as often as every hour – depending on the user). The user can also provide feedbacks about the actual nuisance level or reporting spam calls in order to improve the performance of the firewall.

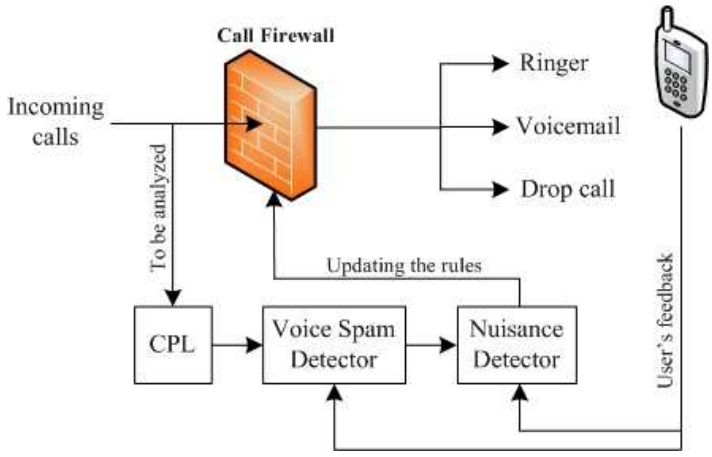


FIGURE 7.20. System overview of Call Firewall constructed with CPL, VSD, and ND for proactively handling the incoming calls.

To show the performance of the Call Firewall, an experiment is conducted with nine-month call logs of 30 users randomly selected from MIT dataset (81) using the latest 60 days for testing while keeping the rest of the data for training. Table 7.2 shows the false negative rate, true negative rate, and true positive rate of all 30 users. False negative rate measures the percentage of the incoming calls that pass through the Call Firewall but should have been

blocked. I assume that all "Missed Call" in my dataset mean that the user does not want to take the call and hence it should be blocked by the Call Firewall. Despite many other reasons for the missed calls such as being away from the phone, not hearing the ringer, and forgetting to switch the phone back to ringer from silent mode, I carry out the experiment with this assumption. True negative rate is a percentage of correctly blocked calls by Call Firewall *i.e.*, (number of blocked calls)/(number of predicted calls to be blocked). True positive rate is a percentage of calls that are correctly let through by Call Firewall *i.e.*, (number of pass-through calls)/(number of predicted calls to be allowed to pass through firewall). Based on this experiment, the Call Firewall performs with the average false negative rate of 10.3981%, true negative rate of 75.6991%, and true positive rate of 83.0321%.

7.5.2. Call Reminder

One of the common problems of everyday life is forgetting to make a phone call that could either be an event-based call such as birthday call, meeting planning call, etc. or a nonevent-based call such as calling parents on weekends, calling girlfriend/boyfriend during a lunch break, etc. Therefore, besides the Intelligent Address Book – an automatic function that computes the probability of outgoing calls based on the recent calling behavior and generates a list of potential callees to help avoid searching for a number to call through a typical lengthy address/contact book, I present here a Call Reminder that makes use of CPL as an outgoing call predictor by integrating it with ND and Event Calendar to generate a "reminder" for the user to place a call to a particular person based on the user's past history, nuisance level, and events.

As shown in Fig. 7.21, CPL periodically makes outgoing call prediction (*e.g.*, hourly), which will be mapped onto the nuisance level computed by ND. The result is then evaluated by the decision maker to generate the call reminder *e.g.*, high probability and low nuisance level would imply prompting a call reminder. The event calendar (a function that normally comes with today's mobile phone) is used to provide details about the call reminder *e.g.*, birthday call, meeting plan, project discussion, etc. The user would be prompted with a reminding message such as "Would like to call John about the ABC conference?", "Would like

TABLE 7.2. The experimental result of the performance of the Call Firewall.

Phone user	False negative (%)	True negative (%)	True positive (%)
1	7.9167	60.5657	81.1345
2	13.6364	70.6599	78.7067
3	20.1220	68.9306	85.9894
4	18.5185	73.4268	84.3915
5	2.8571	71.2957	73.0827
6	6.7805	90.5238	89.2000
7	30.6250	70.9195	68.8482
8	29.0909	73.9037	74.4131
9	5.1049	62.6718	80.4147
10	17.0000	61.2121	74.1722
11	0	85.5233	83.5608
12	26.4151	75.4762	86.7951
13	0	95.4762	100
14	3.3333	74.9153	85.2941
15	3.6364	69.6364	74.5827
16	2.5641	70.0632	81.7840
17	6.6667	61.3043	70.0880
18	30.1887	79.5116	77.1222
19	21.5686	79.4787	80.9110
20	4.5455	68.6689	90.8333
21	0	77.4271	84.6699
22	2.1739	69.8182	82.2090
23	12.1951	75.6426	78.1295
24	0	93.5915	94.1000
25	7.1429	86.4356	92.8571
26	0	71.7921	94.5113
27	8.4337	86.1017	86.8067
28	18.3019	76.2393	82.7376
29	11.9048	88.4960	85.3029
30	1.2195	81.2656	88.0456

to call Alice about the birthday?", "Would you like to call Mom regarding about dinner?". The user records new events into the event calendar for future reminders. Feedback sensor forwards the actual outgoing calls to CPL to be analyzed for prediction as well as provides the user's feedback to BD to calibrate nuisance computation.

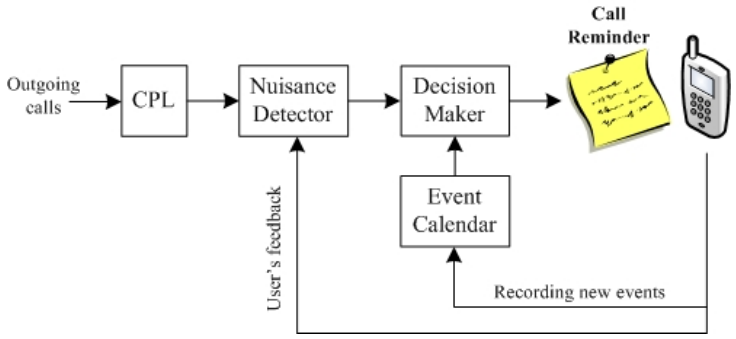


FIGURE 7.21. System overview of Call Reminder constructed with CPL, ND, and Event Calendar for reminding the user to place a call.

To see the performance of the Call Reminder, I conduct an experiment with call logs of 30 users from MIT dataset where the latest 60 days are testing period while the rest of the data are used for initial training. Since there is no event calendar information in my dataset, the performance is based solely on CPL and ND. The goal is to measure the percentage of the calls made because of the prompted reminders generated by the Call Reminder. my assumption here is that each outgoing call needs to be reminded. Clearly, it is not completely realistic. Nonetheless, to get the first glance at how Call Reminder would perform in real life, I conduct the experiment on this assumption. Therefore, with my dataset, I verify for each outgoing call if it would be reminded by the Call Reminder. Table V shows the result of the true positive rate, which is computed as ratio of the number of actual outgoing calls made that are among the five numbers/contacts reminded by the Call Reminder to the total number of outgoing calls. Based on this experiment, the Call Reminder performs with the average true positive rate of 69.2654%. Note that I believe that the performance can be improved relatively with an event calendar. Moreover, in more realistic setup, only

some outgoing calls should be reminded. To see the real performance of the proposed Call Reminder, one would be interested in finding out if the user does make an outgoing call when a call reminder is generated. This among others will be in my future study.

7.6. Related Work

There have been some works on predictive modeling for telephone call demands. In (161), the authors apply the queuing theory to characterize queuing primitives such as the arrival time process, the service-time distribution, and the distribution of customer impatience. In (162), the author develops two variations of Poisson process models for describing count data of call center arrivals which utilized the proposed mixed models technique. There is also a work describing a predictive model for the emergency 9-1-1 call volumes in (163), where the authors used a multiple linear regression model technique to construct the multi-dimensional linear predictor based on the call history. The work that is fairly close to my work is (164) where the authors develop a system for predicting a future communication activity based on the past communication event information. The system analyzes the past communication event information (including phone calls and emails) to determine whether a correlation existed in the past communication and predicted the future communication event based on the current communication event and the correlation. The correlation is computed based on the pattern of incoming and outgoing calls e.g., if a call received from "person A" resulted in a later origination of a call to "person B," the correlation value between the "person A" and the "person B" is increased proportionately and the correlation values corresponding to other persons not dialed is decreased accordingly. The work that is closest to my work is (143) where the authors proposed a Call Predictor (CP), which computes receiving call probability and makes the next-24-hour incoming call prediction based on caller's behavior and reciprocity. The caller's behavior is measured by the caller's call arrival time and inter-arrival time. The reciprocity is measured by the number of outgoing calls per incoming call and the pattern of inter-arrival/departure time. The CP only makes prediction for a pre-specified caller of when the caller will be calling in the next 24-hour time frame. In contrast, my CPL predicts the next-hour callers by generating a list of the potential callers. With

TABLE 7.3. The experimental result of the performance of the Call Reminder.

Phone user	True positive (%)
1	70.5202
2	86.1386
3	100
4	49.6000
5	54.0230
6	54.7445
7	76.3247
8	98.7775
9	62.3919
10	70.7127
11	51.0397
12	72.1470
13	85.4530
14	61.5542
15	56.8110
16	65.8784
17	53.0387
18	92.0854
19	72.9358
20	100
21	90.9091
22	52.0833
23	47.9433
24	55.9934
25	53.7975
26	66.5761
27	73.8872
28	80.5000
29	52.5180
30	69.5767

CPL, user needs not to request prediction for each caller but with one request (*i.e.,* press "predict" button), a list of potential callers will be generated. The main contrast between the CP and CPL is that CP predicts "when" the caller will make a call to the user but CPL predicts "who" will be the caller/callee.

7.7. Conclusion

With the advancement of technologies embedded in today's mobile phones, people begin to engage the mobile phones more and more into many parts of their lives. Today's technology suggests that the mobile phone will eventually become a personal assistant that intelligently provides useful information to help its user making good decisions or even make decisions based on the user's context with the goal of to enhance quality of life. As a step towards this direction, I present here a model for predicting future callers and callees envisaged as a Call Predicted List (CPL). CPL makes use of the user's call history to build a probabilistic model of calling behavior based on the calling patterns and reciprocity. As an incoming call predictor, CPL is a list of numbers/contacts that are the most likely to be the callers within the next hour. As an outgoing call predictor, CPL is generated as a list of numbers/contacts that are the most likely to be dialed when the user attempts to make an outgoing call (by flipping open or unlocking the phone). This helps save time from having to search through a lengthy phone book. The CPL has been evaluated with the real-life call logs from 30 mobile users and it shows a promising result in accuracy.

In this study, I have learned that the phone calls that seem random and unpredictable, it actually can be predicted accurately to some extent. I have also learned that there are however numerous factors that can impact the accuracy of the predictor such as the increase number of callers/callees, the new callers/callees, the mobile social closeness, the change of life schedule, the activeness of callers/callees, and the randomness of the calling pattern.

I am also aware of some limitations of this study such as the size of my datasets and the length of the call logs. I find it very difficult to collect these call logs from the subjects due to the privacy issues and the amount of time taken by each interview (about the social closeness) during my second dataset collection. Each interview lasted about one hour, which

included downloading call logs from the phone service website and collecting feedback about the social closeness. There were only three months of call logs available for downloading from the service provider web page thus I am limited to three months of data for my analysis.

As my future direction, I will continue to investigate other parameters to improve my model as well as continue to collect more data for my future studies.

CHAPTER 8

CONTEXT-AWARE ALERT MODE FOR A MOBILE PHONE

8.1. Introduction

Having a handheld device recognized its user's context fits to the scope of context awareness, which is one of the hottest current research areas. Context awareness aims to enhance our quality of life with intelligent computing devices sensing and reacting to the environment and presence of users. Existing handhled devices such as mobile phones and personal digital assistants (PDA's) have already taken steps towards this computing paradigm.

With the embedded sensors in today's mobile phones such as accelerometer, GPS, and audio sensor, the user's context can be sensed and estimated to some extent using machine learning techniques. In this chapter, I design and evaluate a context-aware mobile computing model, known as *ContextAlert*, that intelligently configures the mobile phone alert mode according to user's situational context. For example, the phone can be automatically set to vibrate mode while the user is in a meeting, automatically configured to handsfree mode while the user is driving, etc.

Several previous works have been done in context awareness such as service discovery, online/mobile social modeling for providing better services, activity recognition, personal/object positioning, and person identification. My work is closely related to activity recognition for which previous works have used either a single or multiple sensors attached to different parts of user's body.

We distinguish this work from other previous works by the following contributions:

(1) I use multiple sensors embeded in the mobile phone, which is more realistic for detecting the user's context than having various sensors attached to different parts of user's body.

(2) I propose a model that uses sensed contextual information to provide a service that better synchronizes the user's daily life with a context-aware alert mode control. With this service, the user can avoid the problems such as forgetting to switch to vibrate mode while in a meeting or a movie theater, and taking the risk of picking up a phone call while driving.

(3) As adaptivity is essential for context-aware computing, within my model I propose a learning mechanism that maintains a constant adaptivity rate for new learning while keeping the catastrophic forgetting problem minimal.

The rest of the chapter is organized as follows: Section 8.2 briefly reviews the literatures in context awareness that is related to my work. Section 8.3 presents the system overview of *ContextAlert*. Section sec:framework describes my proposed framework for *ContextAlert*. My approach in designing *ContextAlert* is evaluated with several experiments and the results are shown in Section 8.5. I point out some limitations of my work in Section 8.6. Section 8.7 concludes this chapter with a summary and an outlook on future work.

8.2. Related Work

Context-aware computing research is scoped by ubiquitous computing, a term that was coined by Mark Weiser (1; 165). It has also been referred to as pervasive computing and ambient intelligence, which is a computing paradigm that makes multiple computing devices available throughout the physical environment and effectively invisible to the user. Several researchers have attempted to define "context" (48; 49; 50; 51; 52) since Schilit et al. (166; 167) first introduced it in 1994. Han et al. (55) divided context into physical, internal, and social context.

Several works in physical context have focused on service discovery in ubiquitous computing environments based on the user's context *e.g.*, (168; 169; 170; 171; 172; 173; 174; 175). Meanwhile, research in the social context area has been reported in both online and mobile social networks by modeling social dynamics and using social context information to provide better service for users *e.g.*, (176; 177; 178; 179; 180; 81; 181; 182). My work is in the

area of internal context, which was defined as an abstract thing inside people such as feeling, thought, task, action, interest, and so on (55). Recent works include context extraction (*e.g.*, (183; 184; 185; 186)), activity recognition (*e.g.*, (187; 188; 189)), personal/object positioning (*e.g.*, (190; 191)), and person identification (*e.g.*, (192; 193; 194)).

Laerhoven and Cakmakci (195) proposed a context-awareness system that learned the user's activities from 2-axis accelerometers, passive infrared sensors, carbon monoxide sensor, microphones, pressure sensors, temperature sensors, touch sensors, and light sensors using Kohonen self-organizing maps and Markov models.

Lester et al. (196) presented a method using accelerometers to determine if two devices were carried by the same person based on a coherence function (a frequency-domain linear correlation).

Lukowicz et al. (197) presented a technique to automatically track the progress of maintenance or assembly tasks using body-worn 3-axis accelerometers, microphones, and computers based on frequency-matching sound classification technique that combined the intensity analysis of signals from microphones at different parts of body and correlation analysis of surrounding sounds and user activity.

Bao and Intille (187) developed a system to detect activities such as walking, sitting, standing, running, and so on using body-worn 2-axis accelerometers based on mean energy, frequency-domain entropy, and correlation and decision tree classifiers.

Krause et al. (72) presented a multi-sensor wearable system that learned context-aware personal preferences by identifying individual user states and observing how the user interacted with the system in these states. This work was based on the previous model proposed by Siewiorek et al. (183). Sensor data were preprocessed using different methods such as fast Fourier transform (FFT) and principal component analysis (PCA), and then clustered using Kohonen self-organizing maps (198) and Markov models.

Jin et al. (199) proposed a context awareness system that distinguished user motion states and recognized emergency situations using a 2-axis accelerometer, heat flux sensor,

galvanic skin response sensor, skin temperature sensor, and near-body ambient temperature sensor based on a fuzzy inference model.

These recent works in internal context area adopt the wearable computer approach, which requires several sensors to be attached to specific parts of the user's body to sense the most accurate context data. These approaches are thus not realistic. Nevertheless, the preprocessing techniques, machine learning approaches, and probabilistic models used in these works are useful.

There are some recent studies reported in recognizing activities using an accelerometer attached to the mobile device. Iso and Yamazaki (200) proposed a gait analyzer based on a 3-axis accelerometer mounted on a mobile phone using a wavelet packet decomposition for preprocessing data and a self-organizing map with Bayesian theory for classification. Yi et al. (201) conducted a study to determine what contextual information could be obtained from a 3-axis accelerometer attached to a personal digital assistant (PDA) by having subjects performed some activities while carrying PDAs.

8.3. Context-Aware Alert Mode Control

The user's context is very complex to be comprehended entirely from sensor data. I nevertheless believe that it can be estimated and interpreted to some extent. With the embedded sensors in the mobile phones such as the accelerometer, GPS, and audio sensor, the user's movement, mobility, and ambient noise level can be sensed respectively.

We propose here a Context-Aware Alert Mode Control (*ContextAlert*) that configures the call alert to the most suitable mode corresponding to user's context. With today's mobile phones, the user has three call alert options: ringer, vibrate, and handsfree. These options are suitable for different situations. Handsfree mode (bluetooth headset) is most suitable when the user is driving a vehicle. In fact, many states in the U.S. have prohibited drivers from talking on mobile phones while driving (202). Vibrate mode is most suitable when the user is in a meeting, theater, library, etc. Ringer is used mostly in general and it is preferred in situations in which the user can be interrupted by a ringer such as while shopping, having lunch, or walking in a park.

Our notion of context is therefore defined as a user's physical situation, which is a cluster of feature attributes obtained from the sensor data at an interval of time. Accordingly, the user's context can be divided into three states:

(1) *Uninterruptible by Ringer (UR)*: In this state, user does not want to be interrupted by a ringer. Normally, this situation occurs while user is in a considerably quiet place with low movement and mobility *e.g.*, in a meeting, in a theater, at a library, etc.

(2) *Interruptible by Ringer – Vehicular Mode (IR-V)*: The user can be interrupted by a ringer in this state but is unable to use hands to operate the phone. This is usually a driving situation, in which the environmental noise level is typically higher than in the UR state. The movement is normally low but the mobility is clearly high.

(3) *Interruptible by Ringer – Non-vehicular Mode (IR-N)*: This state corresponds to situations in which user is interruptible by a ringer and not driving a vehicle. Situations include shopping in a mall, walking with friends in a hall way, having lunch, and jogging in a park. These situations are typically at high ambient noise level, high movement, and low mobility.

With these user context states, *ContextAlert* sets the alert option to the most suitable mode according to Definition 8.1, learns the user's preference from the feedback, and adjusts the inference engine accordingly (shown in Fig. 8.1).

Definition 8.1. If the user's context is UR, then the most suitable alert option is vibrate mode. If the user's context is IR-V, then the most suitable alert option is handsfree mode. If the user's context is IR-N, then the most suitable alert option is ringer mode.

8.4. Framework

This section describes the *ContextAlert* framework, which includes my approach in designing the models and details on sensor data acquisition, data preprocessing, the context classifier, the inference engine, and the adaptive learning mechanism.

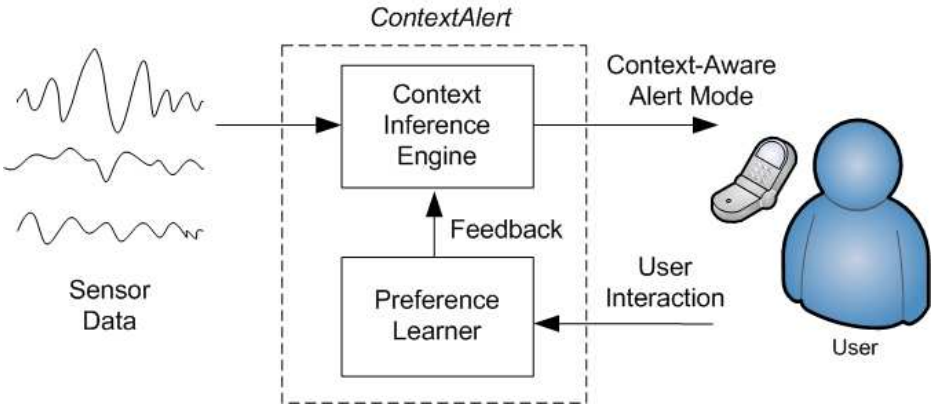


FIGURE 8.1. System overview of *ContextAlert*.

8.4.1. A Three-Step Approach

There are three steps in my design. The first step is "training," which is an offline supervised learning process to construct an initial context map by classifying the labeled training samples into three different context states (UR, IR-V, and IR-N). In this step, sensor data are preprocessed to obtain useful features (details are described in Section 8.4.3), then fed into classifier to generate an initial context map using PCA (details are described in Section 8.4.4). The second step is "inferring," which is an online unsupervised learning process to analyze input sensor data and infer the user's context state based on k-nearest neighbor algorithm (k-NN) and finite state machine model (details are described in Section 8.4.5). The third step is "user preference learning," which is an online supervised learning process to learn the user's preferences based on the feedback (details are described in Section 8.4.6). This three-step approach is illustrated in Fig. 8.2. On top of the three-step approach, a learning algorithm is applied for the system to remain adaptive for new learning while the Catastrophic forgetting problem is maintained at a minimum (details are described in Section 8.4.7).

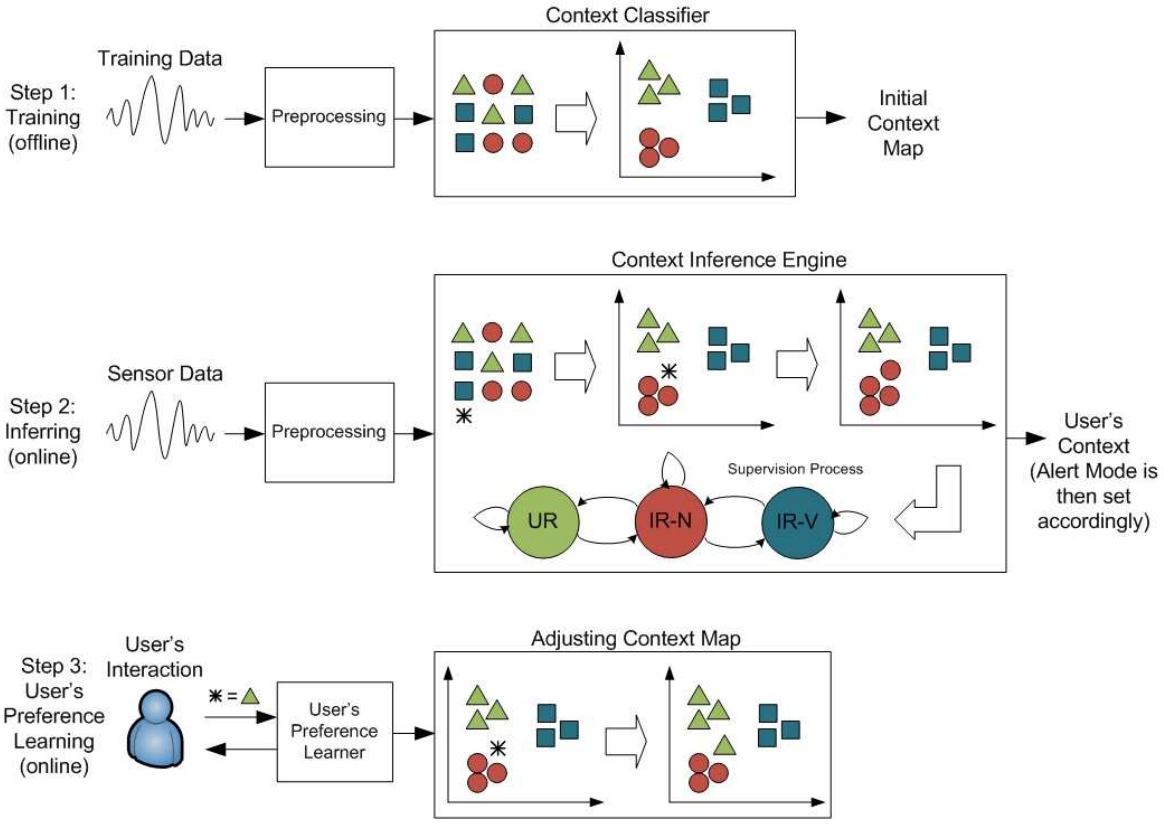


FIGURE 8.2. Our three-step approach constructs an initial context map using supervised learning in the training step, then uses the initial map to estimate user's context in the inferring step, and learns user's preference from the feedback.

8.4.2. Data Acquisition

In this work, the data were collected using the embedded sensors of the G1 phones (203) with Google Android 1.1 operating system, 32 bit Qualcomm MSM7201A (528 MHz CPU clock), 256 MiB ROM, and 192 MiB RAM. These sensors included a 3-axis accelerometer, a GPS navigation system with Qualcomm MSM7201A gpsOne using NIMEA 0183 protocol, and an audio sensor with 16 bit nominal quantization and a sampling frequency of 44,100 Hz.

To acquire data from these sensors, I created an application for G1 phone using Android 1.1 SDK (204). The phone was carried inside the front pants pocket while the data were collected.

8.4.3. Preprocessing Methods

Preprocessing is needed to extract useful features from the raw sensor data. To estimate the user's movement, I compute the magnitude of the force vector by combining the measurements from all three axes using Eq. 69 to derive a net acceleration (a) independent of orientation. Note that if there is no movement, the magnitude is approximately at $1G$ due to the Earth's gravity (9.8 m/s^2).

$$a = \sqrt{c_x^2 + c_y^2 + c_z^2}, \tag{69}$$

where c_x, c_y, and c_z are measurements from x, y, and z axis of accelerometer, respectively.

Figure 8.3 shows the net acceleration's magnitude of a subject walking, standing, running, and sitting. The subject carried the phone in his pant pocket for this experiment and all other experiments in this chapter.

For estimating the user's mobility, I use GPS data to compute the traveling speed by calculating a distance (minimum distance or length of a displacement) between user's current position and the previous one based on the latitude and longitude information. Then the user's traveling speed can be obtained simply by dividing the distance by a time difference between two positions as given by Eq. 70.

$$s = \frac{\sqrt{(\phi_1 - \phi_2)^2 + (\lambda_1 - \lambda_2)^2} \times 111}{\triangle T}, \tag{70}$$

where ϕ_i and λ_i denote a latitude and a longitude value at location i, respectively. The constant 111 is the approximated converting ratio of distance from one geographic degree to kilometer unit. Time difference between two locations is represented by $\triangle T$ in hour units.

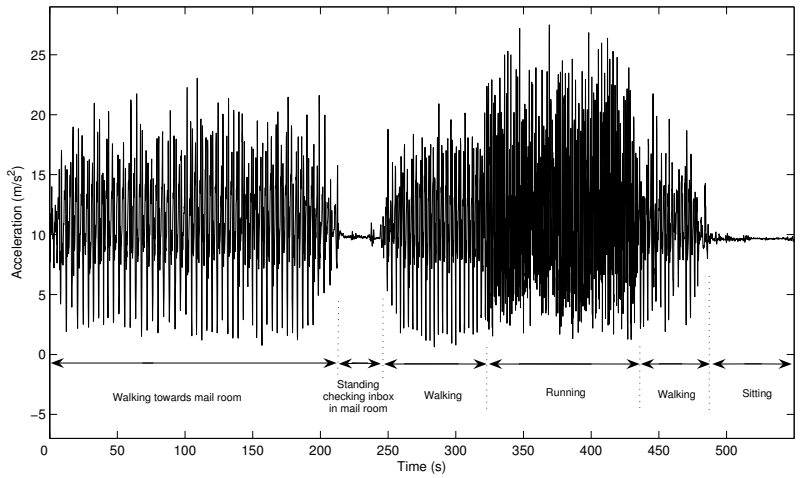


FIGURE 8.3. An example of magnitude of the force vector by combining the measurements from all three axes from accelerometer. Data show the subject walking to a mail room, checking his mail box, walking/running/walking back to an office, and sitting down on a chair.

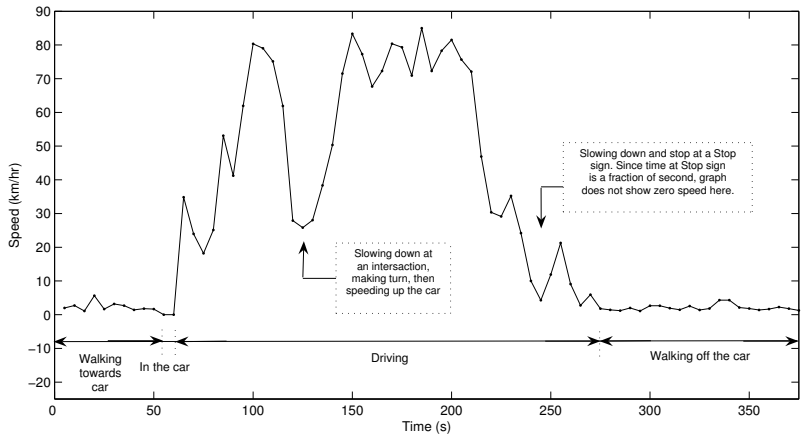


FIGURE 8.4. An example of traveling speed based on GPS information. Data show the subject walking towards a car, driving, then walking away from the car as he reaches the destination.

An example of the traveling speed based on GPS data is illustrated in Fig. 8.4 as a subject walking to a car, driving to the destination, then walking away the car as he arrives.

For the audio sensor data, I sample the audio signal at 8 kHz and extract the running average envelope, which gives us a smoother signal (less noise) than its original signal and

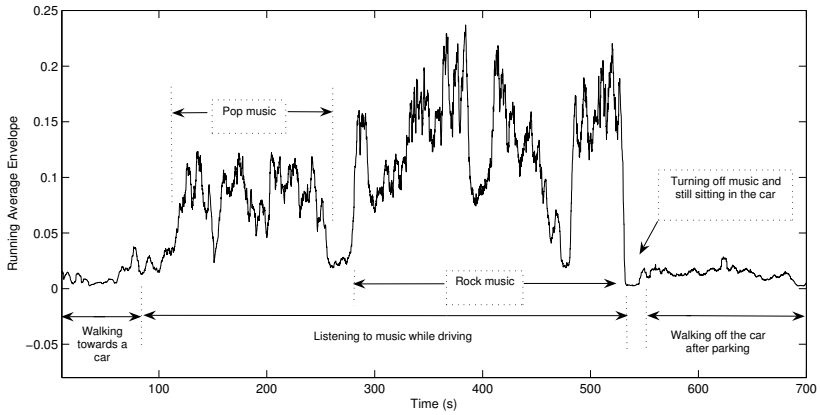


FIGURE 8.5. An example of running average envelope while a subject is walking to a car, driving with music on, and walking away from the car after parking.

peak envelope as my interest in the loudness of the ambient noise (amplitude of the audio signal). I compute the running average envelope (e) with window size of 50 using Eq. 71.

(71)
$$e(n) = \frac{1}{w}\{g(n-w)+g(n-w+1)+...+g(n-1)+g(n)+g(n+1)+...+g(n+w-1)+g(n+w)\},$$

where $g(n)$ is the amplitude of audio signal with $n = \{1, 2, 3, ...\}$ and w is the size of window.

Figure 8.5 shows an example of the running average envelope while a subject is walking to a car, listening to music while driving, and walking away from the car after parking.

8.4.4. Context Classifier

The context classifier is used in the offline training process (step 1) to take preprocessed data and project them onto feature space creating an initial "context map" with M trained data arrays.

With my preprocessed data, the (labeled) input data array of the classifier (x) at any interval of time T can be expressed as follows.

$$x_m = \begin{bmatrix} Var(A_m) \\ E(S_m) \\ E(E_m) \end{bmatrix}, \tag{72}$$

where $A_m = \{a_m(1), a_m(2), ..., a_m(n_a)\}$, $S_m = \{s_m(1), s_m(2), ..., s_m(n_s)\}$, $E_m = \{e_m(1), e_m(2), ..., e_m(n_e)\}$, and n_a, n_s, n_e are the total numbers of data points within T of A_m, S_m, and E_m, respectively. I take the variance ($Var(\cdot)$) of A_m and expected values ($E(\cdot)$) of S_m and E_m. Hence the training data matrix for constructing the initial context map is $X_{trained} = \{x_1, x_2, ..., x_M\}$.

To project my training data onto a context map, I apply PCA (205). I transform my three-dimensional input data to two-dimensional feature space by retaining two principal components that have the maximum variation in the original data array, namely the first and second principal components *i.e.*,

$$Y = W_C' X, \tag{73}$$

where Y is the data on a transformed space (or context map in my case), X is the data matrix, and W_C is the first C singular vectors ($C = 2$ in my case) where $W = [w_1\ w_2\ ...\ w_p]$ (p is the original data's dimensionality, *e.g.*, $p = 3$ in my case), the order of w is according to the variance or eigen value *i.e.*, $var(w_i' X) = \lambda_i$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$, and $'$ denotes transpose.

8.4.5. Context Inference Engine

The context inference engine is in the online inferring process (step 2), which takes a new (unlabeled) preprocessed data array along with the trained data arrays, projects them onto context space, and makes an initial classification for the new data based on k-NN algorithm (206) using the Euclidean distance. The initial classification is then fed into a transition supervision process based on a finite state machine model to make the final inference.

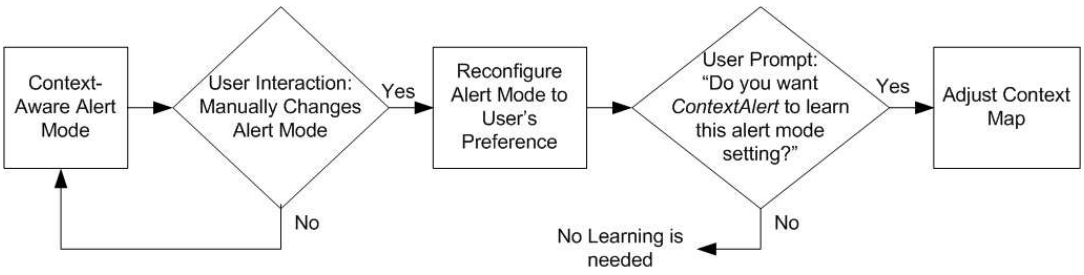


FIGURE 8.6. User preference learning process flow.

Thereby, the input data matrix for the PCA is $X = \{X_{trained}, x_{new}\}$ such that the new and trained data are transformed by the same function. With the new coordinates, the new data is then classified to the most likely context state (Z), which is the most common class amongst the k nearest neighbors in the context space, where $Z \in$\{UR, IR-V, UR-N\}. The initial classified context state then undergoes the supervision process to supervise the transitions from one context state to another. This supervision process uses a finite state machine architecture where each state represents the user's context and transitions are represented by edges between states.

8.4.6. User Preference Learning

This is a process of learning the user's personal preference. It is inevitable that the initial context map does not fit perfectly to the user's preference. This process is therefore essential to personalizing *ContextAlert*. The process can be as simple as the flow diagram shown in Fig. 8.6. Once the user makes a change to the alert mode, the user will be prompted to have *ContextAlert* learn his/her setting. If the answer is "Yes," then the context map is adjusted accordingly. However, if the answer is "No," then no learning is needed and hence the context map is not modified.

8.4.7. Adaptive Learning

With the proposed model, *ContextAlert* would start out highly adaptive with a high learning rate, would gradually become fixed as number of learned data increases. After this

stage, it would be hardly capable of learning any more, which would create a problem as the system needs to remain adaptive. Overwriting previously learned data with the new learning can improve the adaptivity of the system. However, the tradeoff is known in the field of machine learning as the *Stability-Pasticity Dilemma* or *Catastrophic Forgetting*(207), which refers to the problem of designing a learning system to remain plastic or adaptive and preserve its previously learned knowledge while continuing to learn new things, which can also mean preventing the new learning from washing away the memories of prior learning.

To address this challenge in designing a context awareness system, I propose a learning mechanism that remains adaptive while keeping Catastrophic Forgetting minimal. The adaptivity (Λ) can be defined simply as a learning rate for new data as

$$\Lambda = \frac{\text{Amount of New Learning Data}}{\text{Amount of Learned Data}}. \tag{74}$$

In my case, the amount of new learning data is one and the amount of learned data is $M/3$ for each context state. Thus the learning rate decreases exponentially with M. To stay adaptive, M must be fixed and hence removing previously learned data is an option. In this approach, I cannot avoid the Catastrophic Forgetting problem. Nevertheless, I can minimize it.

Forgetting is a loss of memories, which can be quantified as a difference between the set of prior memories before and after a new learning. Let $\xi^{(b)}$ and $\xi^{(a)}$ denote the set of prior memories in three-dimensional space before and after a new learning, respectively.

Definition 8.2. If $x_k^{(b)}$ is the k^{th} memory point before a new learning in three-dimensional space (d_1, d_2, d_3) and $x_k^{(a)}$ is the k^{th} memory point after a new learning, then the difference between the $x_k^{(b)}$ and $x_k^{(a)}$ (γ_k) can be computed using Euclidean distance as

$$\gamma_k = \sqrt{(x_k^{(b)}(d_1) - x_k^{(a)}(d_1))^2 + (x_k^{(b)}(d_2) - x_k^{(a)}(d_2))^2 + (x_k^{(b)}(d_3) - x_k^{(a)}(p_3))^2} \tag{75}$$

If $x_k^{(a)}$ does not exist (or has been removed), then $\gamma_k = \infty$ (complete loss of memory).

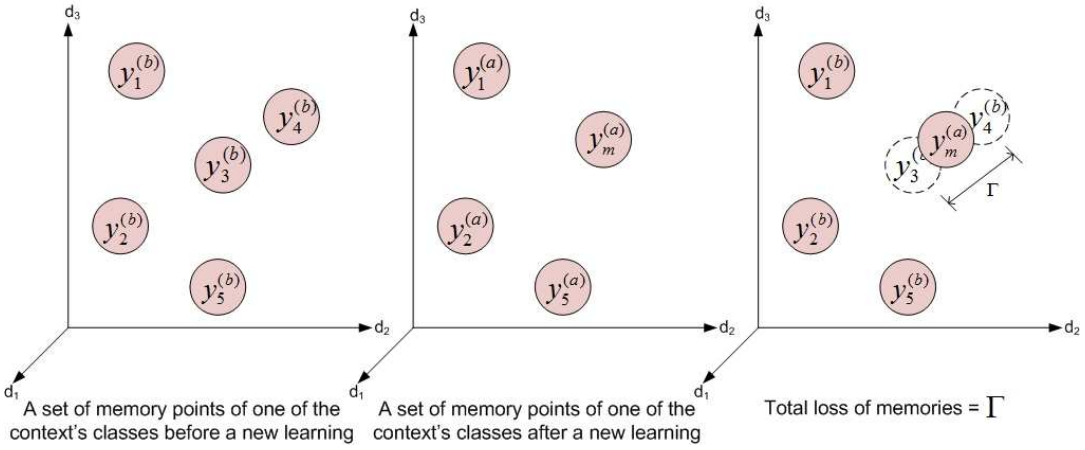


FIGURE 8.7. An example of graphical representation of the merging process for adaptive learning.

DEFINITION 8.3. If $\xi^{(b)} = \{x_1^{(b)}, x_2^{(b)}, ..., x_M^{(b)}\}$ and $\xi^{(a)} = \{x_1^{(a)}, x_2^{(a)}, ..., x_M^{(a)}\}$, then the total loss of memories (Γ) is the sum of γ_k for $k = 1, 2, ..., M$, *i.e.*,

$$\Gamma = \sum_{k=1}^{M} \gamma_k. \tag{76}$$

To minimize the loss of memory from a new learning, I merge two nearest memory points to one memory point located at the mid point between the two. If $x_m^{(a)}$ is the merging of $x_i^{(b)}$ and $x_j^{(b)}$, then the total loss of memory is

$$\begin{aligned} \Gamma &= \sqrt{(x_i^{(b)}(d_1) - x_m^{(a)}(d_1))^2 + (x_i^{(b)}(d_2) - x_m^{(a)}(d_2))^2 + (x_i^{(b)}(d_3) - x_m^{(a)}(d_3))^2} \\ &\quad + \sqrt{(x_j^{(b)}(d_1) - x_m^{(a)}(d_1))^2 + (x_j^{(b)}(d_2) - x_m^{(a)}(d_2))^2 + (x_j^{(b)}(d_3) - x_m^{(a)}(d_3))^2} \\ &= 2\sqrt{(x_i^{(b)}(d_1) - x_m^{(a)}(d_1))^2 + (x_i^{(b)}(d_2) - x_m^{(a)}(d_2))^2 + (x_i^{(b)}(d_3) - x_m^{(a)}(d_3))^2}, \end{aligned} \tag{77}$$

and the merged memory point is occupied at $(\frac{x_i^{(b)}(d_1))+x_j^{(b)}(d_1)}{2}, \frac{x_i^{(b)}(d_2))+x_j^{(b)}(d_2)}{2}, \frac{x_i^{(b)}(d_3))+x_j^{(b)}(d_3)}{2})$. As an example, a graphical representation of the merging process is illustrated in Fig. 8.7.

To summarize my design, a detailed algorithm of the *ContextAlert* is given in Algorithm 8.1.

ALGORITHM 8.1. Context-Aware Alert Mode

Input: Context map's data matrix ($X_{trained}$) and input data array (x_{new})

Output: Context-aware alert mode and a new context map's data matrix ($X_{new\_trained}$)

1. Project $X = \{X_{trained,x_{new}}\}$ onto the context space using PCA *i.e.,* $Y = W_C' X$;
2. Classify y_{new} (transformed x_{new}) to the context $Z \in$ {UR,IR-V,IR-N} using k-NN;
3. Set the alert mode according to Definition 8.1;
4. IF There is an overwrite setting by the user
5. Prompt the user to have the system learned the setting;
6. IF Answer is 'Yes'
7. Reclassify y_{new} according to the new setting;
8. ELSE
9. Do nothing;
10. END IF
11. END IF
12. $X_{new\_trained}$ = Merging of two nearest memory points of x_{new}'s class and the rest of $X_{trained}$;
13. Return the context-aware alert mode;
14. Return $X_{new\_trained}$;

8.5. Experimental Results

In this section, I describe my datasets (Section 8.5.1) as well as conduct four experimental studies to evaluate my approach. In Section 8.5.2, I show the impact of learning by comparing the performance of a model that uses only a fixed initial context map (no learning from new data) to a model that starts off with the same initial context map but its context map grows as it learns new data. In Section 8.5.3, I once again point out that the growth of the context map with new learnings lowers the adaptivity rate and causes "the curse of dimensionality". I thus compare the performance of a model with growing context map with my proposed merging-based context map model. In Section 8.5.4, I show the impact of the proposed

supervision process that can improve the performance of the model. In Section 8.5.5, I show the impact of applying PCA to my model by comparing the performance of my model with and without using PCA.

8.5.1. Datasets

For training, I collected data from three different subjects. Each subject performed ten different activities shown in Table 8.1. Each activity was performed continuously for ten minutes by each subject. With a time interval (T) of five seconds (buffer time), I had 120 labeled data arrays. With ten different activities and 120 data arrays per subject, I thereby had 360 labeled data arrays available for constructing the initial context map.

TABLE 8.1. A list of the ten different activities and their corresponding context states. Four participating subjects performed ten minutes of each activity from which the training data arrays were obtained.

Context State	Activity
UR	Attending a meeting
	Attending a class
	Watching movie at a theater
	Reading books in a library
	Working in an office
IR-N	Walking
	Jogging or Running
	Eating at a restaurant
	Shopping at a supermarket
IR-V	Driving a car

For testing, I collected data from a different group of participating subjects. There were four subjects in this testing group. Each subject performed five different sequences of activities, which are listed on Table 8.2. Each sequence was about one hour. These sequences

consisted of all ten activities listed on Table 8.1, 31,690 seconds (6,338 data arrays) of UR, 21,430 seconds (4,286 data arrays) of IR-N, 18,940 seconds (3,788 data arrays) of IR-V, and total of 14,412 testing data arrays.

The subjects were asked to keep detailed time logs of activities performed, which were then used to do hand-labeling of the testing data.

TABLE 8.2. A list of five different sequences of activities with the corresponding context state and approximate duration. Each sequence was about one hour. Testing data arrays were obtained from having each of four subjects performed these sequences.

Sequence Number	Sequence of activities with the corresponding context state and approximate duration
1	Jogging (IR-N, 3 min.) ⇒ Walking (IR-N, 2 min.) ⇒ Library (UR, 25 min.) ⇒ Walking (IR-N, 5 min.) ⇒ Driving (IR-V, 25 min.)
2	Walking (IR-N, 5 min.) ⇒ Driving (IR-V, 30 min.) ⇒ Walking (IR-N, 10 min.) ⇒ Theater (UR, 15 min.)
3	Walking (IR-N, 5 min.) ⇒ Library (UR, 25 min.) ⇒ Walking (IR-N, 10 min.) ⇒ Restaurant (IR-N, 20 min.)
4	Meeting (UR, 25 min.) ⇒ Walking (IR-N, 3 min.) ⇒ Running (IR-N, 2 min.) ⇒ Class (UR, 30 min.)
5	Working (UR, 10 min.) ⇒ Walking (IR-N, 5 min.) ⇒ Driving (IR-V, 25 min.) ⇒ Walking (IR-N, 5 min.) ⇒ Shopping (IR-N, 15 min.)

8.5.2. Impact of Learning

Ideally, I would like to have a model with one fixed context map that works perfectly for any user but this is not realistic. Therefore, in this section, I attempt to show that such a model performs well to some extent. However, I can improve its performance by learning from the user.

This experiment and others were set up as follows. The initial context map was constructed using 100 data arrays from each of the three context states by randomly selecting training data arrays obtained from the subjects in the training group. The model was tested with the data obtained from the five sequences of activities by four subjects described in Section 8.5.1. The testing was done in the order of the sequence, *i.e.* testing with sequence 1, then sequence 2 then sequence 3, then sequence 3, and so on.

Table 8.3 shows the overall performance from four testing subjects in terms of accuracy rates per sequence (Acc./Seq.) as well as per context (Acc./Cont.) of a fixed initial context map model (FCM), which uses only the initial context map without learning from the user. The result per subject is shown in the Appendix.

TABLE 8.3. Performance of FCM in terms of hits and misses for each context state and each testing sequence of activities. The accuracy rate per context state (Acc./Cont.) is shown in the bottom of the table while the accuracy rate per sequence (Acc./Seq.) is shown in the last column.

Sequence	UR		IR-N		IR-V		Overall		Acc./Seq.
Number	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	(%)
1	1138	0	316	50	1012	108	2466	158	93.98
2	750	0	638	76	1356	108	2744	184	93.72
3	1252	0	564	1112	0	0	1816	1112	62.02
4	2686	0	208	84	0	0	2894	84	97.18
5	512	0	594	644	18	1186	1124	1830	38.05
Total	6338	0	2320	1966	2386	1402	11044	3368	76.63
Acc./Cont. (%)	100.00		54.13		62.99		76.63		

Without learning, the FCM shows 76.63% overall accuracy. With the same initial context map, Table 8.4 shows that I can achieve a much higher accuracy rate of 90.55% with a growing-with-learning context map model (GCM) that keeps all new learning data arrays as it is being tested. Hence the context map grows with learning (the amount of testing data).

I assume here that the user corrects all misclassified data arrays (Step 3 of the three-step approach) so that the GCM does not mislearn the data.

TABLE 8.4. Performance of GCM in terms of hits and misses for each context state and each testing sequence of activities. The accuracy rate per context state (Acc./Cont.) is shown in the bottom of the table while the accuracy rate per sequence (Acc./Seq.) is shown in the last column.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	1136	2	360	6	1078	42	2574	50	98.09
2	750	0	708	6	1464	0	2922	6	99.80
3	1246	6	1024	652	0	0	2270	658	77.53
4	2628	58	272	20	0	0	2900	78	97.38
5	500	12	1002	236	882	322	2384	570	80.70
Total	6260	78	3366	920	3424	364	13050	1362	90.55
Acc./Cont. (%)	98.77		78.53		90.39		90.55		

8.5.3. Adaptivity and The Curse of Dimensionality

A much higher accuracy rate of the GCM comes at a price. As the context map grows with learnings, its adaptivity decreases exponentially (according to Eq. 74). This also increases the computational cost as the cost of k-NN rises with the number of learned data, which is a problem known as "the curse of dimensionality".

The adaptivity of GCM can be computed using Eq. 74 as the average over three context data arrays as $\Lambda = \frac{1}{3}(\frac{1}{6,338} + \frac{1}{4,286} + \frac{1}{3,788}) = 0.000218$. With my proposed merging-based context map model (MCM), which merges the two nearest learned data arrays after each new learning in the context map, Table 8.5 shows that I can achieve a competitive accuracy rate compared with the GCM at 89.34% (about one percent lower). However, the important improvement of the MCM over the GCM is a much higher adaptivity rate of $\Lambda = \frac{1}{100} = 0.01$.

As the MCM prevents the adaptivity from decreasing while minimizing the loss of prior learning, the accuracy stays reasonably high with the context map that remains adaptive.

TABLE 8.5. Performance of MCM in terms of hits and misses for each context state and each testing sequence of activities. The accuracy rate per context state (Acc./Cont.) is shown in the bottom of the table while the accuracy rate per sequence (Acc./Seq.) is shown in the last column.

Sequence	UR		IR-N		IR-V		Overall		Acc./Seq.
Number	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	(%)
1	1128	10	362	4	1082	38	2572	52	98.02
2	750	0	706	8	1464	0	2920	8	99.73
3	1246	6	958	718	0	0	2204	724	75.27
4	2646	40	266	26	0	0	2912	66	97.78
5	506	6	788	450	974	230	2268	686	76.78
Total	6276	62	3080	1206	3520	268	12876	1536	89.34
Acc./Cont. (%)	99.02		71.86		92.93		89.34		

8.5.4. Impact of Supervision Process

With the supervision process, the context state transition is properly guided *e.g.*, if the user is currently driving (IR-V), in the next five sections he/she will either be driving (IR-V) or walking away from the car (IR-N); he/she cannot be in a meeting or class. Adding the supervision process can help improve the accuracy of the model. In fact, experimental results in Table 8.6 show that the accuracy rate of the MCM with the supervision process (MCM-S) is improved to 91.20%, which is higher than the MCM and GCM (with much better adaptivity rate than the GCM).

TABLE 8.6. Performance of MCM-S in terms of hits and misses for each context state and each testing sequence of activities. The accuracy rate per context state (Acc./Cont.) is shown at the bottom of the table while the accuracy rate per sequence (Acc./Seq.) is shown in the last column.

Sequence	UR		IR-N		IR-V		Overall		Acc./Seq.
Number	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	(%)
1	1128	10	362	4	1120	0	2610	14	99.47
2	750	0	706	8	1464	0	2920	8	99.73
3	1246	6	958	718	0	0	2204	724	75.27
4	2646	40	266	26	0	0	2912	66	97.78
5	506	6	788	450	1204	0	2498	456	84.56
Total	6276	62	3080	1206	3788	0	13144	1268	91.20
Acc./Cont. (%)	99.02		71.86		100.00		91.20		

8.5.5. Impact of PCA

Typically, PCA is used to reduce the dimensionality of a dataset consisting of a large number of interrelated variables. In my case, PCA is used not only to reduce the dimensionality of my data matrices but to also reduce the noise of from the embedded sensors (several reports in PCA-based noise filtering *e.g.*, (208), (209), and (210)). This noise reduction process can help improve the performance the classifier and hence improve the accuracy of the model. Without applying PCA, the accuracy of MCM-S is decreased to 85.64% as shown by the experimental results in Table 8.7.

We have conducted several experimental studies to evaluate my approach in designing a model for *ContextAlert*. To summarize the results, Table 8.8 shows the overall accuracy of each model. I have shown that "learning" improves the accuracy of the model but decreases adaptivity, the "merging-based model" helps maintains adaptivity with a reasonable accuracy rate, the "supervision process" is a key element that improves performance, and "PCA" is used to reduce sensor noise that can degrade the performance of the model. From

TABLE 8.7. Performance of MCM-S(no PCA) in terms of hits and misses for each context state and each testing sequence of activities. The accuracy rate per context state (Acc./Cont.) is shown at the bottom of the table while the accuracy rate per sequence (Acc./Seq.) is shown in the last column.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	1138	0	290	76	1120	0	2548	76	97.10
2	750	0	580	134	1464	0	2794	134	95.42
3	1250	2	534	1142	0	0	1784	1144	60.93
4	2686	0	208	84	0	0	2894	84	97.18
5	512	0	606	632	1204	0	2322	632	78.61
Total	6336	2	2218	2068	3788	0	12342	2070	85.64
Acc./Cont. (%)	99.97		51.75		100.00		85.64		

Table 8.8, my proposed model (MCM-S) has the highest performance in both accuracy rate and adaptivity.

Note that the result per subject of each model is available in the Appendix.

TABLE 8.8. Overall performance comparison of different models in terms of adaptivity and accuracy rate.

Model	Average Adaptivity	Accuracy Rate (%)
FCM	0.01	76.63
GCM	0.0000218	90.55
MCM	0.01	89.34
MCM-S	0.01	91.20
MCM-S(no PCA)	0.01	85.64

8.6. Limitations of the Study

We are aware of the following limitations of this study:

(1) The evaluation of the user's preference learning step (Step 3 of the three-step approach) cannot be done in this current study due to the capability of the current model of the G1 phone.

(2) With a limited number of testing data, I can only demonstrate the impact of learning and adaptivity to some extent. I are certain that a much longer period of testing would yield much clearer results than the current study.

(3) Similarly, with a larger number of testing subjects, the performance of my model would have been evaluated more accurately. In this study, I have learned that it is very difficult to recruit subjects to perform sequences of experiments in extended hours due to availability, willingness, and enthusiasm.

8.7. Conclusion

Forgetting to switch to vibrate mode while in a movie theater or a meeting, and taking the risk of picking up a phone call while driving can be avoided if the phone is smart enough to recognize its user's situational context. As the first step towards that direction, I propose a design for a context-aware mobile computing model known as *ContextAlert* that can intelligently switches the alert mode according to the user's context. I divide the user's context into three states: Uninterruptible by ringer (UR), Interuptible by ringer - vehicular mode (IR-V), and Interuptible by ringer - none-vehicular mode (IR-N). The alert mode is to be set to the recognized context state as vibrate, handsfree, and ringer mode for UR, IR-V, and IR-N, respectively. I have proposed a three-step approach in design based on the embedded sensor data from accelerometer, GPS antenna, and microphone of a G1 phone. I have evaluated my model in several aspects using training and testing data collected from participating subjects. Based on the experiments, the proposed model has shown a promising result. Nevertheless, my work had some limitations, such as capability of the phone, amount of testing data, and duration of testing. In my future work, I will continue to examine my model to improve its performance as well as investigate other applications of the model.

CHAPTER 9

ADEQUACY OF DATA FOR CHARACTERIZING CALLER BEHAVIOR

9.1. Introduction

Telecommunication device such as telephone has moved beyond being a mere technological object and has become an integral part of many people's social lives.

This has had profound implications on both how people as individuals perceive communication as well as in the patterns of communication of humans as a society. Learning human behavior has always been the subject of interest in scientific fields (e.g. (211), (212), and (213)). There are also scientific reports in learning and characterizing user and network behavior (e.g. (214), (215), and (86)).

In communication systems, a user can be a "caller" who initiates communication or a "callee" who receives request for a communication from caller. As a callee in a phone network, a user generally has received calls from several callers. I are interested in learning caller behavior. A knowledge of caller behavior can lead to a predictive model which forecasts or predicts the future behavior of the caller such as calling time and hence useful for scheduling and planning (e.g., it can be used to avoid unwanted calls and schedule time for wanted calls). It can also be useful for the Public Safety Answering Point (PSAP) for predicting 9-1-1 (emergency) calls. It can also be beneficial to voice spam detection and prevention, as well as call centers for resource utilization.

Predictive models derived from communication logs have been studied extensively (e.g. (216), (217), and (148)). Recently there has been growing interests in the field of mobile social networks analysis to study human behavior by combing the computer technology and social networks (e.g. (218), (86), (152), and (155)), but due to the unavailability of data, there have been far fewer studies. The Reality Mining Project at Massachusetts Institute of

Technology (80) has made publicly available large datasets which I use for my analysis in this chapter.

Motivation

In (143), authors proposed a Call Predictor which made the next-24-hour incoming call prediction based on caller behavior and reciprocity which were extracted from call history. This raises a question of how much call history is actually needed. Does it mean the more historical data, the better performance of the predictor? To answer this question, I find it interesting to study caller behavior and the adequacy of caller's past history.

Main Contribution

The main contribution of this chapter is to infer the adequacy of historical call data to capture the behavior of the caller in order to construct a predictive model for future behavior observation.

9.2. Real-Life Dataset and Analysis

In our daily life, we receive phone calls from family members, friends, colleagues, supervisors, neighbors, and strangers. I believe that every caller exhibits a unique calling pattern which characterizes the caller behavior.

To study the caller behavior, I use the real-life datasets of 94 individual call logs over nine months of the mobile phone users which were collected at Massachusetts Institute of Technology (MIT) by the Reality Mining Project (80). These 94 individuals are faculties, staffs, and students. The datasets include people with different types of calling patterns and call distributions.

Each call record in the datasets has the 5-tuple information which includes:

- Date (date of call)
- Start time (start time of call)
- Type (type of call i.e., "Incoming" or "Outgoing")
- Call ID (caller/calee identifier)
- Talk time (duration of call).

We use the call logs to derive the traffic profiles for each caller by inferring the Arrival time (time of receiving call from the caller), Inter-arrival time (elapsed time between adjacent incoming calls from the caller), and Talk time (duration of call from the caller).

9.2.1. Arrival Time

Based on my real-life datasets of 94 mobile phone users with more than 2,000 combined callers, I can divide callers into two categories namely Single-peak callers and Multi-peak callers based on their arrival time.

9.2.1.1. *Single-peak Callers.* The single-peak callers are callers who tend to make more calls at around one particular time of the day and less and less number of calls as time of the call deviates from that time (favorite time). Thus, I make a hypothesis that call arrival time has a normal distribution $N(\mu, \sigma^2)$ where μ is the mean and σ^2 is the variance of call arrival time which can be calculated by Eq. 78 and Eq. 79 respectively.

$$\mu = \frac{1}{N}\sum_{n=1}^{N} w(n), \tag{78}$$

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N} \left(w(n) - \mu\right)^2. \tag{79}$$

The arrival time is now treated as a random variable X that consists of number of small random variables $x(1), x(2), x(3), ..., x(N)\}$ where N is the total number of calls and $x(n)$ is the n^{th} call arrival time, is normal random variable which has probability density function (pdf) given by Eq. 80.

$$f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}. \tag{80}$$

Hence the probability of receiving a call from caller k at time x is given by Eq. 81, where μ_k and σ_k^2 are the corresponding mean and variance of call arrival time of caller k.

$$\Pr\{X_k = x\} = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-(x-\mu_k)^2/2\sigma_k^2}, \tag{81}$$

To check my hypothesis, I randomly select 100 callers from my dataset and perform the chi-square goodness-of-fit test (or χ^2-test) (146) (for testing the validity of the assumed distribution for a random phenomenon). I find that 30 callers have normal distribution at significant level $\alpha = 0.1$. Therefore, these 30 callers are considered as single-peak callers and the other 70 callers who do not pass the χ^2-test then belong to another group of callers which will be described in the next section.

As an example, in Fig. 9.1 the histogram of the call arrival time on time-of-the-day scales of a single-peak caller and fitted normal distribution are illustrated where I shift my window of observation to begin at 5AM and end at 4:59AM such that the entire calling pattern is captured in the middle. In fact, I find that it is a proper window of observation for the majority of the callers in my datasets.

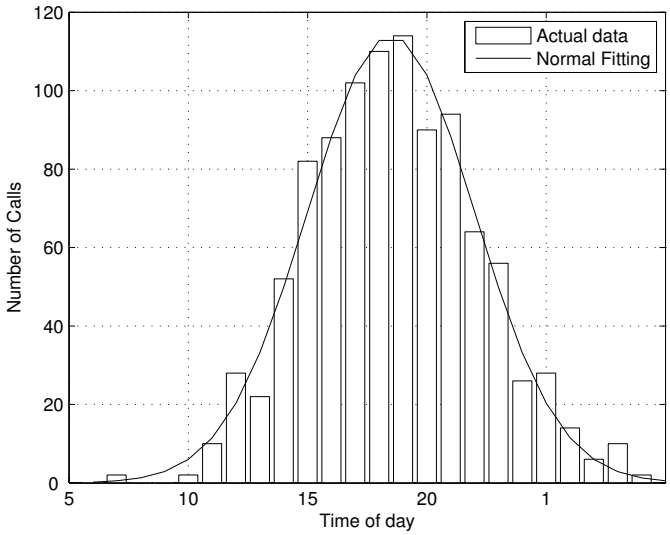


FIGURE 9.1. An example of single-peak caller whose call arrival time is fitted with normal distribution.

9.2.1.2. *Multi-peak Callers.* There is another group of callers whose calling behaviors based on arrival time are more random in the sense that they tend to have more than one favorite time of calling which result in more than one peak in their arrival time histograms.

The normal distribution is obviously not suitable for this type of callers. In fact, none of the parametric probability models fit to their structures. Therefore, probability density model must be determined from the data by using nonparametric density estimation. The most popular method for density estimation is the kernel density estimation (also known as the Parzen window estimator (94)) which is given by Eq. 82.

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right). \tag{82}$$

$K(u)$ is kernel function and h is the bandwidth or smoothing parameter. The most widely used kernel is the Gaussian of zero mean and unit variance which is defined by Eq. 83.

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{83}$$

The choice of the bandwidth h is crucial. Several optimal bandwidth selection techniques have been proposed ((157), (158)). In this chapter, I use AMISE optimal bandwidth selection using the Sheather Jones Solve-the-equation plug-in method (96).

Likewise, the probability of receiving a call from caller k at time x can be computed similarly to Eq. 81 but using probability density function defined in Eq. 82.

As an example, the observed frequency of calls over nine months on time-of-day scales and fitted kernel density estimation are illustrated in Fig. 9.2.

9.2.2. Inter-arrival Time

Caller behavior can also be characterized by the inter-arrival time which is the time interval between adjacent incoming calls as it is monitored from the callee's point of view. Based on my dataset, by observing histograms of the inter-arrival time of all callers I find that they exhibit similar patterns in which the call frequency distribution is peaked at one

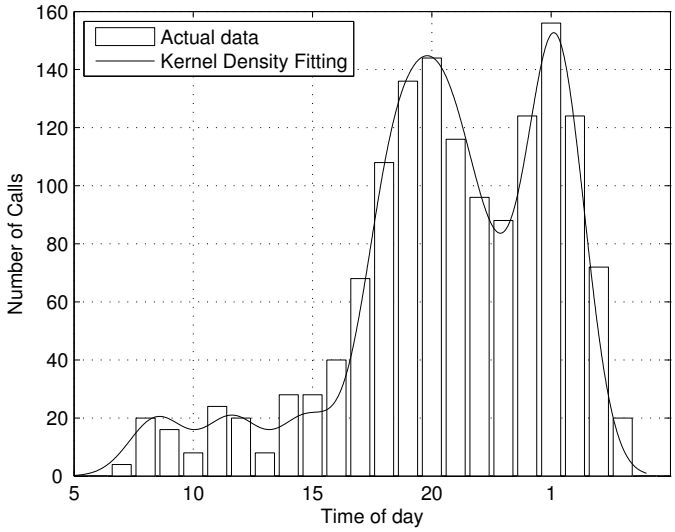


FIGURE 9.2. An example of multi-peak caller whose call arrival time is fitted with kernel density estimation.

particular point and exponentially decreases as inter-arrival time increases. Thus, I make a hypothesis that caller's inter-arrival time has an exponential distribution $exp(\gamma)$ where parameter γ is the rate at which calls are received. The parameter γ can be calculated by Eq. 84 and $E[Z]$ is the expected value of a random variable Z.

$$\gamma = \frac{1}{E[Z]}, \tag{84}$$

where inter-arrival time is a random variable Z, which consists of small random variables $\{z(1), z(2), z(3), ..., z(N)\}$, where N is the total number of calls and $z(n)$ is the inter-arrival time of the nth call, *i.e.* interval of time from $(n-1)^{th}$ to n^{th} call. The pdf is given by Eq. 85.

$$f_Z(z) = \gamma e^{-\gamma z}, \tag{85}$$

Hence the probability of inter-arrival time from caller k is z time unit can be calculated by Eq. 86 where γ_k is the corresponding parameter of inter-arrival time of caller k.

$$\Pr\{Z_k = z\} = \gamma_k e^{-\gamma_k z}. \tag{86}$$

The chi-square goodness-of-fit test is also performed here to validate my hypothesis of assuming exponential distribution for caller's inter-arrival time. The tests are done using a significant level $\alpha = 0.1$ at which all callers pass the test and therefore confirm my hypothesis.

As an example, the histogram of inter-arrival time over nine months and fitted exponential distribution are illustrated in Fig. 9.3.

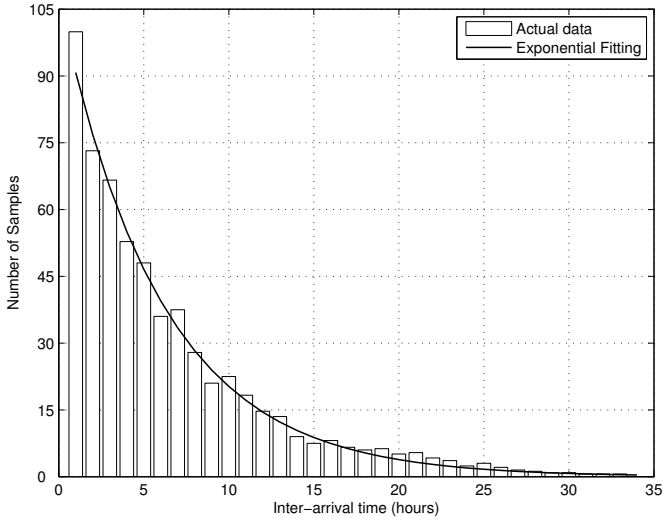


FIGURE 9.3. An example of caller's inter-arrival time is fitted with exponential distribution.

9.2.3. Talk Time

Talk time is the amount of time spent by the caller and callee during the call. From the callee's perspective, caller behavior can also be characterized by the talk time. Based on my observation of the histograms of the talk time of each caller, talk time exhibits an exponential-like pattern. Similar to the inter-arrival time pattern, the exponential distribution $exp(\lambda)$ is initially assumed for the talk time as my hypothesis where parameter λ can be calculated by Eq. 87 and $E[Y]$ is the expected value of a random variable Y.

(87) $$\lambda = \frac{1}{E[Y]}.$$

Random variable Y represents the talk time that consists of small random variables $\{y(1), y(2), y(3), ..., y(N)$ where N is the total number of calls and $y(n)$ is the talk time of the n^{th} call. The pdf is given by Eq. 88.

(88) $$f_Y(y) = \lambda e^{-\gamma y}.$$

Hence the probability of talk time with caller k is y time unit can be calculated by Eq. 89 where λ_k is the corresponding parameter of talk time of caller k.

(89) $$\Pr\{Y_k = y\} = \lambda_k e^{-\lambda_k y}.$$

Similar to my previous cases, the chi-square goodness-of-fit test is also performed using a significant level $\alpha = 0.1$ at which all trials pass the test and therefore confirm my observation and hypothesis for talk time.

An example of a histogram of talk time over nine months of a sample caller who is randomly selected from my datasets and fitted exponential distribution is illustrated in Fig. 9.4.

9.3. Adequacy of Historical Data

The caller behavior based on arrival time, inter-arrival time, and talk time have been characterized in forms of probability models in the previous section. Generally, a probability model is used to predict or estimate the future observation which is conditioned by a knowledge of the historical data. *The question is how much historical data is adequate?* This section attempts to answer this question.

In my case, the historical data is a collection of call logs which is a time series (a collection of observations made sequentially through time (219)). Unfortunately, the call logs are

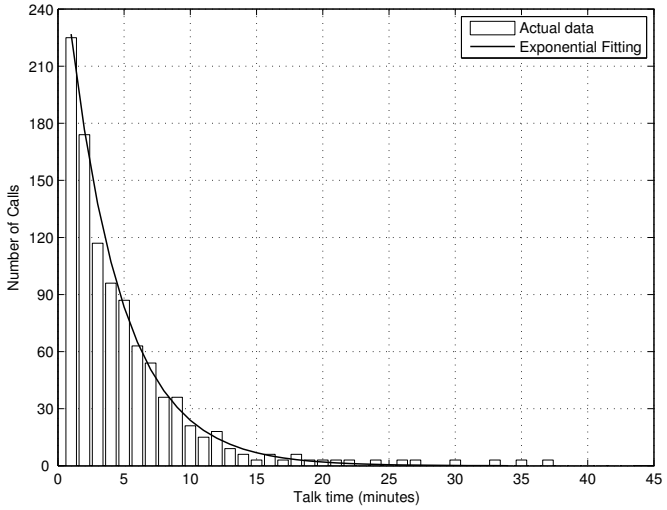


FIGURE 9.4. An example of caller's talk time is fitted with exponential distribution.

not deterministic (or can be predicted exactly) but stochastic in that future is only partly determined by past values, so that the exact predictions of future values are not quite possible and hence have a probability distribution.

The previous section shows that a single-peak caller can be characterized by a normal distribution model $N(\mu, \sigma^2)$ which is characterized be the mean μ and variance σ^2. In attempt to find out how much historical data is actually needed or adequate, I monitor the values of the mean and variance of arrival time for all single-peak callers as more historical data (increased by day) are taken into computations. *We observe the convergence of means and variances.* As an example, Fig. 9.5 shows the convergence of mean and variance of arrival time of a single-peak caller as number of days towards the past increases.

It can be observed that the values of mean and variance converge to nearly constant after taking approximately the last 30 days of historical data. This means that the mean and variance of entire historical data are approximately the same as the mean and variance of the last 30 days of data. Since a single-peak caller is characterized by a normal distribution which depends on mean and variance, it implies that the last 30 days of data is adequate to capture the behavior of the single-peak caller. It is evident in Fig. 9.6 that the pdf from taking entire historical data and taking only last 30 days are similar.

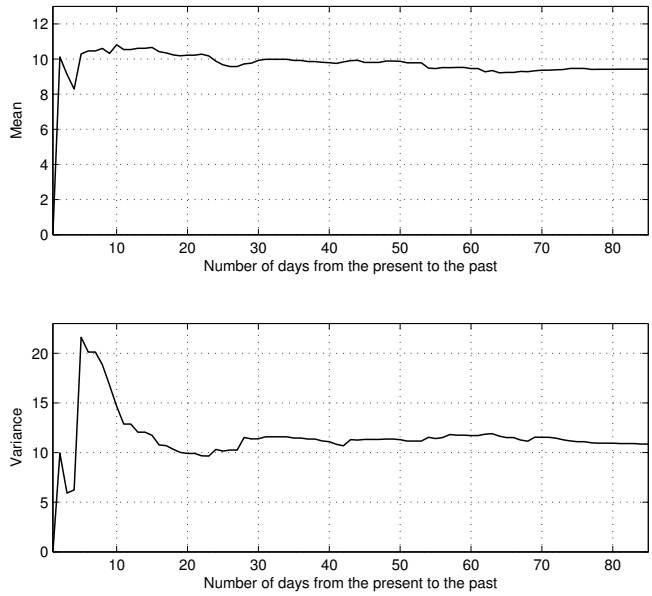


FIGURE 9.5. An example of observed convergence of mean and variance of arrival time of a single-peak caller.

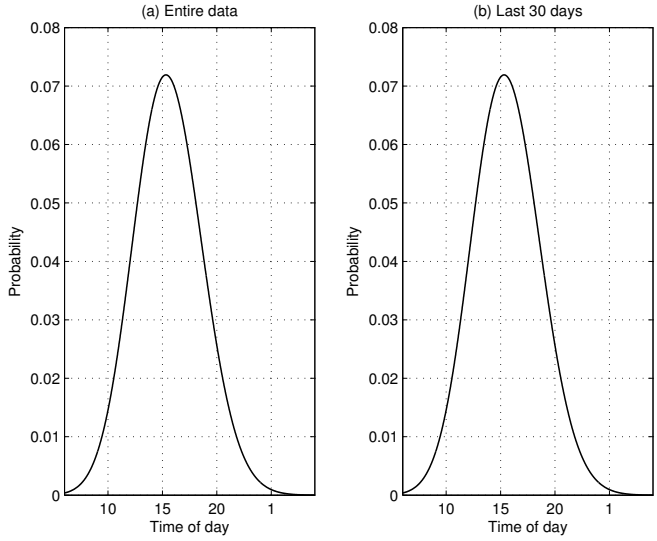


FIGURE 9.6. A comparison of pdf from (a) taking entire historical data and (b) taking only last 30 days of data.

The previous section also shows that the inter-arrival and talk time have exponential distribution $exp(m)$, which depends only on the mean m. Therefore I examine the values of

mean of inter-arrival and talk time as more historical data increases for all callers. However, I find that the convergence time is not observed.

A knowledge of mean and variance might not provide a pattern for a multi-peak caller due to the characteristics of the nonparametric density estimation. However, I believe that it captures physical behavior of a caller. In fact, the convergence of values of mean and variance of call arrival time of multi-peak callers is also observed. Figure 9.7 shows an example of a multi-peak caller whose mean and variance converge as the number of days towards the past increases. It can also be observed that the convergence time is approximately 60 days for this multi-peak caller.

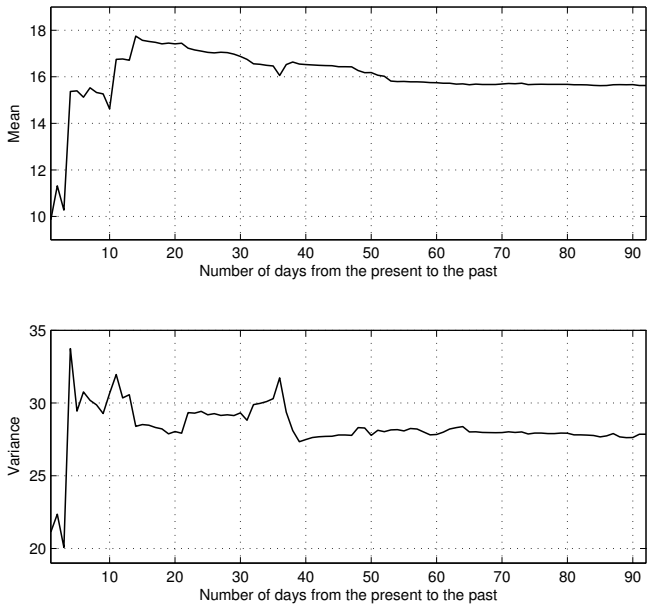


FIGURE 9.7. An example of observed convergence of mean and variance of arrival time of a multi-peak caller.

Figure 9.8 shows the pdf from taking entire historical data and taking the last 60 days of a multi-peak caller whose values of mean and variance are shown in Fig. 9.7. From Fig. 9.8, it appears that both pdf are slightly different in shape even though the mean and variance are nearly the same.

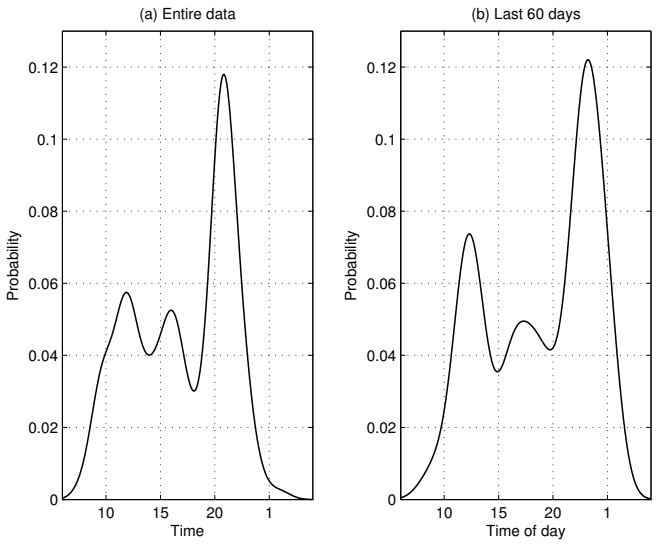


FIGURE 9.8. A comparison of pdf from (a) taking entire historical data and (b) taking only last 60 days of data.

We believe that the call logs represent human behavior associated with trends and changes of behavior over time. Considering historical data within the convergence time may provide us the recent trend of the data which can be more relevant to the future observation.

Our hypothesis is that the future behavior (pattern) of the caller based on call logs is more relevant to the pattern derived from the recent data (trend) than the pattern derived from the entire historical data (given that entire data are more than recent trend data). This hypothesis will be validated by the experiment conducted in the next section.

The crucial issue here is that of the convergence time (recent trend period) therefore I propose a simple technique for finding convergence time using a *Trace Distance* (tD).

Let us consider a sample of a converging signal shown in Fig. 9.9 where vertical axis represents amplitude and horizontal axis represents reversed time (time that runs towards the past) as similar to the plots shown Fig. 9.5 and 9.7.

A trace distance at time $k(tD_k)$ of signal s is a difference between the maximum amplitude and minimum amplitude from time k to infinity (most right-hand side of time k based on Fig. 9.9), which is given by Eq. 90.

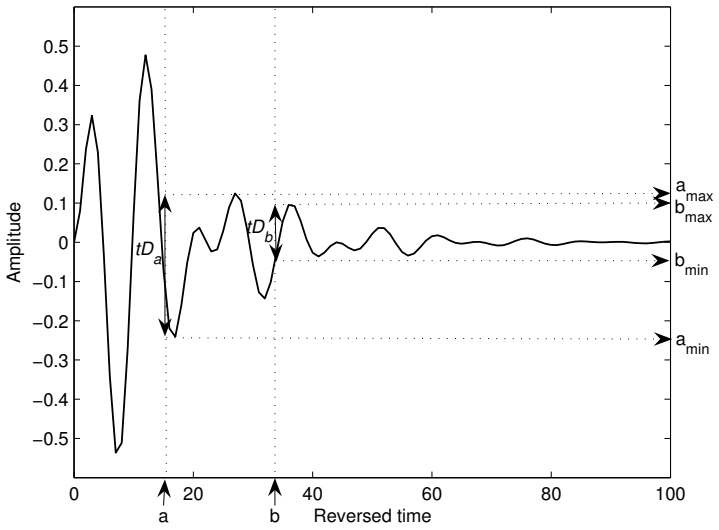


FIGURE 9.9. A converging signal which displays trace distances (tD_a and tD_b at reversed time a and b for demonstrating convergence time computation.

$$tD_k = ||k_{\max}| - |k_{\min}||, \tag{90}$$

where k_{max} and k_{min} are defined by Eq. 91 and Eq. 92, respectively.

$$k_{\max} = \max\{s(k), s(k+1), s(k+2), ..., s(\infty-1), s(\infty)\}, \tag{91}$$

$$k_{\min} = \min\{s(k), s(k+1), s(k+2), ..., s(\infty-1), s(\infty)\}. \tag{92}$$

Thus, the trace distances at time a and b shown in Fig. 9.9 can be computed as $tD_a = ||a_{max}| - |a_{min}||$ and $tD_b = ||b_{max}| - |b_{min}||$.

Therefore, the convergence time (CT) of the signal s is defined as the time that the trace distance (tD) reaches the predefined threshold (tD_{th}) as the trace distance computation starts from reversed time equals to zero to infinity which is given by Eq. 93.

$$CT_s = \{k | tD_k = tD_{th}, k \in \{0, 1, 2, ..., \infty\}\}. \tag{93}$$

For my case, the signal s can be a reversed time series of mean and variance and the variable k represents the number of days towards the past.

An experiment is conducted to find convergence time of the callers in my datasets with tD_{th} set to 1. The convergence time is computed for each caller based on the arrival time. I find an interesting result of a relationship between the caller's convergence time and his/her number of peaks. *The result shows that as the number of peaks increases, the convergence time becomes larger.* Figure 9.10 shows a plot of the average convergence time versus the number of peaks.

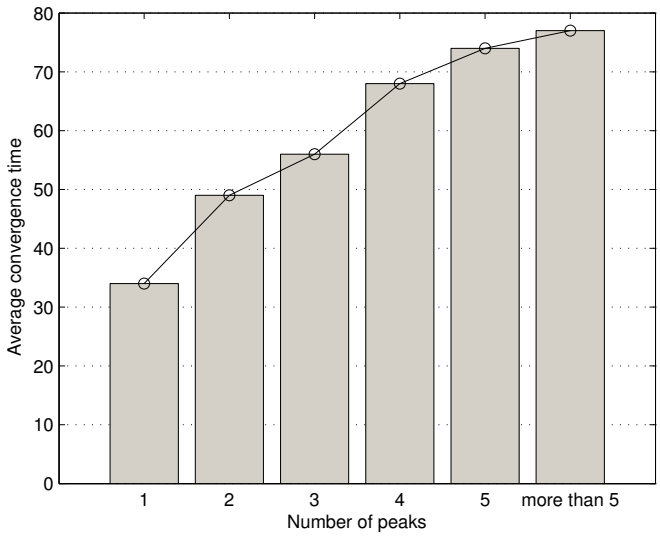


FIGURE 9.10. A plot of the number of peaks versus the average convergence time where the average convergence time becomes larger as the number of peaks increases.

We find that the result is reasonable. People who have random behaviors tend to not establish any behavioral pattern in a short period of time rather expand a recognizable structure over longer period of observation time. For example, a caller who was initially making lots of calls in the morning then started to make some calls in the evening and

he/she eventually is making calls consistently in both morning and evening hours (two-peak caller). It would take longer time to observe this caller's calling behavior than another caller who has been calling only during the morning hours (single-peak caller).

9.4. Validation

To prove my hypothesis in the previous section that the future behavior (pattern) of the caller based on call logs is more relevant to the pattern derived from the recent data (trend) than the pattern derived from the entire historical data, I conduct an experiment.

The experiment is conducted to present the comparison of the relevance or similarity in caller behavior between the future observation and entire historical observation, and the similarity in caller behavior between the future observation and recent trend observation (convergence time).

To measure the similarity in calling behaviors, three measurements are chosen; *Correlation coefficient, Hellinger distance, and Relative entropy.* In addition, performance comparison of the Call Predictor (CP) proposed in (143) is also presented to observe the change in performance as the convergence time is considered.

Correlation coefficient (146) is a number between -1 and 1 which measures the degree to which two random variables are linearly related. A correlation coefficient of 1 implies that there is perfect linear relationship between the two random variables. A correlation coefficient of -1 implies that there is inversely proportional relationship between the two random variables. A correlation coefficient of zero implies that there is no linear relationship between the variables. In many applications, a correlation coefficient is used to measure how well trends in the predicted values follow trends in past actual values or how well the predicted values from a forecast model fit with the real-life data. A correlation coefficient (r) can be computed by Eq. 94 where P and Q are random variables, which consist of small random variables $\{p(1), p(2), p(3,), ..., p(N)\}$ and $\{q(1), q(2), q(3), ..., q(N)\}$ respectively.

$$r = \frac{\sum_{n=1}^{N} \left(p(n) - \bar{P}\right)\left(q(n) - \bar{Q}\right)}{\sqrt{\sum_{n==1}^{N} \left(p(n) - \bar{P}\right)^2 \left(q(n) - \bar{Q}\right)^2}}. \tag{94}$$

Hellinger distance ((220), (221)) has value between 0 and 1 which estimates the distance between probability measures. Let P and Q be the two probability measures which are N-tuple $\{p(1), p(2), p(3,), ..., p(N)\}$ and $\{q(1), q(2), q(3), ..., q(N)\}$ respectively. P and Q satisfy $p_n \geq 0$, $\sum_n p_n = 1$, $q_n \geq 0$, and $\sum_n q_n = 1$. Hellinger distance is 0 implies that $P = Q$. Disjoint P and Q shows the maximum distance of 1. The Hellinger distance ($d_H^2(P, Q)$) between P and Q is given by Eq. 95.

$$d_H^2(P, Q) = \frac{1}{2} \sum_{n=1}^{N} \left(\sqrt{p(n)} - \sqrt{q(n)} \right)^2. \tag{95}$$

Relative entropy or Kullback Leibler distance (90) is a measure of the distance between two probability distributions. The relative entropy is a measure of the difference between assumed distribution Q and the true probability distribution P. Relative entropy is non-negative and is zero if $P = Q$. The relative entropy of Q from P is defined by 96 where $\{p(1), p(2), p(3,), ..., p(N)\}$ and $\{q(1), q(2), q(3), ..., q(N)\}$. Note that I use the convention that $0 \log(0/q) = 0$ and $p \log(p/0) = 1$. The relative entropy ($D(P||Q)$) between P and Q can be computed by Eq. 96.

$$D(P||Q) = \sum_{n=1}^{N} p(n) \log \frac{p(n)}{q(n)}. \tag{96}$$

In my case, P and Q are the N-tuple probability mass functions of the future observation and testing period respectively where the testing period can be either within the convergence time or entire historical data.

Phithakkitnukoon and Dantu (143) proposed a Call Predictor (CP) which computed receiving call probability and made the next-24-hour incoming call prediction based on caller's behavior and reciprocity. The caller's behavior was measured by the caller's call arrival time and inter-arrival time. The reciprocity was measured by the number of outgoing calls per incoming call and inter-arrival/departure time. The CP took into account the entire call historical.

In this experiment, I examine the performance of the CP with considering the convergence time of the call history and compare to the performance of the CP without considering the convergence time (or taking entire call history). The performance is measured in terms of *Error rate* which is defined as a ratio of the number of fault predictions to the total number of predictions made.

The experiment is conducted with 100 randomly selected callers including 30 single-peak callers and 70 multi-peak callers from my datasets. The most recent seven days of call logs are assumed to be future observation. The trace distance threshold tD_{th} is set to 1 to compute the convergence time (CT). The CP repeatedly computes the CT for each of the seven days prior to making call prediction.

Figure 9.11(a), 9.12(a), 9.13(a), and 9.14(a) show the comparisons of the computed correlation coefficients, Hellinger distance, relative entropy, and error rate of the CP respectively of all 100 callers between taking entire historical data (represented with an asterisk (*)) and taking data within the convergence time (represented with a circle (o)) where the first 30 callers are single-peak callers and at rest are multi-peak callers (31-100).

Figure 9.11(b), 9.12(b), 9.13(b), and 9.14(b) show the changes in the values of correlation coefficient, Hellinger distance, relative entropy, and error rate of the CP respectively as the convergence time is considered.

It can be observed that the value of correlation coefficient increases as the convergence time is considered for all 100 callers which tells us that the recent caller behavior or calling pattern is more relevant (correlated) to the future calling pattern than the pattern observed from entire call history.

The values of Hellinger distance, relative entropy, and error rate of the CP decrease as the convergence time is considered, which also confirms that the recent calling pattern is more relevant to the future pattern.

The experimental result is summarized in the Table 9.1 which lists the numerical average values of the correlation coefficient, Hellinger distance, relative entropy, and error rate of the CP when the entire data is considered, as well as when the data within the convergence time

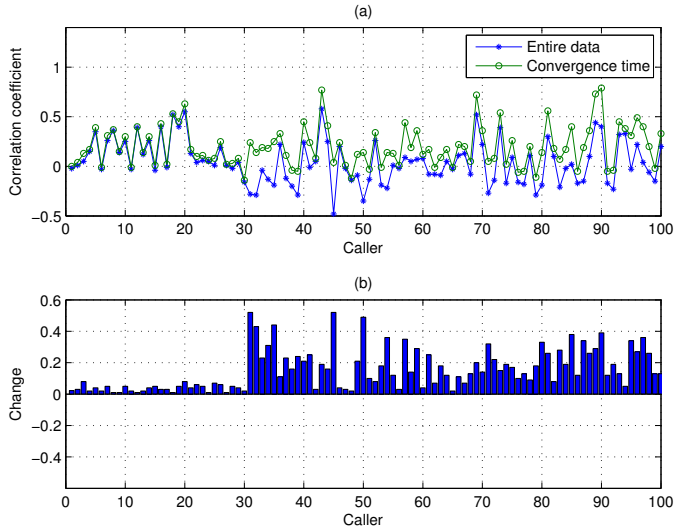


FIGURE 9.11. (a) Comparison of correlation coefficients and (b) its corresponding change from taking entire historical data to taking data within convergence time of each caller.

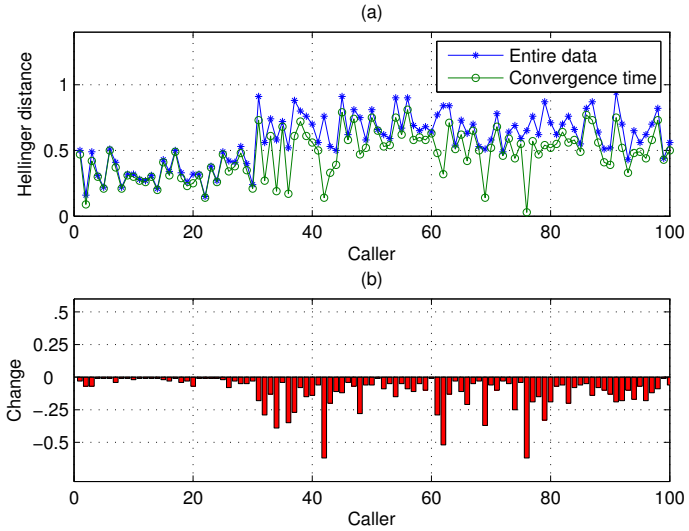


FIGURE 9.12. (a) Comparison of Hellinger distances and (b) its corresponding change from taking entire historical data to taking data within convergence time of each caller.

is considered, and their average changes for categorized single-peak callers and multi-peak

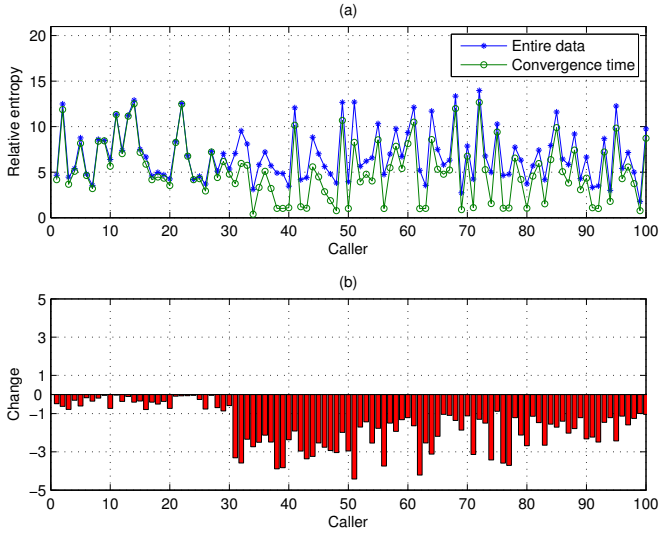


FIGURE 9.13. (a) Comparison of relative entropy and (b) its corresponding change from taking entire historical data to taking data within convergence time of each caller.

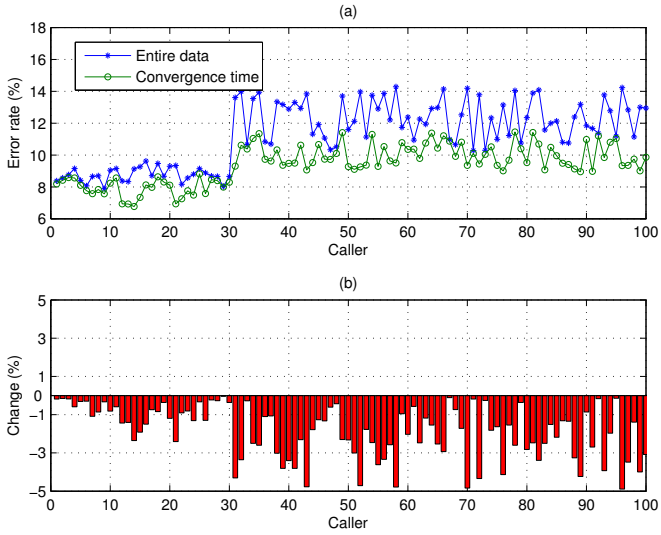


FIGURE 9.14. (a) Comparison of error rate of the call predictor and (b) its corresponding change from taking entire historical data to taking data within convergence time of each caller.

callers. Since the single-peak callers have normal distribution, the change in the similarity measures are relatively low compared to the multi-peak callers.

These experimental results shows that the data within convergence time is adequate to construct a predictive model and in fact it composes a recent pattern which is more similar or relevant to the future pattern than considering pattern composed by the entire historical data.

TABLE 9.1. The average of correlation coefficients (r), Hellinger distance (d_H^2), relative entropy (D), and error rate (Err) of taking entire historical data comparing to taking only data within the convergence time and its average change (increase(+) or decrease(-))

Callers	**Average Measures of Taking Entire Data**				**Average Measures of Taking Data within Convergence Time**				**Average Change**			
	r	d_H^2	D	$Err(\%)$	r	d_H^2	D	$Err(\%)$	r	d_H^2	D	$Err(\%)$
(1-30) Single-peak	0.1476	0.6573	6.9377	8.751	0.1837	0.63	6.547	7.9153	+0.0361	-0.0273	-0.3907	-0.8357
(31-100) Multi-peak	0.0007	0.6791	6.8423	12.3672	0.2043	0.5329	4.6256	10.0429	+0.2036	-0.1462	-2.2167	-2.3243

9.5. Conclusion

In this chapter, I propose a technique to find the adequacy of historical call logs in order to capture the caller behavior (pattern). Firstly, the statistical analysis of real-life datasets to characterize caller behavior is carried out. I classify callers into two groups namely single-hop callers and multi-hop callers based on the distribution of the arrival time of the calls. I have verified the normal distribution for single-hop callers and estimated the distribution for multi-hop callers using kernel density estimator. I have also verified exponential distribution for inter-arrival time and talk time.

Since the caller behavior can be characterized by probability models which are used to predict or estimate the future behavior conditioned by a knowledge of the historical data, the question is how much historical data is adequate.

CHAPTER 10

A RECENT-PATTERN BIASED DIMENSION-REDUCTION FRAMEWORK FOR TIME SERIES DATA

10.1. Introduction

Time series is a sequence of time-stamped data points, which account for a large proportion of the data stored in today's scientific and financial databases. Examples of a time series include stock price, exchange rate, temperature, humidity, power consumption, and event logs. Time series are typically large and of high dimensionality. To improve the efficiency of computation and indexing, dimension-reduction techniques are needed for high-dimensional data. Among the most widely used techniques are PCA (also known as SVD), DFT, and DWT. Other recently proposed techniques are Landmarks (222), PAA (223), APCS (224), PIP (225), Major minima and maxima (226), and Magnitude and shape approximation (227). These techniques were developed to reduce the dimensionality of the time series by considering every part of a time series equally. In many applications such as the stock market, however, recent data are much more interesting and significant than old data, "recent-biased analysis" (the term originally coined by Zhao and Zhang (228)) thus emerges. The recently proposed techniques include Tilt time frame (229), Logarithmic tilted-time window (230), Pyramidal time frame (231), SWAT (232), Equi-segmented scheme (228), and Vari-segmented scheme (228).

Generally, a time series reflects the behavior of the data points (monitored event), which tends to repeat periodically and creates a pattern that alters over time due to countless factors. Hence the data that contains the recent pattern are more significant than just recent data and even more significant than older data. This change of behavioral pattern provides the key to my proposed framework in dimension reduction. Since the pattern changes over time, the most recent pattern is more significant than older ones. In this chapter, I introduce

a new recent-pattern biased dimension-reduction framework that gives more significance to the recent-pattern data (not just recent data) by keeping it with finer resolution, while older data is kept at coarser resolution. With my framework, the traditional dimension-reduction techniques such as SVD, DFT, DWT, Landmarks, PAA, APCS, PIP, Major minima and maxima, and Magnitude and shape approximation can be used. As many applications (233) (234) (235) (236) generate data streams (e.g., IP traffic streams, click streams, financial transactions, text streams at application level, sensor streams), I also show that it is simple to handle a dynamic data stream with my framework.

Idistinguish this work from other previously proposed recent-biased dimension-reduction techniques by the following contributions:

(1) I develope a new framework for dimension reduction by keeping more detail on data that contains the most recent pattern and less detail on older data.

(2) Within this framework, I also propose Hellinger distance-based algorithms for recent periodicity detection and recent-pattern interval detection.

10.2. Background and Related Work

This section reviews traditional dimension reduction methods and briefly describes related work in the recent-biased dimension reduction.

10.2.1. Dimension Reduction

With advances in data collection and storage capabilities, the amount of the data that needs to be processed is increasing rapidly. To improve the efficiency of computation and indexing when dealing with high-dimensional time series or large datasets, dimension reduction is needed. The classical methods include PCA, DFT, and DWT:

PCA (Principal Component Analysis) (237) is a popular linear dimension-reduction technique that minimizes the mean square error of approximating the data. It is also known as the singular value decomposition (SVD), the Karhunen-Loeve transform, the Hotelling transform, and the empirical orthogonal function (EOF) method. PCA is an eigenvector-based

multivariate analysis that seeks to reduce dimension of the data by transforming the original data to a few orthogonal linear combinations (the PCs) with the largest variance.

DFT (Discrete Fourier Transform) has been used for the dimensionality reduction (238) (239) (240) (241) by transforming the original time series (of length N) without changing information content to the frequency domain representation and retaining a few low-frequency coefficients (p, where $p < N$) to reconstruct the series. Fast Fourier transform (FFT) is a popular algorithm to compute DFT with time complexity of $O(N \log N)$.

DWT (Discrete Wavelet Transform) is similar to DFT except that it transforms the time series into time/frequency domain and its basis function is not a sinusoid but generated by the mother wavelet. Haar wavelet (242) is one of the most widely used class of wavelets with time complexity of $O(N)$.

Other proposed techniques include Landmarks, PAA, APCS, PIP, Major minima and maxima, and Magnitude and shape approximation:

Landmark model has been proposed by Perng et al. (222) to reduce dimensionality of time series. The idea is to reduce the time series to the points (time, events) of greatest importance, namely "landmarks". The n-th order landmark of a curve is defined for the point whose n-th order derivative is zero. Hence local maxima and minima are first-order landmarks, and inflection points are second-order landmarks. Compared with DFT and DWT, landmark model retains all peaks and bottoms that normally filtered out by both DFT and DWT.

Keogh et al. (223) have proposed PAA (Piecewise Aggregate Approximation) as a dimension-reduction technique that reduces the time series to the mean values of the segmented equi-length sections. PAA has an advantage over DWT as it is independent of the length of the time series (DWT is only defined for sequences whose length is an integral power of two).

The concept of PAA has later been modified to improve the quality of approximation by Chakrabarti et al. (224) who propose APCS (Adaptive Piecewise Constant Approximation)

that allows segments to have arbitrary lengths. Hence two numbers are recorded for each segment; mean value and length.

PIP (Perpetually Important Points) has been introduced by *Fu et al.* (225) to reduce dimensionality of the time series by replacing the time series with PIPs, which are defined as highly fluctuated points.

Fink et al. (226) have proposed a technique for fast compression and indexing of time series by keeping major minima and maxima and discarding other data points. The indexing is based on the notion of major inclines.

Ogras and Ferhatosmanoglu (227) have introduced a dimension-reduction technique that partitions the high dimensional vector space into orthogonal subspaces by taking into account both magnitude and shape information of the original vectors.

10.2.2. Recent-biased Dimension Reduction

Besides the global dimension reduction, in many applications such as stock prices, recent data are much more interesting and significant than old data. Thus, the dimension-reduction techniques that emphasize more on the recent data by keeping recent data with fine resolution and old data with coarse resolution have been proposed such as Tilt time frame, Logarithmic tilted-time window, Pyramidal time frame, SWAT, Equi-segmented scheme, and Vari-segmented scheme:

Tilt time frame has been introduced by Chen et al. (229) to minimize the amount of data to be kept in the memory or stored on the disks. In the tilt time frame, time is registered at different levels of granularity. The most recent time is registered at the finest granularity, while the more distant time is registered at coarser granularity. The level of coarseness depends on the application requirements.

Similar to the tilt time frame concept but with more space-efficient, Giannella et al. have proposed the logarithmic tilted-time window model (230) that partitions the time series into growing tilted-time window frames at an exponential rate of two *e.g.*, 2, 4, 8, 16, and so forth.

The concept of the pyramidal time frame has been introduced by Aggarwal et al. in (231). With this technique, data are stored at different levels of granularity depending upon the recency, which follows a pyramidal pattern.

SWAT (Stream Summarization using Wavelet-based Approximation Tree) (232) has been proposed by Bulut and Singh to process queries over data streams that are biased towards the more recent values. SWAT is a Haar wavelet-based scheme that keeps only a single coefficient at each level.

Zhao and Zhang have proposed the equi-segmented scheme and the vari-segmented scheme in (228). The idea of the equi-segmented scheme is to divide the time series into equi-length segments and apply a dimension reduction technique to each segment, and keep more coefficients for recent data while fewer coefficients are kept for old data. Number of coefficients to be kept for each segment is set to $\lfloor N/2^i \rfloor$ where N is the length of the time series and segment gets older with the increase of i. For the vari-segmented scheme, the time series is divided into variable length segments with larger segments for older data and smaller segments for more recent data (the length of segment i is set to 2^i). The same number of coefficients are then kept for all segments after applying a dimension reduction technique to each segment.

10.3. Recent-Pattern Biased Dimension-Reduction Framework

Time series data analysis comprises methods that attempt either to understand the context of the data points or to make forecasts based on observations (data points). In many applications, recent data receive more attention than old ones. Generally, a time series reflects the behavior of the data points (monitored event), which tends to repeat periodically and creates a pattern that alters over time due to countless factors. Hence the data that contains recent pattern are more significant than just recent data and even more significant than older data. Typically, future behavior is more relevant to the recent behavior than older ones. My main goal in this work is to reduce dimensionality of a time series with the basic idea of keeping data that contains recent pattern with high precision and older data with low precision. Since the change in behavior over time creates changes in the pattern and

the periodicity rate, I thus need to detect the most recent periodicity rate, which will lead to identifying the most recent pattern. Hence a dimension reduction technique can then be applied. This section presents my novel framework for dimension reduction for time series data, which includes new algorithms for recent periodicity detection, recent-pattern interval detection, and dimension reduction.

10.3.1. Recent Periodicity Detection

Unlike other periodicity detection techniques ((243), (244), (245), (246), (247), and (248)) that attempt to detect the global periodicity rates, my focus here is to find the "most recent" periodicity rate of time series data. Let X denote a time series with N time-stamped data points, and x_i be the value of the data at time-stamp i. The time series X can be represented as $X = x_0, x_1, x_2, ..., x_N$, where x_0 is the value of the most recent data point and x_N is the value of the oldest data point. Let $\Phi(k)$ denote the recent-pattern periodicity likelihood (given by Eq. 97) that measures the likelihood of selected recent time segment (k) being the recent period of the time series, given that the time series X can be sliced into equal-length segments $X_0^k, X_1^k, X_2^k, ..., X_{\lfloor N/k \rfloor - 1}^k$, each of length k, where $X_i^k = x_{ik}, x_{ik+1}, x_{ik+2}, ..., x_{ik+k-1}$.

$$\Phi(k) = \frac{\sum_{i=1}^{\lfloor N/k \rfloor - 1}(1 - d_H^2(\hat{X}_0^k, \hat{X}_i^k))}{\lfloor N/k \rfloor - 1}, \tag{97}$$

where $d_H^2(A, B)$ is Hellinger distance (249), which is widely used for estimating a distance (difference) between two probability measures (*e.g.*, probability density functions (pdf), probability mass functions (pmf)). Hellinger distance between two probability measures A and B can be computed by Eq. 98. A and B are M-tuple $\{a_1, a_2, a_3, ..., a_M\}$ and $\{b_1, b_2, b_3, ..., b_M\}$ respectively, and satisfy $a_m \geq 0, \sum_m a_m = 1, b_m \geq 0$, and $\sum_m b_m = 1$. Hellinger distance of 0 implies that $A = B$ whereas disjoint A and B yields the maximum distance of 1.

$$d_H^2(A, B) = \frac{1}{2} \sum_{m=1}^{M} (\sqrt{a_m} - \sqrt{b_m})^2. \tag{98}$$

In my case, $\hat{X}_0^k$ and $\hat{X}_i^k$ are X_0^k and X_i^k after normalization, respectively, such that they satisfy the above conditions. Thus, $\Phi(k)$ has the values in the range [0, 1] as 0 and 1 imply the lowest and the highest recent-pattern periodicity likelihood, respectively.

DEFINITION 10.1. If a time series X of length N can be sliced into equal-length segments $X_0^p, X_1^p, X_2^p, ..., X_{\lfloor N/p \rfloor - 1}^p$, each of length p, where $X_i^p = x_{ip}, x_{ip+1}, x_{ip+2}, ..., x_{ip+p-1}$, and $p = \arg\max_k \Phi(k)$, then p is said to be the recent periodicity rate of X.

The basic idea of this algorithm is to find the time segment (k) that has the maximum $\Phi(k)$, where $k = 2, 3, ..., \lfloor N/2 \rfloor$. If there is a tie, smaller k is chosen to favor shorter periodicity rates, which are more accurate than longer ones since they are more informative (245). The detailed algorithm is given in Fig. 10.1. Note that $\Phi(1) = 1$ since $d_H^2(\hat{X}_0^1, \hat{X}_i^1) = 0$, hence k begins at 2.

p = PERIODICITY(X)

Input: Time series (X) of length N

Output: Recent periodicity rate (p)

1. FOR $k = 2$ to $\lfloor N/2 \rfloor$
2. Compute $\Phi(k)$;
3. END FOR
4. $p = k$ that maximizes $\Phi(k)$;
5. IF $|k| > 1$
6. $p = min(k)$;
7. END IF
8. Return p as the recent periodicity rate;

FIGURE 10.1. Algorithm for the recent periodicity detection.

10.3.2. Recent-Pattern Interval Detection

After obtaining the recent periodicity rate p, my next step towards dimension reduction for a time series X is to detect the time interval that contains the most recent pattern. This interval is a multiple of p. I base my detection on the *shape* of the pattern and the *amplitude* of the pattern.

For the detection based on the shape of the pattern, I construct three Hellinger distance-based matrices to measure the differences within the time series as follows:

- $D_1^i = [d_1(1), d_1(2), ..., d_1(i)]$ is the matrix whose elements are Hellinger distances between the pattern derived from the X_0^p to X_{j-1}^p ($\bar{X}_{0\to j-1}^p$), which can be simply computed as a mean time series over time segments 0 to $j-1$ given by Eq. 100, and the pattern captured within the time segment j (X_j^p) as follows:

$$d_1(j) = d_H^2(\hat{\bar{X}}_{0\to j-1}^p, \hat{X}_j^p), \tag{99}$$

where

$$\bar{X}_{0\to j-1}^p = \frac{1}{j}\sum_{n=0}^{j-1} x_{np}, \frac{1}{j}\sum_{n=0}^{j-1} x_{np+1}, ..., \frac{1}{j}\sum_{n=0}^{j-1} x_{np+p-1}. \tag{100}$$

Again, the hat on top of the variable indicates the normalized version of the variable.

- $D_2^i = [d_2(1), d_2(2), ..., d_2(i)]$ is the matrix whose elements are Hellinger distance between the most recent pattern captured in the first time segment (X_0^p) and the pattern occupied within the time segment j (X_j^p) as follows:

$$d_2(j) = d_H^2(\hat{X}_0^p, \hat{X}_j^p). \tag{101}$$

- $D_3^i = [d_3(1), d_3(2), ..., d_3(i)]$ is the matrix whose elements are Hellinger distance between the adjacent time segments as follows:

$$d_3(j) = d_H^2(\hat{X}_{j-1}^p, \hat{X}_j^p). \tag{102}$$

These three matrices provide the information on how much the behavior of the time series changes across all time segments. The matrix D_1^i collects the degree of difference that X_j^p introduces to the recent segment(s) of the time series up to $j = i$, where $j = 1, 2, 3, ..., \lfloor N/p \rfloor - 1$. The matrix D_2^i records the amount of difference that the pattern occupied in the time segmentX_j^p makes to the most recent pattern captured in the first time segmentX_0^p up to $j = i$. The matrix D_3^i keeps track of the differences between the patterns captured in the adjacent time segmentsX_{j-1}^p and X_j^pup to $j = i$.

To identify the recent-pattern interval based on the shape of the pattern, the basic idea here is to detect the first change of the pattern that occurs in the time series as I search across all the time segments X_j^p in an increasing order of j starting from $j = 1$ to $\lfloor N/p \rfloor - 1$.

Several changes might have been detected as I search through entire time series, however my focus is to detect the most recent pattern. Therefore, if the first change is detected, the search is over. The change of pattern can be observed from the significant changes of these three matrices. The significant change is defined as follows.

DEFINITION 10.2. If $\mu_{D_k^i}$ and $\sigma_{D_k^i}$ is the mean and the standard deviation of D_k^i and $\mu_{D_k^i} + 2\sigma_{D_k^i} \leq d_k(i+1)$, then X_{i+1}^p is said to make the significant change based on its shape.

y = SIG_CHANGE(D_k^i,$d_k(i+1)$)
Input: Distance matrix (D_k^i) and the corresponding distance element$d_k(i+1)$.
Output: Binary output (y) of 1 implies that there is a significant change made by X_{i+1}^p and 0 implies otherwise.
1. IF $\mu_{D_k^i} + 2\sigma_{D_k^i} \leq d_k(i+1)$
2. $y = 1$;
3. ELSE
4. $y = 0$;
5. END IF

FIGURE 10.2. Algorithm for detecting the significant change.

With the detected significant changes in these distance matrices, the recent-pattern interval based on the shape of the pattern can be defined as follows. The detailed algorithm is given in Fig. 10.3.

DEFINITION 10.3. If X_{i+1}^p introduces a significant change to at least two out of three matrices (D_1^i,D_2^i, and D_3^i), then the recent-pattern interval based on the shape (r_{shape}) is said to be ip time units.

For this shape-based recent-pattern interval detection, the Hellinger distances are computed by taking the normalized version of the patterns in the time segments. Since normalization rescales the amplitude of the patterns, the patterns with similar shapes but significantly different amplitudes will not be detected (see an example illustrated in Fig. 10.4).

r_{shape} = SHAPE_RPI($D_1^{\lfloor N/p \rfloor - 1}$,$D_2^{\lfloor N/p \rfloor - 1}$,$D_3^{\lfloor N/p \rfloor - 1}$)

Input: Three distance matrices ($D_1^{\lfloor N/p \rfloor - 1}$,$D_2^{\lfloor N/p \rfloor - 1}$, $D_3^{\lfloor N/p \rfloor - 1}$).

Output: Shape-based recent-pattern interval (r_{shape}).

1. Initialize r_{shape} to N
2. FOR $i = 2$ to $\lfloor N/p \rfloor - 1$
3. IF SIG_CHANGE(D_1^i,$d_1(i+1)$) + SIG_CHANGE(D_2^i,$d_2(i+1)$) + SIG_CHANGE(D_3^i,$d_3(i+1)$) ≥ 2
4. $r_{shape} = ip$;
5. EXIT FOR LOOP
6. END IF
7. END FOR
8. Return r_{shape} as the recent-pattern interval based on the shape;

FIGURE 10.3. Algorithm for detecting the recent-pattern interval based on the shape of the pattern.

To handle this shortcoming, I propose an algorithm to detect the recent-pattern interval based on the amplitude of the pattern. The basic idea is to detect the significant change in the amplitude across all time segments. To achieve this goal, let $A^i = [a(1), a(2), ..., a(i)]$ denote a matrix whose elements are mean amplitudes of the patterns of each time segment up to time segment i, which can be easily computed by Eq. 103.

$$a(k) = \frac{1}{p} \sum_{n=0}^{p-1} x_{(k-1)p+n}. \tag{103}$$

Similar to the previous case of distance matrices, the significant change in this amplitude matrix can be defined as follows.

DEFINITION 10.4. If μ_{A^i} and σ_{A^i} is the mean and the standard deviation of A^i and $\mu_{A^i} + 2\sigma_{A^i} \leq a(i+1)$, then X_{i+1}^p is said to make the significant change based on its amplitude.

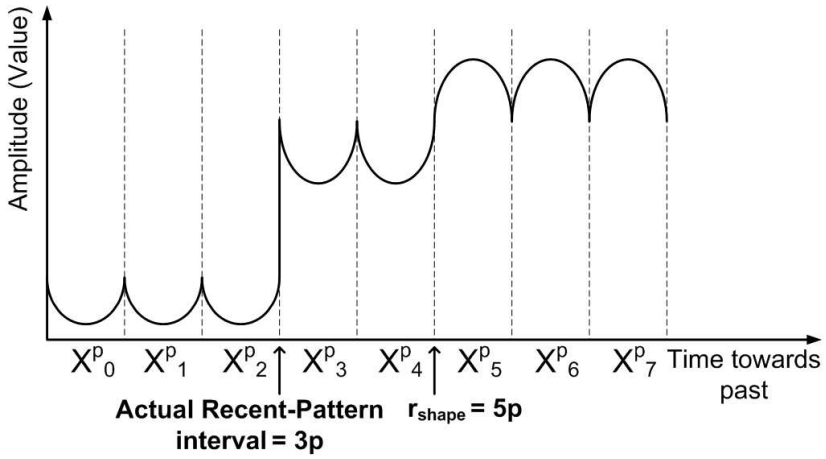


FIGURE 10.4. An example of misdetection for the recent-pattern interval based on the shape of the pattern. SHAPE_RPI(algorithm given in Fig. 10.3) would detect the change of the pattern at the 5^{th} time segment (X^p_5) whereas the actual significant change takes place at the 3^{rd} time segment (X^p_3).

Likewise, with the detected significant change in the amplitude matrix, the recent-pattern interval based on the amplitude of the pattern can be defined as follows. The detailed algorithm is given in Fig. 10.5.

DEFINITION 10.5. If X^p_{i+1} makes a significant change in the matrix (A^i), then the recent-pattern interval based on the amplitude (r_{amp}) is said to be ip time units.

Finally, the recent-pattern interval can be detected by considering both shape and amplitude of the pattern. Based on the above algorithms for detecting the interval of the most recent pattern based on the shape and the amplitude of the pattern, the final recent-pattern interval can be defined as follows.

DEFINITION 10.6. If r_{shape} is the recent-pattern interval based on the shape of the pattern and r_{amp} is the recent-pattern interval based on the amplitude of the pattern, then the final recent-pattern interval(R) is the lowest value among r_{shape} and r_{amp} – *i.e.*, $R = min(r_{shape}, r_{amp})$.

r_{amp} = AMP_RPI($A^{\lfloor N/p \rfloor - 1}$)

Input: The amplitude matrix ($A^{\lfloor N/p \rfloor - 1}$).

Output: Amplitude-based recent-pattern interval (r_{amp}).

1. Initialize r_{amp} to N
2. FOR $i = 2$ to $\lfloor N/p \rfloor - 1$
3. IF SIG_CHANGE(A^i,$a(i+1)$) = 1
4. $r_{amp} = ip$;
5. EXIT FOR LOOP
6. END IF
7. END FOR
8. Return r_{amp} as the recent-pattern interval based on the amplitude;

FIGURE 10.5. Algorithm for detecting the recent-pattern interval based on the amplitude of the pattern.

10.3.3. Dimension Reduction

Our main goal in this work is to reduce dimensionality of a time series. The basic idea is to keep more details for recent-pattern data, while older data kept at coarser level.

Based on the above idea, I propose a dimension-reduction scheme for time series data that applies a dimension reduction technique to each time segment and then keeps more coefficients for data that carries recent-behavior pattern and fewer coefficients for older data.

Let C_i represent the number of coefficients retained for the time segment X_i^p. Since my goal is to keep more coefficients for the recent-pattern data and fewer coefficients for older data, a sigmoid function (given by Eq. 104) is generated and centered at R time units (where the change of behavior takes place).

$$f(t) = \frac{1}{1 + \alpha^{-t/p}}. \tag{104}$$

The decay factor (α) is automatically tuned to change adaptively with the recent-pattern interval (R) by being set to $\alpha = p/R$, such that a slower decay rate is applied to a longer R and vice versa. The number of coefficients for each time segment can be computed as the

area under the sigmoid function over each time segment (given by Eq. 105), so the value of C_i is within the range $[1, p]$.

$$C_i = \left\lceil \int_{X_i^p} f(t)dt \right\rceil . \tag{105}$$

C_i decreases according to the area under the sigmoid function across each time segment as i increases, hence $C_0 \geq C_1 \geq C_2 \geq ... \geq C_{\lfloor N/p \rfloor - 1}$.

Several dimension reduction techniques can be used in my framework. Among the most widely popular techniques are DFT and DWT. For DFT, I keep the first C_i coefficients that capture the low-frequency part of the time series for each time segment (some other techniques for selecting DFT coefficients such as selecting the largest C_i coefficients to preserve the energy (250) or selecting the first largest C_i coefficients (251) can also be applied here). For DWT, the number of coefficients can be computed by Eq. 105 and rounded to the closest integer v, where $v = \left\lceil \frac{p}{2^j} \right\rceil$ and $j = \{0, 1, 2, ..., \log_2 p\}$, *i.e.,* $v \in \{p, \frac{p}{2}, \frac{p}{2^2}, \frac{p}{2^3}, ..., 1\}$. A larger v is chosen if there is a tie.

With this scheme, a time series data can be reduced by keeping the more important portion of data (recent-pattern data) with high precision and the less important data (old data) with low precision. As future behavior is generally more relevant to the recent behavior than old ones, maintaining the old data at low detail levels might as well reduces the noise of the data, which would benefit predictive modeling. This scheme is shown in Fig. 10.6, and the detailed algorithm is given in Fig. 10.7.

Note that if no significant change of pattern is found in the time series, my proposed framework will work similarly to equi-segmented scheme as my R is initially set to N (by default, see Fig. 10.3, Fig. 10.5 and Definition 10.6). Hence the entire series is treated as a recent-pattern data, *i.e.*, more coefficients are kept for recent data and fewer for older data according to (the left-hand side from the center of) the sigmoid function with decay factor $\alpha = p/R$.

It is simple to handle dynamic data streams with my framework. When new data arrive, they are kept in a new segment X_{new}^l until there are p new data points, *i.e.,* $l = p$. If

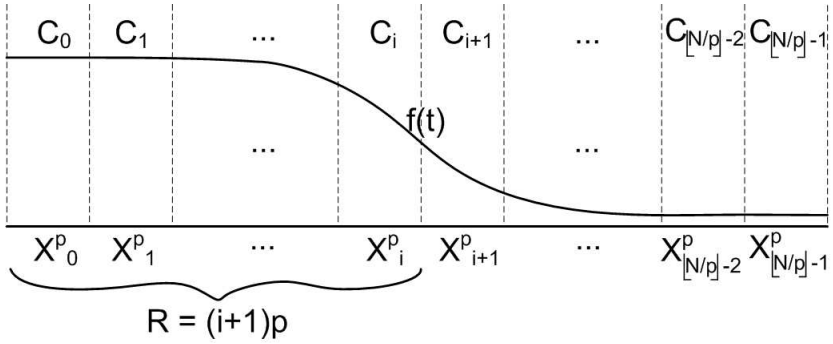


FIGURE 10.6. Recent-pattern biased dimension-reduction scheme for time series data. A time series is partitioned into equal-length segments of length p (recent periodicity rate) and more coefficients are taken for recent-pattern data and fewer coefficients are taken for older data based on the decay rate of a sigmoid function ($f(t)$). For this example, recent-pattern interval (R) is assumed to be $(i+1)p$.

$R = sp$, then only the first $s+1$ segments $(X^p_{new}, \tilde{X}^p_0, \tilde{X}^p_1, ..., \tilde{X}^p_{s-1})$ need to be processed while other segments remain unchanged. Note that $\tilde{X}^p_i$ denotes a reconstructed segment i. The new reconstructed segment $\tilde{X}^p_{new}$ will then become a new $\tilde{X}^p_0$, and other segments' order are incremented by one (*e.g.,* $\tilde{X}^p_0$ becomes $\tilde{X}^p_1$). If the original time series has N data points, then the new reconstructed time series is of length $N+p$. An example is given in Fig. 10.8.

10.4. Performance Analysis

This section contains the experimental results to show the accuracy and effectiveness of my proposed algorithms. In my experiments, I exploit synthetic data as well as real data.

The synthetic data are used to inspect the accuracy of the proposed algorithms for detecting the recent periodicity rate and the recent-pattern interval. This experiment aims to estimate the ability of proposed algorithms in detecting p and R that are artificially embedded into the synthetic data at different levels of noise in the data (measured in terms of SNR (signal-to-noise ratio) in dB). For a synthetic time series with known p and R, my algorithms compute estimated periodicity rate ($\tilde{p}$) and recent-pattern interval ($\tilde{R}$) and compare with

Z = DIMENSION_REDUCTION(X)

Input: A time series (X) of length N.

Output: A reduced time series (Z).

1. p = PERIODICITY(X);
2. Partition X into equal-length segments, each of length p;
3. Compute matrices $D_1^{\lfloor N/p \rfloor - 1}$, $D_2^{\lfloor N/p \rfloor - 1}$, $D_3^{\lfloor N/p \rfloor - 1}$, and $A^{\lfloor N/p \rfloor - 1}$;
4. r_{shape} = SHAPE_RPI($D_1^{\lfloor N/p \rfloor - 1}$,$D_2^{\lfloor N/p \rfloor - 1}$,$D_3^{\lfloor N/p \rfloor - 1}$);
5. r_{amp} = AMP_RPI($A^{\lfloor N/p \rfloor - 1}$);
6. $R = min(r_{shape}, r_{amp})$;
7. Place a sigmoid function $f(t)$ at R;
8. FOR each segment i
9. $\quad Coefs$ = apply dimension-reduction technique for segment i;
10. $\quad$ Compute C_i;
11. $\quad z_i$ = first C_i $Coefs$;
12. END FOR
13. $Z = \{z_0, z_1, z_2, ..., z_{\lfloor N/p \rfloor - 1}\}$; /* Series of selected coefficients */
14. Return Z as the reduced time series;

FIGURE 10.7. Algorithm for detecting the recent-pattern interval based on the amplitude of the pattern.

the actual p and R to see if the estimated values are matched to the actual values. I generate 100 different synthetic time series with different values of p and R. The *error rate* is then computed for each SNR level (0dB to 100dB) as the number of incorrect estimates (Miss) per total number of testing data, *i.e.* Miss/100. The results of this experiment are shown in Fig. 10.9. The error rate decreases with increasing SNR as expected. My recent periodicity detection algorithm performs with no error above 61dB while my recent-pattern interval detection algorithm performs perfectly above 64dB. Therefore, based on this experiment, my proposed algorithms are effective at SNR level above 64dB.

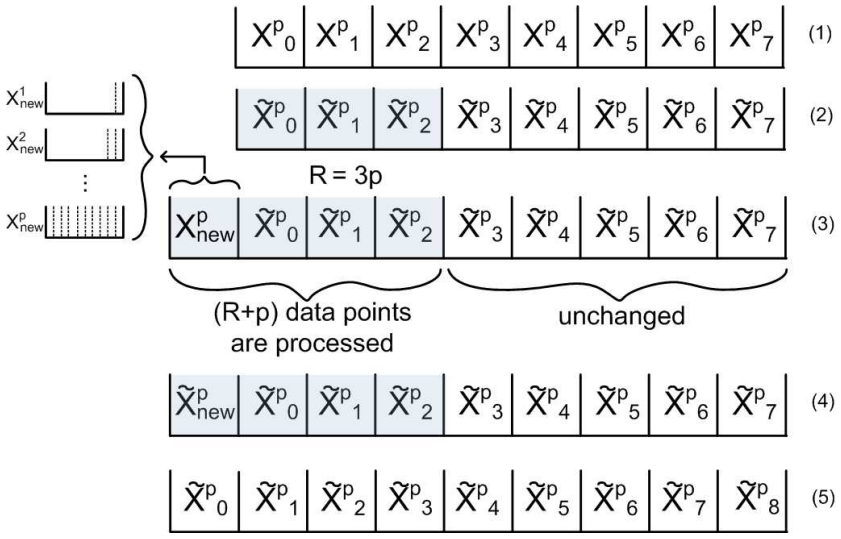


FIGURE 10.8. An example of processing a dynamic data stream. (1) Original data has $7p$ data points. (2) Suppose that $R = 3p$. (3) New data points are kept in a new segment X^l_{new} until $l = p$, then the first $R + p$ data points are processed with other data points unchanged. (4) The reconstructed time series of length $8p$. (5) The new reconstructed segment $\tilde{X}^p_{new}$ becomes a new $\tilde{X}^p_0$, and other segments' order are incremented by one.

I implement my algorithms on three real time series data. The first data contains the number of phone calls (both made and received) on time-of-the-day scales on a monthly basis over a period of six months (January 7^{th}, 2008 to July 6^{th}, 2008) of a mobile phone user (106). The second data contains a series of monthly water usage (ml/day) in London, Ontario, Canada from 1966 to 1988 (252). The third data contains Quarterly S&P 500 index values taken from 1900-1996 (253). Figure 10.10 shows a time series of a mobile phone usage with computed $p = 24$ and $R = 3p = 72$ based on my algorithms. Likewise, Fig. 10.11 shows a time series of a monthly water usage with computed $p = 12$ and $R = 2p = 24$. Similarly, Fig. 10.12 depicts a time series of quarterly S&P 500 index values during 1900-1996 with computed $p = 14$ and $R = 3p = 42$. Based on a visual inspection, one can clearly identify that the recent periodicity rates are 24, 12, and 14; and recent-pattern intervals are $3p$, $2p$,

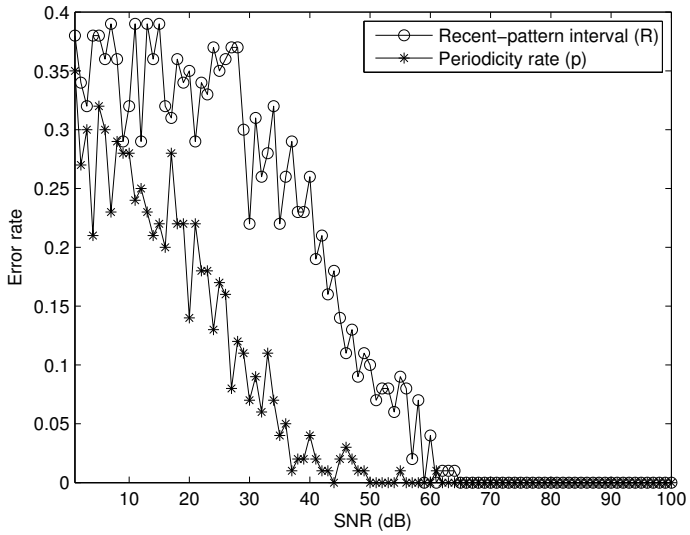


FIGURE 10.9. Experimental result of the error rate at different SNR levels of 100 synthetic time series (with known p and R).

and $3p$ for Fig. 10.10, Fig. 10.11, and Fig. 10.12, respectively, which shows the effectiveness of my algorithms.

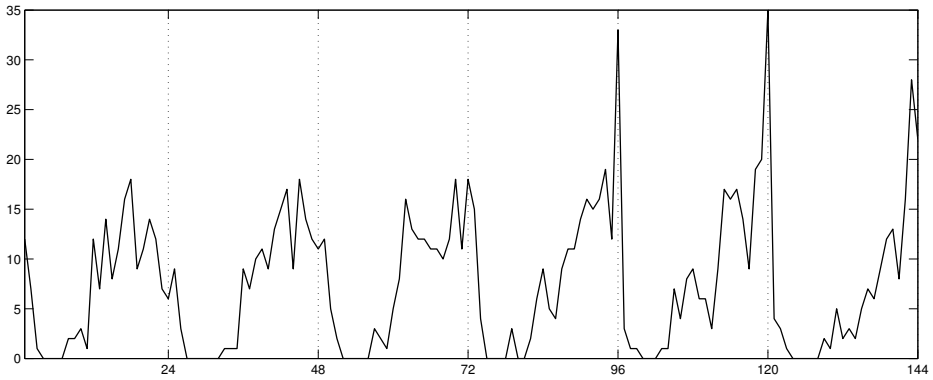


FIGURE 10.10. A monthly mobile phone usage over six months (January 7^{th}, 2008 to July 6^{th}, 2008) with detected $p = 24$ and $R = 3p = 72$.

I implement my recent-pattern biased dimension-reduction algorithm on these three real time series data. Due to the space limitation, the experimental results are only illustrated with DFT and DWT as the dimension-reduction techniques. As the results, the 144-point mobile phone data has been reduced to 75 data points using DFT, which is 48% reduction,

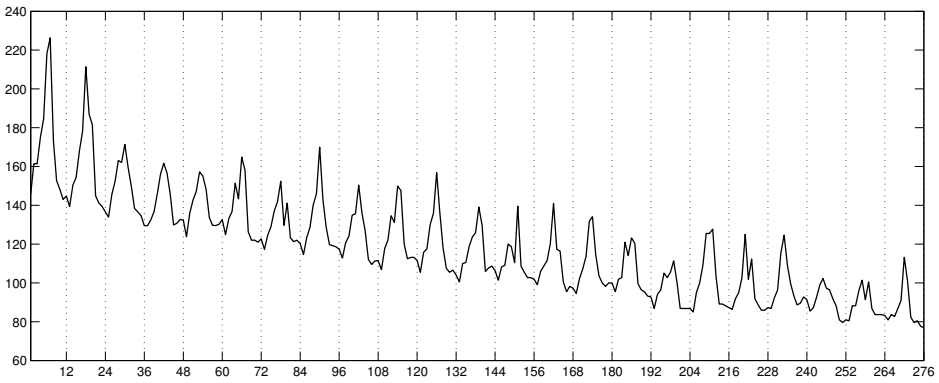


FIGURE 10.11. A monthly water usage during 1966-1988 with detected $p = 12$ and $R = 2p = 24$.

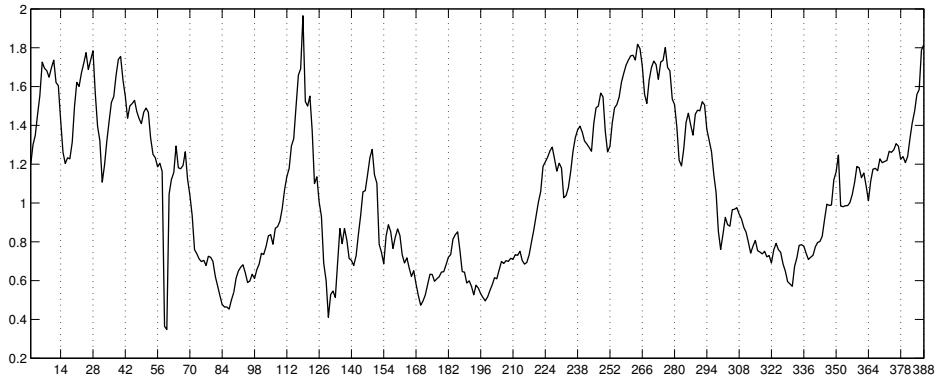
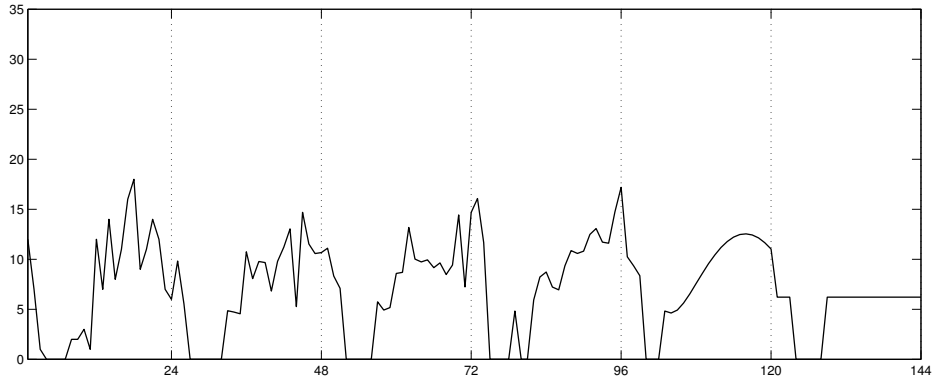


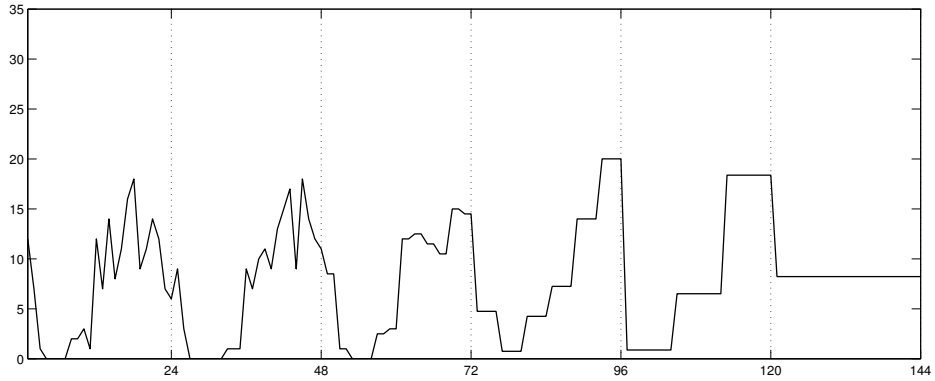
FIGURE 10.12. Quarterly S&P 500 index values taken from 1900-1996 with detected $p = 14$ and $R = 3p = 42$.

and reduced to 70 points using Haar DWT, which is 51% reduction. For the water usage data, since it has a relatively short recent-pattern interval compared to the length of its entire series thus I am able to reduce much more data. In fact, there are 276 data points of water usage data before the dimension reduction and only 46 data points are retained afterward by using DFT and 52 data points kept using DWT, which is 83% and 81% reduction, respectively. Likewise, for the S&P 500 data, I am able reduce 83% of data by keeping 66 DFT coefficients and 81% by keeping 72 DWT coefficients from the original data of length 378.

The reconstructed time series using DFT and DWT for mobile phone data, water usage data, and S&P 500 data are shown in Fig. 10.13(a) and (b), Fig. 10.14(a) and (b), and Fig. 10.15(a) and (b), respectively.



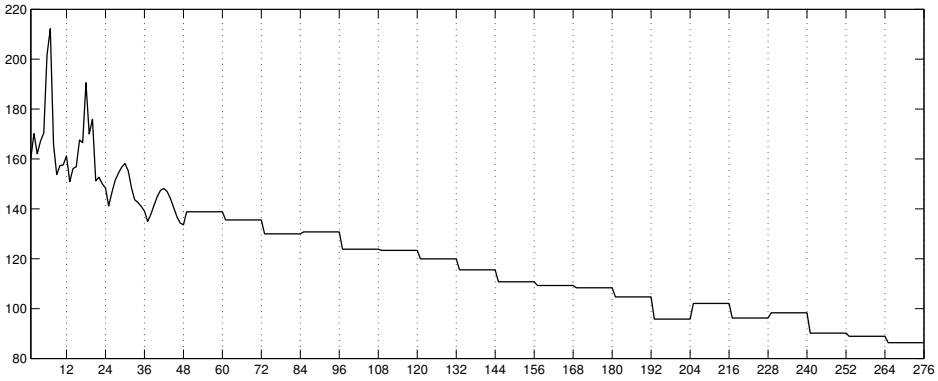
(a)



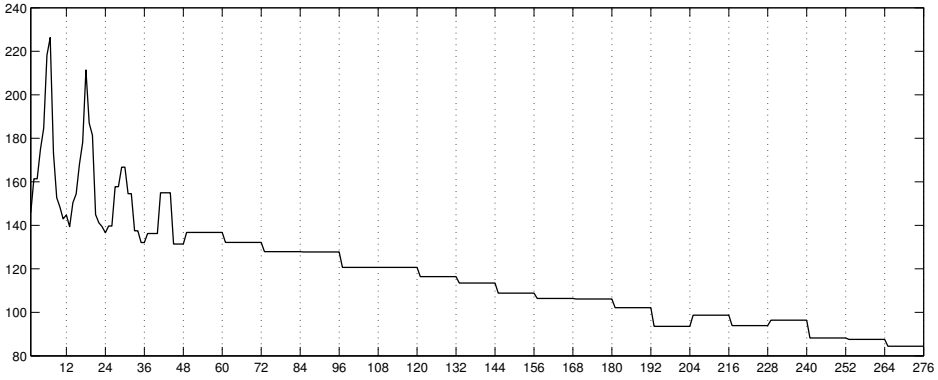
(b)

FIGURE 10.13. (a) The reconstructed time series of the mobile phone data of 75 selected DFT coefficients from the original data of 144 data points, which is 48% reduction. (b) The reconstructed time series of the mobile phone data with 51% reduction by keeping 70 DWT coefficients from the original data of 144 data points.

To compare the performance of my proposed framework with other recent-biased dimension-reduction techniques, a criterion is designed to measure the effectiveness of the algorithm after dimension reduction as following.



(a)



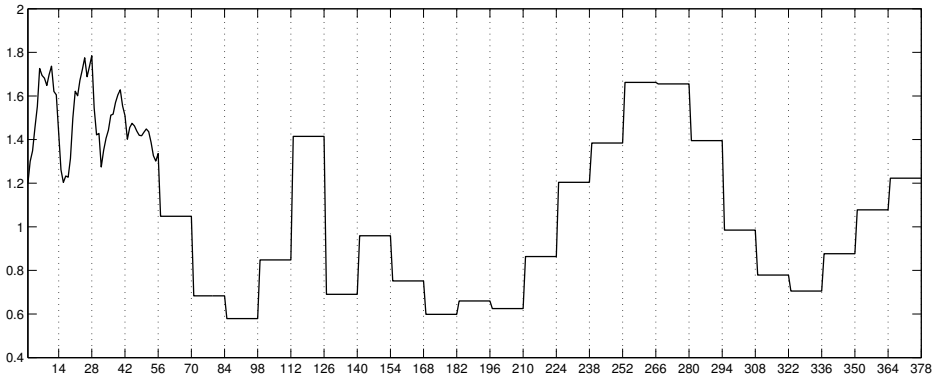
(b)

FIGURE 10.14. (a) The reconstructed time series of the water usage data of 46 selected DFT coefficients from the original data of 276 data points, which is 83% reduction. (b) The reconstructed time series of the water usage data with 81% reduction by keeping 52 DWT coefficients from the original data of 276 data points.

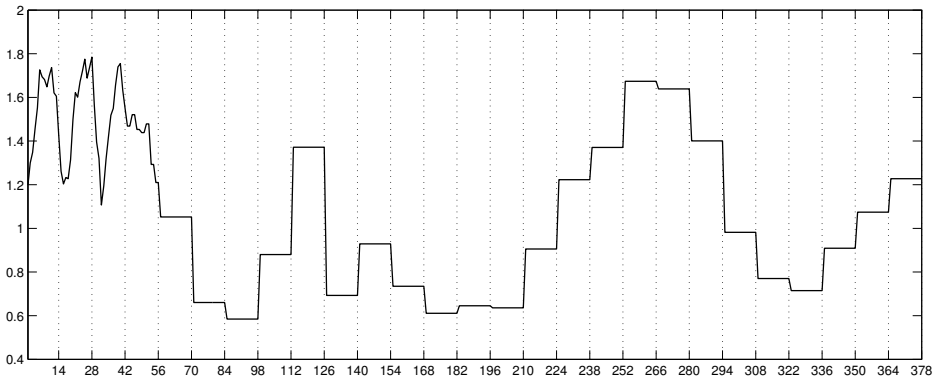
DEFINITION 10.7. If X and $\tilde{X}$ are the original and reconstructed time series, respectively, then the "recent-pattern biased error rate" is defined as

$$Err_{RPB}(X, \tilde{X}) = \mathbf{B} \cdot d_H^2(\hat{X}, \hat{\tilde{X}}) = \frac{1}{2} \sum_{i=0}^{\lfloor N/p \rfloor - 1} b(i) \left(\sqrt{\hat{x}_i} - \sqrt{\hat{\tilde{x}}_i} \right)^2, \tag{106}$$

where $\mathbf{B}$ is a recent-pattern biased vector (which is a sigmoid function in my case).



(a)



(b)

FIGURE 10.15. (a) The reconstructed time series of the S&P 500 data of 66 selected DFT coefficients from the original data of 378 data points, which is 83% reduction. (b) The reconstructed time series of the S&P 500 data with 81% reduction by keeping 72 DWT coefficients from the original data of 378 data points.

DEFINITION 10.8. If X and $\tilde{X}$ are the original and reconstructed time series, respectively and $Err_{RPB}(X, \tilde{X})$ is the recent-pattern biased error rate, then the Reduction-to-Error Ratio (RER) is defined as

$$RER = \frac{Percentage\ Reduction}{Err_{RPB}(X, \tilde{X})}. \tag{107}$$

I compare the performance of my recent-pattern biased dimension-reduction algorithm (RP-DFT/DWT) to equi-DFT/DWT, vari-DFT/DWT (with $k = 8$ (228)), and SWAT as I apply these algorithms on the mobile phone, water usage, and S&P 500 data.

Table 10.1 shows the values of percentage reduction, recent-pattern biased error rate, and RER for each algorithm based on DFT. It shows that SWAT has the highest reduction rates as well as the highest error rates in all three data. For the mobile phone data, the values of the percentage reduction are the same for my RP-DFT and equi-DFT because R is exactly a half of the time series hence the sigmoid function is placed at the half point of the time series ($N/2$) that makes it similar to equi-DFT (in which the number of coefficients is exponentially decreased). The error rate of my RP-DFT is however better than equi-DFT by keeping more coefficients particularly for the "recent-pattern data" and fewer for older data instead of keeping more coefficients for just recent data and fewer for older data. As a result, RP-DFT performs with the best RER among others. For the water usage data, even though RP-DFT has a higher error rate than equi-DFT, R is a relatively short portion with respect to the entire series thus RP-DFT is able to achieve much higher reduction rate, which results in a better RER and the best among others. For S&P 500 data, my RP-DFT is able to reduce more data than equi-DFT and vari-DFT with the lowest error rate, hence it has the highest RER.

TABLE 10.1. Performance comparison of my proposed RP-DFT and other well-known techniques (equi-DFT, vari-DFT, and SWAT) based on Percentage Reduction, Recent-pattern biased error rate (Err_{RBP}), and RER from the real data.

Data	Percentage Reduction				Err_{RBP}				RER			
	RP-DFT	equi-DFT	vari-DFT	SWAT	RP-DFT	equi-DFT	vari-DFT	SWAT	RP-DFT	equi-DFT	vari-DFT	SWAT
Mobile phone	0.479	0.479	0.750	0.972	0.0175	0.0301	0.0427	0.192	**27.458**	15.915	17.573	5.078
Water usage	0.837	0.479	0.739	0.986	0.00712	0.00605	0.0168	0.0641	**117.550**	79.201	43.996	15.375
S&P 500	0.829	0.479	0.742	0.989	0.00735	0.00739	0.00895	0.0811	**112.891**	64.875	82.899	12.210

Likewise, Table 10.2 shows the values of percentage reduction, recent-pattern biased error rate, and RER for each algorithm based on DWT. Similar to the results of the DFT-based

algorithms, my proposed RP-DWT performs with the best RER among other algorithms in all three data. One may notice that the values of the percentage reduction are different from DFT-based algorithms. This is due to the rounding process of C_i to the closest integer v (described in Section 3.3).

TABLE 10.2. Performance comparison of my proposed RP-DWT and other well-known techniques (equi-DWT, vari-DWT, and SWAT) based on Percentage Reduction, Recent-pattern biased error rate (Err_{RBP}), and RER from the real data.

Data	Percentage Reduction				Err_{RBP}				RER			
	RP-DWT	equi-DWT	vari-DWT	SWAT	RP-DWT	equi-DWT	vari-DWT	SWAT	RP-DWT	equi-DWT	vari-DWT	SWAT
Mobile phone	0.514	0.500	0.750	0.972	0.0167	0.0283	0.0401	0.192	**30.794**	17.683	18.689	5.078
Water usage	0.812	0.493	0.739	0.986	0.00650	0.00561	0.0159	0.0650	**124.852**	87.854	46.565	15.152
S&P 500	0.810	0.495	0.742	0.989	0.00728	0.00711	0.00856	0.0811	**111.182**	69.561	87.783	12.211

Furthermore, I perform additional experiments on 30 more real time series , which represent data in finance, health, chemistry, hydrology, industry, labour market, macro-economic, and physics. These data are publicly available at the "Time Series Data Library(254)," which has been created by professor Rob J. Hyndman from Monash University. My RP-DFT/DWT also show better performance than other techniques for all 30 time series data (the results are shown in the Appendix).

In addition to the results of the performance comparison on the real data, I generate 100 synthetic data with different values of p and R to further evaluate my algorithm compared to the others. After applying each algorithm to these 100 different synthetic time series, Table 10.3 and Table 10.4 show the average values of percentage reduction, recent-pattern biased error rate, and RER for each algorithm based on DFT and DWT, respectively. These tables show that my proposed algorithm (both DFT-based and DWT-based) yields better RER than others.

TABLE 10.3. Performance comparison of my proposed RP-DFT and other well-known techniques (equi-DFT, vari-DFT, and SWAT) based on the average Percentage Reduction, Recent-pattern biased error rate (Err_{RBP}), and RER from 100 synthetic data.

Algorithm	Percentage Reduction	Err_{RBP}	RER
RP-DFT	0.758	0.0209	**36.268**
equi-DFT	0.481	0.0192	25.052
vari-DFT	0.748	0.0385	19.429
SWAT	0.975	0.109	8.945

TABLE 10.4. Performance comparison of my proposed RP-DWT and other well-known techniques (equi-DWT, vari-DWT, and SWAT) based on the average Percentage Reduction, Recent-pattern biased error rate (Err_{RBP}), and RER from 100 synthetic data.

Algorithm	Percentage Reduction	Err_{RBP}	RER
RP-DWT	0.745	0.0192	**38.802**
equi-DWT	0.488	0.0190	25.682
vari-DWT	0.748	0.0341	21.935
SWAT	0.975	0.108	9.028

10.5. Conclusion

Dimensionality reduction is an essential process of many high-dimensional data analysis. In this chapter, I present a new recent-pattern biased dimension-reduction framework for time series data. With my framework, more details are kept for recent-pattern data, while older data are kept at coarser level. Unlike other recently proposed dimension reduction techniques for recent-biased time series analysis, my framework emphasizes on keeping the data that carries the most recent pattern, which is the most important data portion in the time series with a high resolution while retaining older data with a lower resolution. I

show that several dimension-reduction techniques such DFT and DWT can be used with my framework. Moreover, I also show that it is simple and efficient to handle dynamic data streams with my framework. My experiments on synthetic data as well as real data demonstrate that my proposed framework is very efficient and it outperforms other well-known recent-biased dimension reduction techniques. As my future directions, I will continue to examine various aspects of my framework to improve its performance.

CHAPTER 11

CONCLUSION

Context awareness is the idea of computing devices sense and respond to the user's context. There have been numerous attempts to formally define "context" (48; 49; 50; 51; 52; 53; 54; 55). According to the latest definition given by Han et al. (55), context can be divided into social context, internal context, and physical context. As an increasing number of people rely more on mobile phones, mobile phones have become more personalized. This personalization turns the mobile phone into a sensor that can essentially monitor a user's behavior. Currently, the mobile phone is lacking the ability to mine these monitored behavior then sense and react intelligently to the context. The ultimate goal is to build intelligence into the mobile phone to enhance quality of life. In this dissertation, I thus focus on developing models for inferring social and internal context based on mobile phone records.

This dissertation presents a combination of empirical work, measurements, experiments, explanatory modeling, analysis of mathematical models, design of algorithms, data mining techniques, survey study, and designs of frameworks. The research focus of this dissertation is to analyze and infer social and internal context of mobile phone users. This dissertation contributes to three research areas including social context-aware computing, internal context-aware computing, and data mining. In social context-aware computing, I analyze calling patterns and develop a framework for inferring social context. In internal context-aware computing, I present new mobile phone applications and develop a model for inferring intentional and situational context. In data mining, I develop frameworks for computing and detecting recent behavior/pattern of data.

11.1. Summary of Contributions

I summarize the contributions of this dissertation by grouping them by the area of contribution with references to the chapters and research questions/answers as shown in Table 11.1. The dissertation includes to the following steps:

(1) I start with analyzing mobile social data as a whole, which then leads to a analysis of the individual users by inferring social context and revealing the significant role of social context in communication patterns.

(2) I then zoom in to individual internal context and its applications for the mobile phone users.

(3) Inferring social and internal context of human users introduces a challenging research problem in human behavioral data mining. I thus take an initial step to address the problem and propose the solutions.

TABLE 11.1. Structure of the dissertation with references to the chapters and research questions

Areas	Chapters	Research Questions	Answers
Social context-aware computing	2, 3	Q1	A1
	4, 5	Q2	A2
Internal context-aware computing	6, 7	Q3	A3
	8	Q4	A4
Human behavioral data mining	9, 10	Q5	A5

11.1.1. Research Questions

Q1. What information can be extracted from a given set of call logs? Can any relationships be drawn? What is the usefulness of these information and relationships?

Q2. Can social context be accurately inferred from a given user's call logs? How does social context impact calling behavior? What is the usefulness of the inferred social context?

Q3. Can a mobile phone infer the user's context such as intention? What can the inferred context be used to benefit the user?

Q4. Can a mobile phone infer the user's context such as situation? What can the inferred context be used to benefit the user?

Q5. Human behavior data mining is an integral part of context-aware computing. Context is determined by the current state of mind (internal), relationship (social), and surroundings (physical). Thus the current state of context is important and can be derived from the recent behavior and pattern. Can the recent behavior be detected? What can the detected recent behavior be used for?

11.1.2. Answers to Research Questions

A1. For a given set of call logs, I am able to capture calling behavior on various features. These features are used in randomness analysis and classification. Based on randomness analysis, I find that (i) the randomness associated with the user's location is highly correlated with calling time and vice versa, and (ii) the randomness in inter-connected time is highly correlated with the time spent talking on each phone call. Based on classification analysis, I find that the call logs can be used to accurately classify face-to-face social networks. The findings extend my understanding of the mobile phone user's calling behavior pattern and are useful for mobile phone service computing (*e.g.*, providing the right service for the right user) and business marketing (*e.g.*, targeting the right market segment).

A2. For a given user's call logs, the social context (social closeness and social tie) can be accurately inferred based on the amount of time and intensity of communication. Furthermore, social group sizes and their successive ratio can be identified. I find that social context play a significant role in calling behavior. My studies show that (i) the closer the social tie, the higher the similarity, (ii) a closer tie implies higher reciprocity, and (iii) the inter-contact time increases as social closeness becomes distant. The inferred social context can be beneficial to mobile phone service provider,

privacy settings, anomaly detection, phone call filtering, epidemiology (mobile virus outbreaks), and business marketing.

A3. The user's context such as intention of making a phone call to a particular person can be inferred by making use of the call history. Probability of making such as phone call be estimated using machine learning techniques. With the same algorithm, probability of receiving a phone call from a particular person can also be estimated. This intentional context can be used to provide three useful applications for the user. First, a list of the most likely contacts/numbers to be dialed can be generated when the user wants to make a call (by flipping open or unlocking the phone). This helps reduce the searching time. Second, a list of numbers/contacts that are most likely to be the callers within the next hour can be generated for the user. This is useful in situations that the user is certain about his/her unavailability for accepting any incoming calls over the next hour (e.g., having a flight, attending a class, having a meeting) thus it is important to know who will be calling during the next hour so the user could perhaps make a call to the persons to inform of his/her next-hour schedule as some calls could be too important to miss. Third, a next-day incoming-call forecast that estimates the arrival time of incoming calls. This can be used to assist daily scheduling (helps avoid unwanted calls and schedule a time for wanted calls).

A4. The user's situational context can be inferred using embedded sensors such as accelerometer, GPS antenna, and microphone. The situational context can be classified into different states such as Uninterruptible by Ringer state (*e.g.*, in a meeting, in a movie theater, etc.), Interruptible by Ringer - Vehicular state (*i.e.*, driving a vehicle), and Interruptible by Ringer - Non Vehicular state (*i.e.*, walking, jogging, shopping, etc.). The inferred context state can be used to control the alert mode (vibrate, handsfree, ringer) by automatically setting alert mode according to the user's context state. With this functionality, forgetting to switch to vibrate mode

while in a movie theater or a meeting, and taking the risk of picking up a phone call while driving can be avoided.

A5. Recent behavior can be detected where the behavior can be treated as a density function or a time series. When treated as a density function, Gaussian distribution is assumed such that the behavior is characterized by mean and variance. The convergence of mean and variance in reverse time determines the recent behavior. On the other hand, when treated as a time series, the recent behavior is characterized by amplitude and shape. The recent behavior is determined by the significant change in the underlying features. The detected recent behavior is very useful for context-aware computing, predictive modeling, and data reduction.

11.2. Vision of Future Studies

The emergence of the mobile social networking applications suggests that the mobile device user population is on the rise. People are expected to engage with their mobile devices longer and more often. This opens up a unique opportunity for computer scientists to not only study but also design and create computing systems that comprehend individual's as well as network's behavior and context. As systems like Facebook, MySpace, hi5, Twitter, and Google are expanding to mobile networks, a new social networking paradigm is created. This unique network inherit some aspects of face-to-face and mobile social networks as well as introduces new characteristics. My future research direction is to harness mobile online social networks to understand, predict, and ultimately, enhance mobile social systems.

APPENDIX A

SURVEY OF MOBILE PHONE USAGE AND SOCIAL CLOSENESS

The following is the survey that we have used for our analysis of mobile social groups and its validation in section 4.2.2:

- - - - - Survey begins here - - - - -

Behavior Analysis of Mobile Phone Users

This project is aimed to provide a better understanding of behavior, pattern, and social structure of mobile phone users, as well as facilitate research and development of mobile social applications as we take an early evolutionary steps toward a new era of mobile and pervasive computing, which is aimed to enhance quality of life with more sensitive and responsive mobile devices.

Survey Process:

The survey process is carried in two simple steps. First, you will download the call record details (records having details of each call you have dialed or received) from your service provider's website. The required information for the survey from the downloaded call records are explained in the *Data Collection* process. You are requested to bring the downloaded information (soft copy) to a 15 minute session. Then, you will review how to identify mobile social closeness for your associated callers/callees in the *Mobile Social Closeness Identification* process and provide your social closeness for each associated call ID in the *Feedback* process.

1. Data Collection: You are requested to download the call detail records from your cellular service providers for the last 3 months (longer period preferred). You would be able to download these call records in an Excel sheet format (we can help you if have any problem in this regard). Next, you need to merge the call records of all those months into a single excel file (again we can help you if required). The call information for our survey is described in the following table. When necessary, you may have to remove some unnecessary data (fields) in the excel sheet.

Date	Start Time	Type	Anonymous Call ID	Talk-time
1/5/2008	2:20 PM	Incoming	C1	2
1/6/2008	3:15 PM	Outgoing	C2	28
...	...	...	...	...

(1) Date: Date when the call has taken place. This should be in the format MM/DD/YYYY.

(2) Start Time: Start time of the call. This should be in the format HH:MM AM/PM.

(3) Type: The call type *i.e.,* whether the call is an "Incoming" or an "Outgoing" call. Some service providers record this field as "Incoming" for an incoming call and the destination location for an outgoing call.

(4) Anonymous Call ID: You can choose an anonymous Call ID to each of the caller/callee. If you are unable to do that, we can run your data through our system and generate a set of anonymous ID's to each caller/callee.

(5) Talk time: The amount of time spent during the call.

2. Mobile Social Closeness Identification: You are requested to attend a 15 minute session for the data analysis. You are requested to identify the Social Closeness for each Call ID as following:

Enter "**1**" if Call ID indicates the person who is a *Socially Closest Member*:
These are the people with whom you maintain the highest socially connectivity. Most of the calls you receive, come from individuals within this category. You receive more calls from them and you tend to talk with them for longer periods. Typically, the face-to-face social tie of these people is family member, friend, and colleagues.

Enter "**2**" if Call ID indicates the person who is a *Socially Near Member*:
People in this group are not as highly connected as family members and friends, but when you connect to them, you talk to them for considerably longer periods. Mostly, you observe

intermittent frequency of calls from these people. These people are typically neighbors and distant relatives.

Enter "**3**" if Call ID indicates the person who is a *Socially Distant Member*:
These individuals have less connection with your social life. These people call you with less frequency. You acknowledge them rarely. Among these would be, for example, a newsletter group or a private organization with whom you have previously subscribed. This group also includes individuals who have no previous interaction or communication with you. You have the least tolerance for calls from them e.g., strangers, telemarketers, fund raisers.

3. Feedback: Your call records will be processed using our system to extract all distinct Call IDs, and then you will be asked to identify Social Closeness for each Call ID as shown in an example below.

Anonymous Call ID	Social Closeness
C1	1
C2	3
C3	2
C4	1
.	.
.	.
.	.

- - - - - Survey ends here - - - - -

APPENDIX B

EXPERIMENTAL RESULTS OF EACH SUBJECT FOR EACH MODEL DESCRIBED IN CHAPTER 8

This appendix includes the experimental results of each subject for each model described in Chapter 8.

TABLE B.1. Performance of FCM for Subject 1 and 2.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	300	0	96	0	288	12	684	12	98.28
2	180	0	142	38	318	42	640	80	88.89
3	300	0	142	278	0	0	442	278	61.39
4	660	0	40	20	0	0	700	20	97.22
5	120	0	116	184	12	288	248	472	34.44
Total	1560	0	536	520	618	342	2714	862	75.89
Acc./Cont.	100.00		50.76		64.38		75.89		

(a) Subject 1

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	270	0	70	26	182	78	522	104	83.39
2	220	0	172	8	338	14	730	22	97.07
3	300	0	138	292	0	0	438	292	60.00
4	666	0	46	24	0	0	712	24	96.74
5	132	0	142	180	4	304	278	484	36.48
Total	1588	0	568	530	524	396	2680	926	74.32
Acc./Cont.	100.00		51.73		56.96		74.32		

(b) Subject 2

TABLE B.2. Performance of FCM for Subject 3 and 4.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	290	0	76	20	272	18	638	38	94.38
2	170	0	160	14	334	26	664	40	94.32
3	300	0	150	296	0	0	450	296	60.32
4	674	0	68	22	0	0	742	22	97.12
5	126	0	78	214	2	286	206	500	29.18
Total	1560	0	532	566	608	330	2700	896	75.08
Acc./Cont.	100.00		48.45		64.82		75.08		

(c) Subject 3

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	278	0	74	4	270	0	622	4	99.36
2	180	0	164	16	366	26	710	42	94.41
3	352	0	134	246	0	0	486	246	66.39
4	686	0	54	18	0	0	740	18	97.63
5	134	0	258	66	0	308	392	374	51.17
Total	1630	0	684	350	636	334	2950	684	81.18
Acc./Cont.	100.00		66.15		65.57		81.18		

(d) Subject 4

TABLE B.3. Performance of GCM for Subject 1 and 2.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	300	0	96	0	300	0	696	0	100.00
2	180	0	174	6	360	0	714	6	99.17
3	296	4	208	212	0	0	504	216	70.00
4	652	8	56	4	0	0	708	12	98.33
5	108	12	252	48	236	64	596	124	82.78
Total	1536	24	786	270	896	64	3218	358	89.99
Acc./Cont.	98.46		74.43		93.33		89.99		

(a) Subject 1

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	268	2	92	4	218	42	578	48	92.33
2	220	0	180	0	352	0	752	0	100.00
3	300	0	184	246	0	0	484	246	66.30
4	642	24	64	6	0	0	706	30	95.92
5	132	0	282	40	226	82	640	122	83.99
Total	1562	26	802	296	796	124	3160	446	87.63
Acc./Cont.	98.36		73.04		86.52		87.63		

(b) Subject 2

TABLE B.4. Performance of GCM for Subject 3 and 4.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	290	0	94	2	290	0	674	2	99.70
2	170	0	174	0	360	0	704	0	100.00
3	300	0	332	114	0	0	632	114	84.72
4	656	18	90	0	0	0	746	18	97.64
5	126	0	170	122	214	74	510	196	72.24
Total	1542	18	860	238	864	74	3266	330	90.82
Acc./Cont.	98.85		78.32		92.11		90.82		

(c) Subject 3

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	278	0	78	0	270	0	626	0	100.00
2	180	0	180	0	392	0	752	0	100.00
3	350	2	300	80	0	0	650	82	88.80
4	678	8	62	10	0	0	740	18	97.63
5	134	0	298	26	206	102	638	128	83.29
Total	1620	10	918	116	868	102	3406	228	93.73
Acc./Cont.	99.39		88.78		89.48		93.73		

(d) Subject 4

TABLE B.5. Performance of MCM for Subject 1 and 2.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	298	2	96	0	300	0	694	2	99.71
2	180	0	174	6	360	0	714	6	99.17
3	296	4	188	232	0	0	484	236	67.22
4	656	4	54	6	0	0	710	10	98.61
5	114	6	162	138	246	54	522	198	72.50
Total	1544	16	674	382	906	54	3124	452	87.36
Acc./Cont.	98.97		63.83		94.38		87.36		

(a) Subject 1

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	266	4	94	2	222	38	582	44	92.97
2	220	0	178	2	352	0	750	2	99.73
3	300	0	166	264	0	0	466	264	63.84
4	652	14	70	0	0	0	722	14	98.10
5	132	0	216	106	258	50	606	156	79.53
Total	1570	18	724	374	832	88	3126	480	86.69
Acc./Cont.	98.87		65.94		90.43		86.69		

(b) Subject 2

TABLE B.6. Performance of MCM for Subject 3 and 4.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	288	2	94	2	290	0	672	4	99.41
2	170	0	174	0	360	0	704	0	100.00
3	298	2	314	132	0	0	612	134	82.04
4	658	16	80	10	0	0	738	26	96.60
5	126	0	116	176	230	58	472	234	66.86
Total	1540	20	778	320	880	58	3198	398	88.93
Acc./Cont.	98.72		70.86		93.82		88.93		

(c) Subject 3

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	276	2	78	0	270	0	624	2	99.68
2	180	0	180	0	392	0	752	0	100.00
3	352	0	290	90	0	0	642	90	87.70
4	680	6	62	10	0	0	742	16	97.89
5	134	0	294	30	240	68	668	98	87.21
Total	1622	8	904	130	902	68	3428	206	94.33
Acc./Cont.	99.51		87.43		92.99		94.33		

(d) Subject 4

TABLE B.7. Performance of MCM-S for Subject 1 and 2.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	298	2	96	0	300	0	694	2	99.71
2	180	0	174	6	360	0	714	6	99.17
3	296	4	188	232	0	0	484	236	67.22
4	656	4	54	6	0	0	710	10	98.61
5	114	6	162	138	300	0	576	144	80.00
Total	1544	16	674	382	960	0	3178	398	88.87
Acc./Cont.	98.97		63.83		100.00		88.87		

(a) Subject 1

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	266	4	94	2	260	0	620	6	99.04
2	220	0	178	2	352	0	750	2	99.73
3	300	0	166	264	0	0	466	264	63.84
4	652	14	70	0	0	0	722	14	98.10
5	132	0	216	106	308	0	656	106	86.09
Total	1570	18	724	374	920	0	3214	392	89.13
Acc./Cont.	98.87		65.94		100.00		89.13		

(b) Subject 2

TABLE B.8. Performance of MCM-S for Subject 3 and 4.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	288	2	94	2	290	0	672	4	99.41
2	170	0	174	0	360	0	704	0	100.00
3	298	2	314	132	0	0	612	134	82.04
4	658	16	80	10	0	0	738	26	96.60
5	126	0	116	176	288	0	530	176	75.07
Total	1540	20	778	320	938	0	3256	340	90.55
Acc./Cont.	98.72		70.86		100.00		90.55		

(c) Subject 3

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	276	2	78	0	270	0	624	2	99.68
2	180	0	180	0	392	0	752	0	100.00
3	352	0	290	90	0	0	642	90	87.70
4	680	6	62	10	0	0	742	16	97.89
5	134	0	294	30	308	0	736	30	96.08
Total	1622	8	904	130	970	0	3496	138	96.20
Acc./Cont.	99.51		87.43		100.00		96.20		

(d) Subject 4

TABLE B.9. Performance of MCM-S(noPCA) for Subject 1 and 2.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	300	0	86	10	300	0	686	10	98.56
2	180	0	140	40	360	0	680	40	94.44
3	298	2	138	282	0	0	436	284	60.56
4	660	0	42	18	0	0	702	18	97.50
5	120	0	120	180	300	0	540	180	75.00
Total	1558	2	526	530	960	0	3044	532	85.12
Acc./Cont.	99.87		49.81		100.00		85.12		

(a) Subject 1

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	270	0	62	34	260	0	592	34	94.57
2	220	0	160	20	352	0	732	20	97.34
3	300	0	136	294	0	0	436	294	59.73
4	666	0	46	24	0	0	712	24	96.74
5	132	0	154	168	308	0	594	168	77.95
Total	1588	0	558	540	920	0	3066	540	85.02
Acc./Cont.	100.00		50.82		100.00		85.02		

(b) Subject 2

TABLE B.10. Performance of MCM-S(noPCA) for Subject 3 and 4.

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	290	0	74	22	290	0	654	22	96.75
2	170	0	134	40	360	0	664	40	94.32
3	300	0	138	308	0	0	438	308	58.71
4	674	0	68	22	0	0	742	22	97.12
5	126	0	96	196	288	0	510	196	72.24
Total	1560	0	510	588	938	0	3008	588	83.65
Acc./Cont.	100.00		46.45		100.00		83.65		

(c) Subject 3

Sequence Number	UR		IR-N		IR-V		Overall		Acc./Seq. (%)
	Hit	Miss	Hit	Miss	Hit	Miss	Hit	Miss	
1	278	0	68	10	270	0	616	10	98.40
2	180	0	146	34	392	0	718	34	95.48
3	352	0	122	258	0	0	474	258	64.75
4	686	0	52	20	0	0	738	20	97.36
5	134	0	236	88	308	0	678	88	88.51
Total	1630	0	624	410	970	0	3224	410	88.72
Acc./Cont.	100.00		60.35		100.00		88.72		

(d) Subject 4

APPENDIX C

ADDITIONAL RESULTS FOR PERFORMANCE COMPARISON OF THE PROPOSED METHOD (RP-DFT/DWT) WITH EQUI-DFT/DWT, VARI-DFT/DWT, AND SWAT

The following are the additional results for performance comparison of our proposed method (RP-DFT/DWT) with equi-DFT/DWT, vari-DFT/DWT, and SWAT; using 30 different real time series, which represent data in finance, health, chemistry, hydrology, industry, labour market, macro-economic, and physics. These time series data were taken from the "Time Series Data Library(254),". Tables C.1 and C.2 show the results based on DFT and DWT, respectively where Table C.3 gives brief description of these data. Our proposed framework shows better performance than other techniques for all 30 time series data.

TABLE C.1. Performance comparison based on DFT, from additional 30 real data.

Data	Percentage Reduction				Err_{RBP}				RER			
	RP-DFT	equi-DFT	vari-DFT	SWAT	RP-DFT	equi-DFT	vari-DFT	SWAT	RP-DFT	equi-DFT	vari-DFT	SWAT
1	0.471	0.494	0.738	0.963	0.0163	0.0306	0.0376	0.1696	**28.974**	16.156	19.644	5.676
2	0.331	0.493	0.744	0.949	0.0053	0.0133	0.0158	0.1408	**62.464**	37.043	47.194	6.738
3	0.345	0.479	0.714	0.952	0.0122	0.0470	0.1055	0.1968	**28.201**	10.203	6.768	4.838
4	0.316	0.485	0.749	0.984	0.0104	0.0272	0.0549	0.2320	**30.446**	17.814	13.640	4.242
5	0.818	0.481	0.740	0.989	0.0140	0.0108	0.0142	0.0207	**58.611**	44.555	52.130	47.714
6	0.235	0.485	0.749	0.984	0.0075	0.0464	0.0492	0.2334	**31.375**	10.454	15.231	4.217
7	0.951	0.479	0.749	0.999	0.0177	0.0168	0.0176	0.0214	**53.822**	28.600	42.583	46.767
8	0.904	0.479	0.749	0.999	0.0059	0.0057	0.0064	0.0099	**153.961**	84.512	117.506	101.118
9	0.315	0.482	0.745	0.975	0.0154	0.0268	0.0504	0.2010	**20.485**	18.019	14.800	4.848
10	0.622	0.493	0.730	0.946	0.0074	0.0070	0.0098	0.0156	**84.252**	70.943	74.488	60.764
11	0.478	0.482	0.736	0.980	0.0020	0.0028	0.0062	0.0206	**235.127**	172.483	118.015	47.460
12	0.381	0.484	0.735	0.982	0.0267	0.0570	0.0713	0.1784	**14.253**	8.481	10.304	5.506
13	0.377	0.483	0.742	0.987	0.0073	0.0131	0.0219	0.1298	**51.375**	36.699	33.859	7.602
14	0.350	0.479	0.720	0.960	0.0120	0.0447	0.0943	0.1598	**29.046**	10.714	7.632	6.009
15	0.378	0.482	0.736	0.980	0.0163	0.0279	0.0424	0.2064	**23.148**	17.248	17.372	4.746
16	0.410	0.479	0.747	0.993	0.0115	0.0303	0.0600	0.3440	**35.586**	15.835	12.443	2.888
17	0.417	0.479	0.750	0.958	0.0093	0.0151	0.0275	0.2178	**44.835**	31.777	27.246	4.399
18	0.496	0.479	0.750	0.992	0.0011	0.0011	0.0021	0.0043	**466.390**	437.727	355.765	233.486
19	0.681	0.490	0.745	0.957	0.0075	0.0069	0.0103	0.0279	**90.373**	70.876	72.485	34.337
20	0.357	0.479	0.743	0.986	0.0052	0.0073	0.0177	0.2222	**68.927**	65.709	42.051	4.441
21	0.739	0.479	0.745	0.979	0.0021	0.0020	0.0025	0.0382	**358.734**	240.248	300.908	25.631
22	0.828	0.479	0.745	0.990	0.0037	0.0030	0.0036	0.1191	**223.294**	157.506	209.507	8.314
23	0.330	0.485	0.730	0.978	0.0018	0.0043	0.0055	0.1873	**187.689**	113.363	133.004	5.220
24	0.323	0.487	0.735	0.981	0.0022	0.0041	0.0058	0.2034	**147.656**	119.216	126.773	4.824
25	0.301	0.485	0.730	0.978	0.0021	0.0032	0.0063	0.2184	**142.674**	152.220	116.428	4.475
26	0.397	0.479	0.750	0.972	0.0026	0.0142	0.0237	0.2111	**151.063**	33.807	31.685	4.606
27	0.323	0.491	0.748	0.969	0.0072	0.0141	0.0259	0.2149	**44.891**	34.872	28.884	4.508
28	0.332	0.479	0.747	0.993	0.0226	0.1095	0.1136	0.2594	**14.646**	4.378	6.574	3.830
29	0.909	0.480	0.749	0.999	0.0128	0.0107	0.0123	0.0165	**71.207**	45.014	60.800	60.476
30	0.350	0.479	0.743	0.986	0.0033	0.0050	0.0077	0.0791	**104.790**	95.906	95.947	12.463

TABLE C.2. Performance comparison based on DWT, from additional 30 real data.

Data	Percentage Reduction				Err_{RBP}				RER			
	RP-DWT	equi-DWT	vari-DWT	SWAT	RP-DWT	equi-DWT	vari-DWT	SWAT	RP-DWT	equi-DWT	vari-DWT	SWAT
1	0.434	0.505	0.738	0.963	0.0186	0.0256	0.0295	0.1710	**23.283**	19.752	24.986	5.629
2	0.303	0.500	0.744	0.949	0.0042	0.0116	0.0166	0.1428	**72.848**	43.193	44.743	6.644
3	0.298	0.476	0.714	0.952	0.0118	0.0438	0.1295	0.1813	**25.161**	10.866	5.515	5.254
4	0.308	0.502	0.749	0.984	0.0133	0.0279	0.0477	0.2320	**23.062**	17.995	15.686	4.242
5	0.832	0.501	0.740	0.989	0.0138	0.0106	0.0139	0.0213	**60.150**	47.412	53.225	46.422
6	0.296	0.502	0.749	0.984	0.0075	0.0463	0.0495	0.2370	**39.486**	10.850	15.145	4.153
7	0.953	0.500	0.749	0.999	0.0157	0.0154	0.0160	0.0194	**60.789**	32.369	46.875	51.600
8	0.902	0.500	0.749	0.999	0.0059	0.0052	0.0054	0.0099	**152.863**	95.681	138.949	101.118
9	0.446	0.497	0.745	0.975	0.0152	0.0235	0.0462	0.2113	**29.358**	21.147	16.116	4.612
10	0.622	0.500	0.730	0.946	0.0078	0.0074	0.0096	0.0156	**79.501**	67.430	75.748	60.764
11	0.439	0.492	0.736	0.980	0.0020	0.0028	0.0080	0.0206	**215.898**	176.285	92.084	47.460
12	0.403	0.504	0.735	0.982	0.0286	0.0562	0.0665	0.1784	**14.083**	8.975	11.052	5.506
13	0.426	0.497	0.742	0.987	0.0083	0.0114	0.0263	0.1298	**51.036**	43.611	28.187	7.602
14	0.380	0.480	0.720	0.960	0.0118	0.0507	0.0811	0.1598	**32.077**	9.468	8.879	6.009
15	0.439	0.492	0.736	0.980	0.0163	0.0218	0.0360	0.2064	**26.883**	22.594	20.449	4.746
16	0.380	0.498	0.747	0.993	0.0112	0.0303	0.0615	0.3440	**33.952**	16.448	12.144	2.888
17	0.438	0.500	0.750	0.958	0.0102	0.0165	0.0320	0.2178	**42.852**	30.365	23.415	4.399
18	0.496	0.500	0.750	0.992	0.0011	0.0012	0.0021	0.0043	**444.962**	420.208	357.143	232.963
19	0.670	0.500	0.745	0.957	0.0062	0.0065	0.0096	0.0245	**108.643**	76.868	77.568	39.113
20	0.369	0.497	0.743	0.986	0.0052	0.0080	0.0184	0.2342	**71.344**	62.277	40.454	4.212
21	0.734	0.495	0.745	0.979	0.0025	0.0019	0.0026	0.0382	**295.906**	256.706	281.440	25.631
22	0.821	0.498	0.745	0.990	0.0032	0.0027	0.0030	0.1090	**258.705**	182.266	248.850	9.082
23	0.292	0.506	0.730	0.978	0.0017	0.0042	0.0055	0.1873	**167.909**	120.118	132.497	5.220
24	0.265	0.507	0.735	0.981	0.0016	0.0043	0.0066	0.2034	**168.254**	118.823	111.972	4.824
25	0.369	0.506	0.730	0.978	0.0024	0.0035	0.0061	0.2184	**155.310**	145.351	119.010	4.475
26	0.450	0.500	0.750	0.972	0.0103	0.0132	0.0217	0.2111	**43.689**	37.774	34.549	4.606
27	0.307	0.504	0.748	0.969	0.0072	0.0150	0.0307	0.2101	**42.722**	33.653	24.339	4.611
28	0.277	0.498	0.747	0.993	0.0201	0.1042	0.1155	0.2594	**13.793**	4.782	6.464	3.830
29	0.909	0.501	0.749	0.999	0.0104	0.0126	0.0143	0.0173	**87.039**	39.605	52.464	57.865
30	0.433	0.496	0.743	0.986	0.0028	0.0048	0.0077	0.0817	**154.643**	103.812	95.947	12.071

TABLE C.3. Data description.

Data	Brief Description(254)
1	I.C.I. Closing prices 25 Aug '72-19 Jan '73 (Financial Times).
2	Dow Jones utility index Aug 28-Dec 18 '72 (Wall Street Journal).
3	Monthly returns for AT&T, Jan 1961 through Dec. 1967.
4	Monthly interest rates Government Bond Yield 2-year securities, Reserve Bank of Australia.
5	IBM common stock closing prices: daily, 17th May 1961 to 2nd November 1962.
6	IBM common stock closing prices: daily, 29th June 1959 to 30th June 1960.
7	Daily closing price of IBM stock, Jan. 1st 1980 - Oct. 8th 1992.
8	Daily S & P 500 index of stocks, Jan. 1st 1980 - Oct. 8th 1992.
9	Monthly closings of the Dow-Jones industrial index, Aug. 1968 - Aug. 1981.
10	Annual yield of grain on Broadbalk field at Rothamsted 1852-1925.
11	Chemical concentration readings.
12	Chemical process temperature readings.
13	Chemical process viscosity readings.
14	Chemical process: viscocity data.
15	Chemical process concentration readings.
16	SacClearwater river at Kamiah, Idaho. 1911 – 1965.
17	Mean monthly flow, tree river, 1969 – 1976.
18	Monthly temperature, coppermine, 1933 – 1976.
19	Monthly demand repair parts large/heavy equip. Iowa 1972 – 1979.
20	Carbon dioxide output from gas furnace: percent of output gas. Sampling interval 9 seconds.
21	Motor vehiclesengines and parts/CPI, Canada, 1976-1991.
22	Monthly U.S. female (20 years and over) unemployment figures (10**3) 1948-1981.
23	Wisconsin employment time series, food and kindred products, Jan. 1961 - OCt. 1975.
24	Civilian labour force in Australia each month: thousands of persons. Feb 1978 - Aug 1995.
25	Wisconsin employment time series, fabricated metals, Jan. 1961 - OCt. 1975.
26	Quarterly gross fixed capital expenditure - public, Australia: millions of dollars, 1989/90 prices. Sep 1959 - Jun 1995.
27	Quarterly gross fixed capital expenditure - private equipment, Australia: millions of dollars, 1984/85 prices. Sep 1959 - Mar 1991.
28	Daily brightness of a variable star on 600 successive midnights.
29	Monthly means of daily relative sunspot numbers, Jan 1749 - Mar 1977.
30	Annual sunspot numbers 1700-1979.

BIBLIOGRAPHY

[1] M. Weiser, "Some computer science issues in ubiquitous computing," *Commun. ACM*, vol. 36, no. 7, pp. 75–84, 1993.

[2] a. Neto, Renato F. Bulc T. N. Kudo, and M. da Graça Pimentel, "Using a software process for ontology-based context-aware computing: a case study," in *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*. New York, NY, USA: ACM, 2006, pp. 61–70.

[3] J. Choi and H.-J. Moon, "Software engineering issues in developing a context-aware exhibition guide system," in *SNPD '08: Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 840–845.

[4] J. Choi, "Software architecture for extensible context-aware systems," in *ICHIT '08: Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 811–816.

[5] H. Ming, K. Oyama, and C. K. Chang, "Human-intention driven self adaptive software evolvability in distributed service environments," in *FTDCS '08: Proceedings of the 2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 51–57.

[6] C. K. Chang, K. Oyama, H. Jaygarl, and H. Ming, "On distributed run-time software evolution driven by stakeholders of smart home development (invited paper)," in *ISUC '08: Proceedings of the 2008 Second International Symposium on Universal Communication*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 59–66.

[7] C. Patrikakis, P. Karamolegkos, A. Voulodimos, M. H. A. Wahab, N. S. A. M. Taujuddin, C. Hanif, L. Pareschi, D. Riboni, S. G. Weber, A. Heinemann, S.-c. S. Cheung, J. Chaudhari, and J. K. Paruchuri, "Security and privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 73–75, 2007.

[8] S. Trabelsi, L. Gomez, and Y. Roudier, "Context-aware security policy for the service discovery," in *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 477–482.

[9] M. Fahrmair, W. Sitou, and B. Spanfelner, "Privacy management for context transponders," in *SAINT-W '07: Proceedings of the 2007 International Symposium on Applications and the Internet Workshops*. Washington, DC, USA: IEEE Computer Society, 2007, p. 14.

[10] C. Li, Y. Zhang, and L. Duan, "Establishing a trusted architecture on pervasive terminals for securing context processing," in *PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 639–644.

[11] J. Indulska, "Challenges in the design and development of context-aware applications," in *UIC '08: Proceedings of the 5th international conference on Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1–1.

[12] A. Helal and J. Hammer, "Ubidata: requirements and architecture for ubiquitous data access," *SIGMOD Rec.*, vol. 33, no. 4, pp. 71–76, 2004.

[13] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 15.

[14] M. Choi, W. Park, and Y.-K. Kim, "A hybrid cache cohrency scheme for ubiquitous mobile clients," in *ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 181–188.

[15] Y.-H. Lee, A. Thomo, K. Wu, and V. King, "Scalable ubiquitous data access in clustered sensor networks," in *SSDBM '08: Proceedings of the 20th international conference*

on Scientific and Statistical Database Management. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 533–540.

[16] Y. Akahoshi, Y. Kidawara, and K. Tanaka, "A database-oriented wrapper for ubiquitous data acquisition/access environments," in *ICUIMC '08: Proceedings of the 2nd international conference on Ubiquitous information management and communication*. New York, NY, USA: ACM, 2008, pp. 25–32.

[17] S. S. P. Kumar, "Pervasive sensing and computing," in *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2005, p. 3.

[18] H. El Zabadani, "Self sensing spaces," Ph.D. dissertation, Gainesville, FL, USA, 2006, adviser-Helal, Abdelsalam.

[19] T. Sohn, W. G. Griswold, J. Scott, A. LaMarca, Y. Chawathe, I. Smith, and M. Chen, "Experiences with place lab: an open source toolkit for location-aware computing," in *ICSE '06: Proceedings of the 28th international conference on Software engineering*. New York, NY, USA: ACM, 2006, pp. 462–471.

[20] D. J. Yates, E. M. Nahum, J. F. Kurose, and P. Shenoy, "Data quality and query cost in pervasive sensing systems," *Pervasive Mob. Comput.*, vol. 4, no. 6, pp. 851–870, 2008.

[21] C. Anagnostopoulos and S. Hadjiefthymiades, "On the application of epidemical spreading in collaborative context-aware computing," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 4, pp. 43–55, 2008.

[22] Z. Maamar, Q. Z. Sheng, and B. Benatallah, "On composite web services provisioning in an environment of fixed and mobile computing resources," *Inf. Technol. and Management*, vol. 5, no. 3-4, pp. 251–270, 2004.

[23] V. Poladian, J. P. Sousa, D. Garlan, and M. Shaw, "Dynamic configuration of resource-aware services," in *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 604–613.

[24] K. Curran, M. Mulvenna, C. Nugent, and A. Galis, "Challenges and research directions in autonomic communications," *Int. J. Internet Protoc. Technol.*, vol. 2, no. 1, pp. 3–17, 2007.

[25] Z. Maamar, H. Yahyaoui, and Q. H. Mahmoud, "Dynamic management of uddi registries in a wireless environment of web services: Concepts, architecture, operation, and deployment," *J. Intell. Inf. Syst.*, vol. 28, no. 2, pp. 105–131, 2007.

[26] J. Jin and K. Nahrstedt, "Qos-aware service management for component-based distributed applications," *ACM Trans. Internet Technol.*, vol. 8, no. 3, pp. 1–31, 2008.

[27] M. Lawo, O. Herzog, and H. Witt, "An industrial case study on wearable computing applications," in *MobileHCI '07: Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*. New York, NY, USA: ACM, 2007, pp. 448–451.

[28] M. Lawo, O. Herzog, P. Lukowicz, and H. Witt, "Using wearable computing solutions in real-world applications," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 3687–3692.

[29] I. Oakley, J. Sunwoo, and I.-Y. Cho, "Pointing with fingers, hands and arms for wearable computing," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 3255–3260.

[30] P. de la Hamette and G. Tröster, "Architecture and applications of the fingermouse: a smart stereo camera for wearable computing hci," *Personal Ubiquitous Comput.*, vol. 12, no. 2, pp. 97–110, 2008.

[31] H. Iben, "A methodical approach to evaluating the use of audio-interfaces for mobile and wearable computing," in *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. New York, NY, USA: ACM, 2008, pp. 541–542.

[32] B. E. John and D. D. Salvucci, "Multipurpose prototypes for assessing user interfaces in pervasive computing systems," *IEEE Pervasive Computing*, vol. 4, no. 4, pp. 27–34, 2005.

[33] A. Schmidt, L. Terrenghi, and P. Holleis, "Methods and guidelines for the design and development of domestic ubiquitous computing applications," *Pervasive Mob. Comput.*, vol. 3, no. 6, pp. 721–738, 2007.

[34] K. Leichtenstern and E. Andre, "User-centred development of mobile interfaces to a pervasive computing environment," in *ACHI '08: Proceedings of the First International Conference on Advances in Computer-Human Interaction*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 114–119.

[35] S. Neely, G. Stevenson, C. Kray, I. Mulder, K. Connelly, and K. A. Siek, "Evaluating pervasive and ubiquitous systems," *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 85–88, 2008.

[36] S. E. Chang, "Implementation and empirical evaluation of voice-enabled web applications," *Int. J. Inf. Technol. Manage.*, vol. 8, no. 2, pp. 178–195, 2009.

[37] S. Counts, H. ter Hofte, and I. Smith, "Mobile social software: realizing potential, managing risks," in *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2006, pp. 1703–1706.

[38] O. Bohl, S. Manouchehri, S. Ammermueller, and O. Gerstheimer, "Mobile social software - potentials and limitations of enabling social networking on mobile devices," in *ICMB '07: Proceedings of the International Conference on the Management of Mobile Business*. Washington, DC, USA: IEEE Computer Society, 2007, p. 63.

[39] J. Thom-Santelli, "Mobile social software: Facilitating serendipity or encouraging homogeneity?" *IEEE Pervasive Computing*, vol. 6, no. 3, pp. 46–51, 2007.

[40] C. Heyer, M. Brereton, and S. Viller, "Cross-channel mobile social software: an empirical study," in *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 1525–1534.

[41] S. K. Kane and P. V. Klasnja, "Supporting volunteer activities with mobile social software," in *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2009,

pp. 4567–4572.

[42] G. Chen, M. Li, and D. Kotz, "Data-centric middleware for context-aware pervasive computing," *Pervasive Mob. Comput.*, vol. 4, no. 2, pp. 216–253, 2008.

[43] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," in *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*. New York, NY, USA: ACM, 2008, pp. 113–122.

[44] P. Dai and G. Xu, "Context-aware computing for assistive meeting system," in *PETRA '08: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM, 2008, pp. 1–7.

[45] W. Xue, H. Pung, P. P. Palmes, and T. Gu, "Schema matching for context-aware computing," in *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*. New York, NY, USA: ACM, 2008, pp. 292–301.

[46] W. Dargie, *Context-Aware Computing and Self-Managing Systems*. Chapman & Hall/CRC, 2009.

[47] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *WMCSA '94: Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 1994, pp. 85–90.

[48] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, and W. V. d. Velde, "Advanced interaction in context," in *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag, 1999, pp. 89–101.

[49] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Hanover, NH, USA, Tech. Rep., 2000.

[50] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001.

[51] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," in *HICSS*

'03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9. Washington, DC, USA: IEEE Computer Society, 2003, pp. 292–302.

[52] P. Prekop and M. Burnett, "Activities, context and ubiquitous computing," *Computer Communications*, vol. 26, p. 1168, 2003.

[53] N. A. Bradley and M. D. Dunlop, "Toward a multidisciplinary model of context to support context-aware computing," *Hum.-Comput. Interact.*, vol. 20, no. 4, pp. 403–446, 2005.

[54] N. O'Connor, R. Cunningham, and V. Cahill, "Self-adapting context definition," in *SASO '07: Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 336–339.

[55] L. Han, S. Jyri, J. Ma, and K. Yu, "Research on context-aware mobile computing," in *AINAW '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 24–30.

[56] P. Okimoto and K. Ferguson, "Social networking going mobile, nielsen finds u.k. and u.s. lead in assessing social networks via mobile phones," August 8th 2008. [Online]. Available: http://www.nielsenmedia.com/nc/portal/site/Public/menuitem.55dc65b4a7d5adff3f65936147a062a0
?vgnextoid=b03335bccf3c9110VgnVCM100000ac0a260aRCRD

[57] O. Oyewola, "Research Revealed Revenue from Mobile Social Networking Market to Hit $3.3B," August 14th 2008. [Online]. Available: http://solokay.blogspot.com/2008/08/research-revealed-revenue-from-mobile.html

[58] N. Oliver and F. Flores-Mangas, "MPTrain: a mobile, music and physiology-based personal trainer," pp. 21–28, 2006.

[59] A. Clauset and N. Eagle, "ersistence and periodicity in a dynamic proximity network," *Proceedings of Discrete Mathematics and Theoretical Computer Science Workshop on*

Computational Methods for Dynamic Interaction Networks, 2007.

[60] X. Anguera and N. Oliver, "MAMI: multimodal annotations on a camera phone," in *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services.* New York, NY, USA: ACM, 2008, pp. 379–382.

[61] Z. Naor, "Searching for multiple mobile users," *IEEE Transactions on Mobile Computing*, vol. 7, no. 9, pp. 1071–1083, 2008.

[62] C. Driver and S. Clarke, "An application framework for mobile, context-aware trails," *Pervasive Mob. Comput.*, vol. 4, no. 5, pp. 719–736, 2008.

[63] T. Caus, S. Christmann, and S. Hagenhoff, "Development of context-aware mobile services: an approach to simplification," *Int. J. Mob. Commun.*, vol. 7, no. 2, 2009.

[64] M. de Reuver and T. Haaker, "Designing viable business models for context-aware mobile services," *Telemat. Inf.*, vol. 26, no. 3, pp. 240–248, 2009.

[65] B. Skov and T. Hoegh, "Supporting information access in a hospital ward by a context-aware mobile electronic patient record," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 205–214, 2006.

[66] P. Pawar, K. Wac, B.-J. van Beijnum, P. Maret, A. van Halteren, and H. Hermens, "Context-aware middleware architecture for vertical handover support to multi-homed nomadic mobile services," in *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.* New York, NY, USA: ACM, 2008, pp. 481–488.

[67] O. Davidyuk, J. Riekki, V.-M. Rautio, and J. Sun, "Context-aware middleware for mobile multimedia applications," in *MUM '04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia.* New York, NY, USA: ACM, 2004, pp. 213–220.

[68] K. Damasceno, N. Cacho, A. Garcia, A. Romanovsky, and C. Lucena, "Context-aware exception handling in mobile agent systems: the moca case," in *SELMAS '06: Proceedings of the 2006 international workshop on Software engineering for large-scale*

multi-agent systems. New York, NY, USA: ACM, 2006, pp. 37–44.

[69] O. Riva and S. Toivonen, "The dynamos approach to support context-aware service provisioning in mobile environments," *J. Syst. Softw.*, vol. 80, no. 12, pp. 1956–1972, 2007.

[70] C. R. G. de Farias, M. M. Leite, C. Z. Calvi, R. M. Pessoa, and J. G. P. Filho, "A mof metamodel for the development of context-aware mobile applications," in *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing.* New York, NY, USA: ACM, 2007, pp. 947–952.

[71] A. Devaraju, S. Hoh, and M. Hartley, "A context gathering framework for context-aware mobile solutions," in *Mobility '07: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology.* New York, NY, USA: ACM, 2007, pp. 39–46.

[72] A. Krause, "Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 113–127, 2006, senior Member-Smailagic,, Asim and Fellow-Siewiorek,, Daniel P.

[73] J. Ho and S. S. Intille, "Using context-aware computing to reduce the perceived burden of interruptions from mobile devices," in *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 2005, pp. 909–918.

[74] J. Cao, N. Xing, A. T. S. Chan, Y. Feng, and B. Jin, "Service adaptation using fuzzy theory in context-aware mobile computing middleware," in *RTCSA '05: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.* Washington, DC, USA: IEEE Computer Society, 2005, pp. 496–501.

[75] V. Sacramento, M. Endler, and F. N. Nascimento, "A privacy service for context-aware mobile computing," in *SECURECOMM '05: Proceedings of the First International*

Conference on Security and Privacy for Emerging Areas in Communications Networks. Washington, DC, USA: IEEE Computer Society, 2005, pp. 182–193.

[76] L. Yan and K. Sere, "A formalism for context-aware mobile computing," in *ISPDC '04: Proceedings of the Third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 14–21.

[77] S. Drakatos, "Context-aware data caching for mobile computing environments," Ph.D. dissertation, Miami, FL, USA, 2006, adviser-Pissinou, Niki and Adviser-Makki, Kia.

[78] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli, "A mobile computing middleware for location- and context-aware internet data services," *ACM Trans. Internet Technol.*, vol. 6, no. 4, pp. 356–380, 2006.

[79] D. Chalmers, N. Dulay, and M. Sloman, "A framework for contextual mediation in mobile and ubiquitous computing applied to the context-aware adaptation of maps," *Personal Ubiquitous Comput.*, vol. 8, no. 1, pp. 1–18, 2004.

[80] Massachusetts Institute of Technology, "Reality mining project," September 2007. [Online]. Available: http://reality.media.mit.edu

[81] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, May 2006.

[82] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[83] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences (2nd Edition)*. Lawrence Erlbaum, January 1988.

[84] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, October 1986.

[85] R. B. Cattell and S. Vogelmann, "Inferring social network structure using mobile phone data," *Multivariate Behavioral Research*, vol. 12, pp. 289–325, July 2007.

[86] N. Eagle, A. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," *PNAS*, 2007.

[87] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educ. Psychol. Meas.*, vol. 20, pp. 141–151, October 1960.

[88] P. P. De Groen, "An introduction to total least squares," *Niew Archief Voor Wiskunde*, vol. 14, p. 237, 1996.

[89] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, 2002.

[90] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[91] R. J. Quinlan, *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.

[92] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1973.

[93] I. Good, *Probability and the Weighing of Evidence*. Charles Griffin, London, 1950.

[94] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[95] M. P. Wand and M. C. Jones, *Kernel Smoothing (Monographs on Statistics and Applied Probability)*. Chapman & Hall/CRC, December 1994.

[96] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society*, vol. Series B, 1991.

[97] CTIA, "Wireless quick facts," 2003. [Online]. Available: http://www.ctia.org/media/index.cfm/AID/10323

[98] J. S. Coleman, *An Introduction To Mathematical Sociology*. Collier-Macmillan, London, 1964.

[99] C. P. Kottak, *Cultural Anthropology*. Mcgraw Hill, New York, 1991.

[100] R. Scupin, *Cultural Anthropology - A Global Perspective*. Prentice Hall, Englewood Cliffs, 1992.

[101] R. I. M. Dunbar, "Neocortex size as a constraint on group size in primates," *Journal of Human Evolution*, vol. 20, pp. 469–493, 1992.

[102] W. X. Zhou, D. Sornette, R. A. Hill, and R. I. M. Dunbar, "Discrete hierarchical organization of social group sizes," *Proceedings of the Royal Society B: Biological Sciences*, vol. 272, no. 1561, pp. 439–444, 2005.

[103] M. Granovetter, "The strength of weak ties," *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[104] P. V. Marsden and K. E. Campbell, "Measuring tie strength," *Social Forces*, vol. 63, no. 2, pp. 482–501, December 1984.

[105] P. Kolan, R. Dantu, and J. ao W. Cangussu, "Nuisance level of a voice call," *TOMCCAP*, vol. 5, no. 1, 2008.

[106] S. Phithakkitnukoon and R. Dantu, "UNT mobile phone communication dataset," 2008. [Online]. Available: http://nsl.unt.edu/santi/data_desc.pdf

[107] L. Wasserman, "All of statistics: A concise course in statistical inference," *The American Statistician*, vol. 59, no. 2, p. 203, April 2005.

[108] D. Sornette, *Phys. Rep.*, vol. 297, no. 239, 1998.

[109] W. X. Zhou and D. Sornette, *Phys. Rev.*, vol. E 66, no. 046111, 2002.

[110] A. Erzan, *Phys. Lett.*, vol. A, no. 225, 1997.

[111] A. Erzan and J. P. Eckmann, *Phys. Lett.*, vol. 78, no. 3245, 1997.

[112] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in Fortran*. Cambridge University Press, January 1992.

[113] J. Birtchnell, "Personality set within an octagonal model of relating," *American Psychological Association*, vol. Circumplex models of personality and emotions, pp. 155–182, 1997.

[114] L. Jamieson, *Intimacy, personal relationship in modern societies*. Oxford: Polity Press & Blackwell Publishers Ltd., 1998.

[115] M. Popovic, D. Milne, and P. Barrett, "The scale of perceived interpersonal closeness (PICS)," *Clinical Psychology and Psychotherapy*, vol. 10, pp. 286–301, 2003.

[116] K. Kayser and D. P. Himle, "Dysfunctional beliefs about intimacy," *Journal of Cognitive Psychotherapy*, no. 8, pp. 127–139, 1994.

[117] M. D. Sherman and M. H. Thelen, "Fear of Intimacy Scale: validation and extension with adolescents," *Journal of Marital and Family Therapy*, vol. 13, pp. 507–521, 1996.

[118] J. Orlofsky, J. Marcia, and I. Lesser, "Ego identity status and the intimacy versus isolation crisis of young adulthood," *Journal of Personality and Social Psychology*, vol. 27, pp. 211–219, 1973.

[119] M. T. Schaefer and D. H. Olson, "Assessing intimacy: the PAIR inventory," *Journal of Marital and Family Therapy*, vol. 7, pp. 47–60, 1981.

[120] L. Milne, *Social therapy. A guide to social support interventions for mental health practitioners*. Chichester: John Wiley & Sons, 1999.

[121] R. D. Nolker and L. Zhou, "Social computing and weighting to identify member roles in online communities," *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 87–93, 2005.

[122] G. S. Mesch and I. Talmud, "Online friendship formation, communication channels, and social closeness," *International Journal of Internet Science*, vol. 1, no. 1, pp. 26–44, 2006.

[123] A. V. Zhdanova, L. Predoiu, T. Pellegrini, and D. Fensel, "D.: A Social Networking Model of a Web Community," *Proceedings of the 10th International Symposium on Social Communication*, 2007.

[124] J. Bleecker, "What's your social doing in my mobile? design patterns for mobile social software," *Proceeding of WWW2006 Workshop MobEA IV Empowering the Mobile Web*, pp. 1–6, May 2006.

[125] L. Hossain, K. Chung, and S. Murshed, "Exploring temporal communication through social networks," 2007, pp. 19–30.

[126] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[127] R. I. M. Dunbar, "Coevolution of neocortical size, group size and language in humans," *Behavioral and Brain Sciences*, vol. 16, no. 4, pp. 681–735, 1993.

[128] R. I. M. Dunbar and M. Spoors, "Social networks, support cliques, and kinship," *Human Nature*, vol. 6, pp. 273–291, 1995.

[129] R. A. Hill and R. I. M. Dunbar, "Social network size in humans," *Human Nature*, vol. 14, no. 1, pp. 53–72, 2003.

[130] T-Mobile, "MyFaves," June 2009. [Online]. Available: http://www.t-mobile.com/templates/generic.aspx?passet=Pln_Lst_MyFavesLrnDemo

[131] P. Bhaskar and S. I. Ahamed, "Privacy in pervasive computing and open issues," in *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 147–154.

[132] J. Hakkila and J. Mantyjarvi, "Collaboration in context-aware mobile phone applications," in *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1.* Washington, DC, USA: IEEE Computer Society, 2005, p. 33.

[133] X. Wan, E. Milios, N. Kalyaniwalla, and J. Janssen, "Link-based anomaly detection in communication networks," in *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 402–405.

[134] W.-K. Wong, A. Moore, G. Cooper, and M. Wagner, "Rule-based anomaly pattern detection for detecting disease outbreaks," in *Eighteenth national conference on Artificial intelligence.* Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 217–223.

[135] P. Kolan and R. Dantu, "Socio-technical defense against voice spamming," *ACM Trans. Auton. Adapt. Syst.*, vol. 2, no. 1, pp. 1–44, 2007.

[136] R. Dantu and P. Kolan, "Detecting spam in voip networks," in *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*. Berkeley, CA, USA: USENIX Association, 2005, pp. 5–5.

[137] J. Cheng, S. H. Wong, H. Yang, and S. Lu, "Smartsiren: virus detection and alert for smartphones," in *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2007, pp. 258–271.

[138] G. E. Smith and P. D. Berger, "The impact of direct marketing appeals on charitable marketing effectiveness," *Academy of Marketing Science*, vol. 24, no. 3, pp. 219–232, 1996.

[139] S. J. and R. Croson, "The impact of social influence on the voluntary provision of public goods," *Working paper*, 2005.

[140] ——, "The impact of social comparisons on nonprofit fundraising," *Forthcoming in Research in Experimental Economics*, 2005.

[141] ——, ""i" give, but "we" give more: The impact of identity and the mere social information effect on donation behavior," *Journal of Marketing Research*, 2005.

[142] S. Phithakkitnukoon and R. Dantu, "Mobile social closeness and similarity in calling patterns," in *Proceedings of the IEEE International Conference on Social Computing*, 2009.

[143] ——, "Predicting calls — new service for an intelligent phone," in *MMNS '07: Proceedings of the 10th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 26–37.

[144] C. A. Hidalgo and C. Rodriguez-Sickert, "The dynamics of a mobile phone network," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 12, pp. 3017–3024, May 2008.

[145] S. Phithakkitnukoon and R. Dantu, "Adequacy of data for characterizing caller behavior," in *Proceedings of SNAKDD 2008: KDD Workshop on Social Network Mining and Analysis, in conjunction with the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, 2008.

[146] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering, 2nd ed.* Prentice Hall, 1993.

[147] M. H. Magalhaes, R. Ballini, P. Molck, and F. Gomide, "Combining forecasts for natural streamflow prediction," in *Processing of NAFIPS '04: IEEE Annual Meeting of Fuzzy Information*, vol. 1, 2004, pp. 390–394.

[148] C. Guang, G. Jian, and D. Wei, "Nonlinear-periodical network traffic behavioral forecast based on seasonal neural network model," in *Proceedings of International Conference on Communications, Circuits and Systems, 2004 (ICCCAS 2004)*, 2004, pp. 683–687.

[149] W. Tycha, D. J. Pedregal, P. C. Young, and J. Davies, "An unobserved component model for multi-rate forecasting of telephone call demand: the design of a forecasting support system," *International jouornal of forecasting*, vol. 18, no. 4, pp. 673–695, 2002.

[150] H. J. V. and N. R. D., "Neural networks and traditional time series methods: a synergistic combination in state economic forecasts," *IEEE Transaction on neural network*, vol. 8, no. 4, pp. 863–873, 1997.

[151] N. Eagle and A. Pentland, "Eigenbehaviors: Identifying structure in routine," in *Proc. Roy. Soc. A (in submission)*, 2006.

[152] ——, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.

[153] R. Sheldrake and P. Smart, "Testing for telepathy in connection with e-mails," *Perceptual and Motor Skills*, vol. 10, pp. 771–786, 2005.

[154] R. Dantu and P. Kolan, "Survey of calling patterns," 2006. [Online]. Available: http://secnet.csci.unt.edu/nuisance/index.htm

[155] N. Eagle, "Machine perception and learning of complex social systems," in *Ph.D. Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology*, 2005.

[156] H. Husna, S. Phithakkitnukoon, E. Baatarjav, and R. Dantu, "Quantifying presence using calling patterns," in *Proceedings of COMSWARE 2008: IEEE International Conference on Communication Systems Software and Middleware and Workshops*, 2008, pp. 184–187.

[157] M. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal American Statistics Association*, vol. 433, no. 91, pp. 401–407, March 1996.

[158] M. P. Wand and M. C. Jones, "Multivariate plug-in bandwidth selection," *Computational Statistics*, vol. 9, pp. 97–117, 1994.

[159] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[160] J. Rosenberg and C. Jennings, "The Session Initiation Protocol (SIP) and Spam," *Internet informational RFC 5039*, 2008.

[161] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao, ""Statistical analysis of a telephone call center: a queueing science perspective"," Wharton Financial Institutions Center, Tech. Rep. 03-12, November 2002.

[162] S. Aldor-Noiman, "Forecasting demand for a telephone call center: Analysis of desired versus attainable precision," *Master Thesis*, 2006.

[163] H. Jasso, T. Fountain, C. Baru, W. Hodgkiss, D. Reich, and K. Warner, "Prediction of 9-1-1 call volumes for emergency event detection," in *dg.o '07: Proceedings of the 8th annual international conference on Digital government research.* Digital Government Society of North America, 2007, pp. 148–154.

[164] C. E. Harless and T. J. Kowalski, "System and method for correlating incoming and outgoing telephone calls using predictive logic," *U.S. Patent 6084954*, 2000.

[165] M. Weiser, "The computer for the 21st century," pp. 933–940, 1995.

[166] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *WMCSA '94: Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 1994, pp. 85–90.

[167] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, pp. 22–32, 1994.

[168] M. H. Coen, "Design principles for intelligent environments," in *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 547–554.

[169] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz, "An architecture for a secure service discovery service," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM, 1999, pp. 24–35.

[170] A. Friday, N. Davies, and E. Catterall, "Supporting service discovery, querying and interaction in ubiquitous computing environments," in *MobiDe '01: Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*. New York, NY, USA: ACM, 2001, pp. 7–13.

[171] H. Chen, A. Joshi, and T. Finin, "Dynamic service discovery for mobile computing: Intelligent agents meet jini in the aether," *Cluster Computing*, vol. 4, no. 4, pp. 343–354, 2001.

[172] F. Zhu, M. Mutka, and L. Ni, "Research on context-aware mobile computing," in *Splendor: A Secure, Private, and Location-Aware Service Discovery Protocol Supporting Mobile Services*. Fort Worth, TX, USA: Proceedings of the First IEEE Ann. Conf. Pervasive Computing and Comm. (PerCom03), March 2003, pp. 1–8.

[173] S. Chetan, J. Al-Muhtadi, R. Campbell, and M. Mickunas, "A middleware for enabling personal ubiquitous spaces," in *System Support for Ubiquitous Computing Workshop at Sixth Annual Conference on Ubiquitous Computing (UbiComp 2004)*, September 2005.

[174] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile context-aware service access," *Comput. Commun.*, vol. 31, no. 5, pp. 935–949, 2008.

[175] K.-L. Park, U. H. Yoon, and S.-D. Kim, "Personalized service discovery in ubiquitous computing environments," *IEEE Pervasive Computing*, vol. 8, no. 1, pp. 58–65, 2009.

[176] E. Paulos and E. Goodman, "The familiar stranger: anxiety, comfort, and play in public places," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2004, pp. 223–230.

[177] B. Davis and K. Karahalios, "Telelogs: a social communication space for urban environments," in *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*. New York, NY, USA: ACM, 2005, pp. 231–234.

[178] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software." *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.

[179] A. Oulasvirta, M. Raento, and S. Tiitta, "Contextcontacts: re-designing smartphone's contact book to support mobile awareness and collaboration," in *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*. New York, NY, USA: ACM, 2005, pp. 167–174.

[180] L. Buriano, "Exploiting social context information in context-aware mobile tourism guides," in *Mobile Guide 2006*. ACM, 2006.

[181] V. Kostakos, E. O'Neill, and A. Shahi, "Building common ground for face to face interactions by sharing mobile device context." in *LoCA*, ser. Lecture Notes in Computer Science, M. Hazas, J. Krumm, and T. Strang, Eds., vol. 3987. Springer, 2006, pp. 222–238.

[182] M.-S. Jian, K. S. Yang, and C.-L. Lee, "Context and location aware public/personal information service based on rfid system integration," *WTOS*, vol. 7, no. 6, pp. 774–784, 2008.

[183] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong, "Sensay: A context-aware mobile phone," in *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2003, p. 248.

[184] A. Krause, D. P. Siewiorek, A. Smailagic, and J. Farringdon, "Unsupervised, dynamic identification of physiological and activity context in wearable computing," in *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2003, p. 88.

[185] B. Adams, D. Phung, and S. Venkatesh, "Extraction of social context and application to personal multimedia exploration," in *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2006, pp. 987–996.

[186] H. Husna, S. Phithakkitnukoon, E.-A. Baatarjav, and R. Dantu, "Quantifying presence using calling patterns." in *COMSWARE*. IEEE, 2008, pp. 184–187.

[187] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive*, 2004, pp. 1–17.

[188] U. Munetoshi, K. Ken'ichiro, T. Atsuya, and H. Takeshi, "Autonomy position detection by using recognition of human walking motion," in *IEICE Japan*, vol. J87-A, no. 1, 2004, pp. 78–86.

[189] S. Koichi, "Measurement and analysis of human behavior using wearable sensors," in *Proceedings of The 25th Asian Conference on Remote Sensing*, vol. 2, 2004, pp. 1218–1223.

[190] M. Kourogi and T. Kurata, "Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera," in *ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA: IEEE Computer Society, 2003, p. 103.

[191] X. Liu, "System support for pervasive multimedia systems," Ph.D. dissertation, 2006, adviser-Shenoy,, Prashant and Adviser-Corner,, Mark D.

[192] K. Bernardin and R. Stiefelhagen, "Audio-visual multi-person tracking and identification for smart environments," in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*. New York, NY, USA: ACM, 2007, pp. 661–670.

[193] J. Suutala and J. Röning, "Methods for person identification on a pressure-sensitive floor: Experiments with multiple classifiers and reject option," *Inf. Fusion*, vol. 9, no. 1, pp. 21–40, 2008.

[194] P. W. Grosse, H. Holzapfel, and A. Waibel, "Confidence based multimodal fusion for person identification," in *MM '08: Proceeding of the 16th ACM international conference on Multimedia*. New York, NY, USA: ACM, 2008, pp. 885–888.

[195] K. V. Laerhoven and O. Cakmakci, "What shall we teach our pants?" in *ISWC '00: Proceedings of the 4th IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2000, p. 77.

[196] J. Lester, B. Hannaford, and G. Borriello, ""are you with me?" - using accelerometers to determine if two devices are carried by the same person," in *Pervasive*, 2004, pp. 33–50.

[197] P. Lukowicz, J. Ward, H. Junker, M. Stager, G. Troster, A. Atrash, and T. Starner, "Recognizing workshop activity using body worn microphones and accelerometers," 2004.

[198] T. Kohonen, *Self-Organizing Map, Third Ed.* Newyork, USA: Springer, 2001.

[199] G. H. Jin, S. B. Lee, and T. S. Lee, "Context awareness of human motion states using accelerometer," *J. Med. Syst.*, vol. 32, no. 2, pp. 93–100, 2008.

[200] T. Iso and K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," in *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. New York, NY, USA: ACM, 2006, pp. 141–144.

[201] J. S. Yi, Y. S. Choi, J. A. Jacko, and A. Sears, "Context awareness via a single device-attached accelerometer during mobile computing," in *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices*

& services. New York, NY, USA: ACM, 2005, pp. 303–306.

[202] Governors Highway Safety Association, "Cell phone driving laws," March 2009. [Online]. Available: http://www.ghsa.org/html/stateinfo/laws/cellphone_laws.html

[203] T-Mobile, "T-mobile's g1 phone," April 2009. [Online]. Available: http://www.t-mobileg1.com/

[204] Android, "Android developers," April 2009. [Online]. Available: http://developer.android.com/

[205] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, October 2002.

[206] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)*. The MIT Press, 2006.

[207] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[208] A. de Cheveigné and J. Z. Simon, "Denoising based on time-shift pca," *Journal of neuroscience methods*, vol. 165, no. 2, pp. 297–305, September 2007.

[209] C. Lo, D. D. Turner, and R. O. Knuteson, "A principal component analysis noise filter value-added procedure to remove uncorrelated noise from atmospheric emitted radiance interferometer (aeri) observations," 2006.

[210] Y. M. Jung, "Principal component analysis based two-dimensional (pca-2d) correlation spectroscopy: Pca denoising for 2d correlation spectroscopy," *Bull. Korean Chem. Soc.*, vol. 24, no. 9, pp. 1345–1350, 2003.

[211] M. M. Rahman, K. Nakamura, and S. Ishikawa, "Recognizing human behavior using universal eigenspace," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*. Washington, DC, USA: IEEE Computer Society, 2002, p. 10295.

[212] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*. New York, NY, USA: ACM,

2003, pp. 303–316.

[213] N. Moënne-loccoz, F. Brmond, and M. Thonnat, "Recurrent bayesian network for the recognition of human behaviors from video," in *ICVS*, 2003, pp. 68–77.

[214] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 195–205, 2002.

[215] G. Resta and P. Santi, "The qos-rwp mobility and user behavior model for public area wireless networks," in *MSWiM '06: Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems.* New York, NY, USA: ACM, 2006, pp. 375–384.

[216] W. Tycha, D. J. Pedregal, P. C. Young, and J. Davies, "An unobserved component model for multi-rate forecasting of telephone call demand: the design of a forecasting support system," *International Journal of Forecasting*, vol. 1, pp. 683–687, 2004.

[217] W.-T. Fu and P. Pirolli, "Snif-act: a cognitive model of user navigation on the world wide web," *Hum.-Comput. Interact.*, vol. 22, no. 4, pp. 355–412, 2007.

[218] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Comput.*, vol. 11, no. 1, pp. 229–242, 1999.

[219] C. Chatfield, *The Analysis of Time Series: An Introduction, Sixth Edition.* Chapman & Hall/CRC, 2004.

[220] M. Fannes and P. Spincemaille, "The mutual affinity of random measures," *eprint arXiv:math-ph/0112034*, 2001.

[221] D. Pollard, *Asymptopia, 1st edition.* http://www.stat.yale.edu/pollard/, 2000.

[222] C. shing Perng, H. Wang, S. R. Zhang, and D. S. Parker, "Landmarks: A new model for similarity-based pattern querying in time series databases," in *Proceedings of Inter. Conf. Data Engineering (ICDE)*, 2000, pp. 33–42.

[223] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2000.

[224] K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM Trans. Database Syst.*, vol. 27, no. 2, pp. 188–228, 2002.

[225] T. chung Fu, F. lai Chung, V. Ng, and R. Luk, "Pattern discovery from stock time series using self-organizing maps," in *Notes KDD2001 Workshop on Temporal Data Mining)*, 2001, pp. 27–37.

[226] E. Fink, K. B. Pratt, and H. S. Gandhi, "Indexing of time series by major minima and maxima," in *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, 2003, pp. 2332–2335.

[227] U. Y. Ogras and H. Ferhatosmanoglu, "Dimensionality reduction using magnitude and shape approximations," in *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management.* New York, NY, USA: ACM, 2003, pp. 99–107.

[228] Y. Zhao and S. Zhang, "Generalized dimension-reduction framework for recent-biased time series analysis," *IEEE Trans. on Knowl. and Data Eng.*, vol. 18, no. 2, pp. 231–244, 2006, senior Member-Zhang, Shichao.

[229] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang, "Multi-dimensional regression analysis of time-series data streams," in *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases.* VLDB Endowment, 2002, pp. 323–334.

[230] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, "Mining frequent patterns in data streams at multiple time granularities," pp. 191–210, 2003.

[231] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases.* VLDB Endowment, 2003, pp. 81–92.

[232] A. Bulut and A. Singh, "Swat: Hierarchical stream summarization in large networks," in *Proceedings of Inter. Conf. on Data Engineering*, 2003, pp. 303–314.

[233] D. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, C. Erwin, E. Galvez, M. Hatoun, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing,

R. Yan, and S. Zdonik, "Aurora: a data stream management system," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2003, pp. 666–666.

[234] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, F. Reiss, and M. A. Shah, "Telegraphcq: continuous dataflow processing," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2003, pp. 668–668.

[235] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk, "Gigascope: a stream database for network applications," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2003, pp. 647–651.

[236] D. Madigan, "Dimacs working group on monitoring message streams," 2003. [Online]. Available: http://stat.rutgers.edu/∼madigan/mms/

[237] I. K. Fodor, "A survey of dimension reduction techniques," Livermore, CA, USA, Tech. Rep., 2002.

[238] R. Agrawal, C. Faloutsos, and A. N. Swami, "Efficient similarity search in sequence databases," in *FODO '93: Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. London, UK: Springer-Verlag, 1993, pp. 69–84.

[239] "On similarity-based queries for time series data," in *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 1999, p. 410.

[240] W.-K. Loh, S.-W. Kim, and K.-Y. Whang, "A subsequence matching algorithm that supports normalization transform in time-series databases," *Data Min. Knowl. Discov.*, vol. 9, no. 1, pp. 5–28, 2004.

[241] K. W. Chu and M. H. Wong, "Fast time-series searching with scaling and shifting," in *PODS '99: Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.* New York, NY, USA: ACM, 1999, pp. 237–248.

[242] C. Burus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer.* Prentice-Hal Inc., 1998.

[243] C. Berberidis, W. G. Aref, M. Atallah, I. Vlahavas, and A. Elmagarmid, "Multiple and partial periodicity mining in time series databases," in *Proceedings of 15th European Conference on Artificial Intelligence*, 2002, pp. 370–374.

[244] M. Elfeky, W. Aref, and A. Elmagarmid, "Using convolution to mine obscure periodic patterns in one pass," in *Proceedings of the 9th International Conference on Extending Database Technology (EDBT04.* Springer, 2004, pp. 605–620.

[245] M. G. Elfeky, "Periodicity detection in time series databases," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 7, pp. 875–887, 2005, senior Member-Aref, Walid G. and Senior Member-Elmagarmid, Ahmed K.

[246] P. Indyk, N. Koudas, and S. Muthukrishnan, "Identifying representative trends in massive time series data sets using sketches," in *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 363–372.

[247] S. Ma and J. L. Hellerstein, "Mining partially periodic event patterns with unknown periods," in *Proceedings of the 17th International Conference on Data Engineering.* Washington, DC, USA: IEEE Computer Society, 2001, pp. 205–214.

[248] J. Yang, W. Wang, and P. S. Yu, "Mining asynchronous periodic patterns in time series data," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD*, 2000, pp. 275–279.

[249] G. L. Yang and L. M. L. Cam, *Asymptotics in Statistics: Some Basic Concepts.* Berlin, German: Springer, 2000.

[250] F. Mörche, "Time series feature extraction for data mining using dwt and dft," Marburg, Germany, Tech. Rep. 33, 2003.

[251] Y. Zhao, C. Zhang, and S. Zhang, "A recent-biased dimension reduction technique for time series data," in *Proc. of PAKDD05. Accuracy.* Springer, 2005, pp. 75–757.

[252] K. W. Hipel and A. I. McLeod, *Time Series Modelling of Water Resources and Environmental Systems.* Elsevier Science B.V., 1994.

[253] S. Makridakis, S. Wheelwright, and R. Hyndman, *Forecasting: Methods and Applications (3rd ed).* Wiley, 1998.

[254] R. J. Hyndman, "Time series data library," February 2009. [Online]. Available: http://www.robhyndman.info/TSDL