RICE UNIVERSITY

Context in Mobile System Design: Characterization, Theory, and Implications

by

Ahmad Rahmati

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

DISSERTATION COMMITTEE

Lin Zhong, Chair
Associate Professor
Electrical and Computer Engineering

Edward Knightly
Professor
Electrical and Computer Engineering

Ashutosh Sabharwal
Associate Professor
Electrical and Computer Engineering

Philip Kortum
Professor in the Practice, Faculty Fellow
Psychology

HOUSTON, TEXAS
APRIL 2012

Abstract

Context in Mobile System Design: Characterization, Theory, and Implications

by

Ahmad Rahmati

Context information brings new opportunities for efficient and effective applications and services on mobile devices. Many existing work exploit the context dependency of mobile usage for specific applications, and show significant, quantified, performance gains by utilizing context. In order to be practical, such works often pay careful attention to the energy and processing costs of context awareness while attempting to maintain reasonable accuracy. These works also have to deal with the challenges of multiple sources of context, which can lead to a sparse training data set.

Even with the abundance of such work, quantifying context-dependency and the relationship between context-dependency and performance achievements remains an open problem, and solutions to manage the challenges of context awareness remain ad-hoc. To this end, this dissertation methodologically quantifies and measures the context dependency of three principal types of mobile usage in a methodological, application agnostic yet

practical manner. The three usages are the websites that users visit, the phone numbers they call, and the *apps* (i.e. purchased or preinstalled applications) they use. While this dissertation measures the context dependency of these three usage, its methodology can be readily extended to other context-dependent mobile usage and system resources. This dissertation further presents SmartContext, a framework to systematically optimize the energy cost of context awareness by selecting among different context sources, while satisfying system designers' cost-accuracy tradeoffs. Finally, this thesis investigates the collective effect of social context on mobile usage, by separating and comparing LiveLab users based on their socioeconomic status.

The analysis and findings are based on usage and context traces collected in real-life settings from 24 iPhone users over one year. This dissertation presents findings regarding the context dependency of three principal types of mobile usage; visited websites, phone calls, and app usage. The methodology and lessons presented here can be readily extended to other forms of context and context-dependent usage and resources. They guide the development of context aware systems, and highlight the challenges and expectations regarding the context dependency of mobile usage.

Acknowledgments

First and foremost, I would like to thank my advisor, Lin Zhong, who has been a model researcher and has provided endless resources that enabled me to carry out successful research throughout my graduate studies. I would also like to thank the members of my thesis committee, Edward Knightly, Ashutosh Sabharwal, and Philip Kortum, for their insightful suggestions and comments. I am grateful to numerous other Rice faculty members as well, who I have learned a lot from. Special thanks go to the department chair, Behnaam Aazhang, for his leadership and support, and to Farinaz Koushanfar for her helpful discussions and insights. The research presented in this thesis would have been impossible without the LiveLab data set, which has been a collaborative, multidisciplinary effort. I would like to recognize fellow graduate students Clayton Shepard and Chad Tossell in this regard.

I am in debt to many great friends and teachers. In fact, I would have a hard time attempting to distinguish some of my teachers from friends. Special thanks to my teachers at Sharif and Allameh Helli, and to my very dear friends throughout my life. Finally, I cannot even begin to express enough gratitude towards my wonderful family, and in particular my parents, for their continuous love, care, and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern mobile systems such as smartphones and tablets are already important part of human lives. They are not only computationally powerful but also have a rich capability to sense their external and internal environment. Similar to the definition by Schilit et al. in [1], this dissertation refers to the last known condition of these environments collectively as *context*. Context dependency can be broadly defined as a set of strict or probabilistic rules and relations between context(s) and the outcome [2].

Context has in the past been widely exploited to provide more usable mobile devices and services, such as through content adaptation [3, 4], user interaction [5], and information delivery [6, 7]. Context has also been widely exploited to provide enhanced system efficiency and performance, such as for energy management [8, 9] and network selection [10]. These designs exploit the context dependency of mobile usage and/or mobile resources for specific purposes, and show significant, quantified, performance gains. The usefulness of context has been so significant that many researchers have designed and implemented frameworks for the specific task of sensing and processing context [11-13].

Context aware systems often have to deal with two fundamental challenges. First: dealing with multiple sources of context is challenging; due to the curse of dimensionality [14], simply treating them as a multidimensional vector results in a sparse training set. Second: liberal application of context can quickly drain the devices battery, as some context sensors are extremely energy hungry. To address the sparseness challenge, existing work often limit the number of context sources, e.g. to one [8] or two [9], and/or employ ad-hoc or expert solutions to combine multiple sources of context, e.g. [10]. To address the energy challenge, they often employ ad-hoc schemes along one or more of these lines: reducing the frequency of accessing costly context [15-17], avoiding them altogether [10, 15, 16], or substituting them with other context [18-21].

While many existing work exploit the context dependency of mobile usage for specific purposes, and show significant, quantified, performance gains by utilizing context, Ad-hoc and application specific approaches towards the challenges of context awareness mean that the designers need to design and evaluate a new solution for every context-based system. Furthermore, existing work stop short of quantifying context-dependency itself, and there is no clear relationship between their performance achievements and context-dependency. This dissertation seeks to liberate the mobile system designer from these challenges, by presenting a formal yet practical definition of context dependency, which provides insight into the performance of context-aware applications and services while remaining application agnostic. This dissertation presents a methodological approach to the challenges of using different sources of context, while managing their energy costs by selecting among different context sources while satisfying the system designer's cost-accuracy tradeoffs. This work measures the context dependency of three principal types of mobile usage, using unprecedented real-life context and usage traces collected from 24 iPhone users over one year. The mobile usages are websites users visit, the phone calls they make, and the apps they launch[1]. The context information is obtained from sensors built into the phone (i.e. real-time clock, Cell ID, Accelerometer, and GPS), as well as the phone's last known usage states (i.e. web, phone, and app use).

These findings not only guide the development of future context aware systems that depend on these principal types of usage, but they highlight the expectations one should have regarding the relationship between context and mobile usage. Furthermore, the methodology and lessons learned can be extended to other forms of context, and context-dependent usage and/or system resources as well.

Studying context dependency can be extremely challenging, as it needs a large trace collected in real life user studies. This is enabled by the LiveLab methodology, which addresses the technical and human factors that previously limited user studies of this duration and scope, enabling us to collect unprecedented usage and context data from 24 iPhone users continuously for one year. A key component of LiveLab is a reprogrammable

[1] Note that this dissertation uses the word *app* and *application* differently. App is used to refer to applications that are installed on the phones, either built-in or obtained from the App Store. Application is used to refer to its more general meaning, i.e. use case. Similarly, application agnostic refers to multipurpose, not dedicated to a single service or purpose.

in-device logger designed for long-term field studies with minimum energy, usability and privacy intrusions. While extensive logging of PC usage has been reported in past literature, privacy concerns and more importantly battery lifetime have limited the scope of mobile phone based studies in the past. LiveLab overcomes these challenges by its careful design and implementation, and demonstrates that such logging, if done properly, is indeed possible on modern mobile phones.

Towards quantifying the context dependency of mobile usage and optimizing its cost, this dissertation makes five technical contributions:

First, it identifies *estimation accuracy* based on maximum a posteriori (MAP) estimation as an application agnostic yet practical measure for context dependency. In contrast with other theoretical metrics that are applicable applied to multinomial data, such as entropy as a measure of uncertainty and pseudo R square as a measure of correlation, estimation accuracy provides practical insight into the performance of many potential applications, while remaining application agnostic. To allow the efficient calculation of posterior probabilities, the performance of several methods of categorizing and binning context measurements into a limited number of categories are presented, for both continuous and discrete context sources. Furthermore, the challenge of data sparseness when dealing with multiple sources of context is addressed by presenting and comparing several prominent classifier combination methods.

Second, using the LiveLab data set, this dissertation presents a series of interesting findings regarding the context dependency of mobile usage, as follows: 1) The effectiveness of different context varies based on the usage to be estimated, as well as the number of accepted responses. Yet, combining multiple sources of context uncovers their combined strength. 2) Even though multiple context sources are dependent, Bayesian Combination performs well for combining context information. 3) The context-dependency of usage remains relatively constant even for durations of one to three months, instead of the full 12 months. This indicates that a smaller data set would be sufficient for context-awareness. 4) Supervised Binning can greatly increase estimation accuracy by keeping a large number of samples in each category or bin, while allowing fine molding of the bins. 5) Even though users are diverse in their usage, substantial context dependency can be observed among all of them.

Third, this dissertation presents SmartContext, a framework to dynamically or statically optimize the cost-accuracy tradeoffs of context awareness, while ensuring a minimum accuracy for each estimation event. SmartContext takes advantage of the classifier combination algorithms explored in Chapter 4.2.2

that have little overhead, starting with context information that is less costly. SmartContext moves on to more costly context sources only if they are necessary to meet the system designer's cost-accuracy tradeoff. It is shown that by utilizing energy hungry context only at uncertain times, SmartContext can achieve an estimation accuracy within 1% of the maximum possible accuracy, while significantly reducing energy costs, by 60% or more.

Fourth, this dissertation presents and evaluates several sample applications that benefit from context dependency of mobile usage. These applications highlight the practical value of estimation accuracy as a measure of context dependency, and attest to the effectiveness of context for estimating usage. The best performing methods presented here, i.e. using Supervised Binning and Bayesian combination, consistently outperform common non-context-based methods.

Fifth, this thesis investigates a different form of context, social context, to measure its effect on mobile usage. To this end, the LiveLab users are separated into two carefully selected, distinct, socioeconomic groups, and their collective usage patterns are analyzed and compared. These findings provide researchers and system designers with a better understanding of how users vary, resulting in better support of a broader range of individuals with different backgrounds, capabilities, skills and interests.

Chapter 2

Related Work

Prior literature (e.g. [2]) refers to context dependency as a set of strict or probabilistic rules and relations between context(s) and the outcome. Context information itself has been referred to knowledge regarding the location and identity of the user and its surroundings [22, 23]. A more accurate and broad definition of context information is given by Dey and Abowd, as “any information that can be used to characterize the situation of an entity” [24].

Context information has in the past been widely used for specific applications and system mechanisms. They depend on the context dependency of device usage and resources, and show significant, quantified, performance gains by exploiting context. The usefulness of context has been so significant that many researchers have designed and implemented frameworks for the specific task of sensing and processing context, and for determining user state from context information. Each of these is covered in detail in this section.

These designs and others depend on the context dependency of device usage and resources, and show significant, quantified, performance gains by exploiting context. This dissertation picks up where the above efforts stop short; it presents a formal definition of context dependency, as well as practical methods to calculate it, using one or multiple sources of continuous or discrete context. This dissertation abstains from focusing on a single application, yet provides practical insight into the relationship between context-dependency of three principal types of mobile usage, i.e. website, phone call, and app usage, and performance achievements of individual applications and services.

2.1. Applications and services utilizing context-awareness

Context information has in the past been widely used for specific applications and services. Such efforts can typically be grouped into several categories, as below. These works often use context information to estimate certain usage, resources, or conditions, which can collectively be called as outcome variables. Such work attest to the usefulness of context for various applications and services, and show significant, quantified, performance gains. Yet, these works often do not provide much insight into how context-dependent is their outcome variable. Even worse, the close-to-ideal performance gains of such applications might create a false illusion that their dependent variable is extremely context dependent. In contrast, this dissertation provides an in depth analysis on quantifying context-dependency itself, in particular for three principal types of mobile usage, i.e. website, phone call, and app usage.

2.1.1. **Context-based adaptation**

Context information is often used to adapt application contents and device settings. Location based services are one of the most popular context-aware applications that adapt their content and elements according to context. A tourist guide is probably the most researched of location based applications. In [7], Cheverest et al. present and evaluate a such an application that provides navigation tools as well as traditional tourist information. In [25], Schwinger et al. survey a number of tourist guide applications. Location awareness enables many other applications as well. For example, in [6], Marmasse and Schmandt present comMotion, a location based reminder system that provides user-defined and/or predefined content according to the device's location. In [26], Chen and Kotz survey many other, older, applications, such as a virtual shopping assistant application and infrastructure from Asthana et al. [27] in the early 1990s that utilizes the customers location within a store.

Context-aware application and services that adapt their content and elements according to context are not limited to location-based applications. For example, in [3], Lemlouma et al. present semantics that allow the adaptation of web content according to the capabilities and context of the mobile device. As another example, in [4], Kane et al. adapt the size of user interface elements according to the user's motion, choosing larger elements when the user is moving.

Adapting device settings is another application of context information. Examples include [28], by Schmidt et al., which detects the phone's state, e.g. in hand, outside, on a table, and adjusts the ringer volume and vibration alert accordingly. The Smart Actions functionality of the newly released Motorola Droid Razr phone [29] is a commercial example. It provides built-in support for such adaptations; the user can specify certain settings to be changed, or apps to be run, automatically according to certain context, such as location and time.

2.1.2. **Context-based system services**

Context-based system services are often geared towards improving the efficiency or performance of the mobile device, either automatically or by providing information or mechanisms to individual applications. The most common of such work focuses on network services, taking appropriate actions based on context, such as selecting the best wireless interface, route, or time for data transfers. Most of the research focusing on predicting network associations and/or conditions utilizes only same network conditions as the predictor (context). For example, in [30], Lee and Hou utilize semi-Markov models to predict the Wi-Fi access points users will associate to, and the duration they will remain associated, based on their history of Wi-Fi associations. They show that utilize this predictions, it is possible to load balance among access points, reducing the maximum access point load of their sample traces (the association history of WLAN users at Dartmouth college [31]) by three fold.

Nicholson and Noble also utilize Markov models [32], to predict short term (10s of seconds) Wi-Fi performance according to location indicators derived from visible Wi-Fi access points. They evaluate their system for streaming media, and uploading delay tolerant content.

Still, other work focus on estimating and predicting current and future Wi-Fi network conditions without probing the Wi-Fi network, using other context as predictors. In particular, the work by Rahmati and Zhong show that selecting between cellular and Wi-Fi networks using estimated Wi-Fi network conditions instead of measuring the conditions can significantly improve device energy efficiency [33]. Using their field study collected traces, they showed that context information such as time, visible cellular towers, and motion, can produce energy savings up to 90% of the ideal case for data transfers typical of a health monitoring application. The ideal case is defined as if the device knew Wi-Fi conditions without any energy cost. Furthermore, they showed that it is possible to predict Wi-Fi availability for one and ten hours into the future with 95% and 90% accuracy [10].

There have been many other system services, beyond predicting network conditions, presented by the research community as well. For example, Cugola and Migliavacca in [34] show that context information regarding mobile users, such as the location and movement, can be utilized instead of fixed addresses to route packets.

Other researchers have focused on using context information to manage the energy characteristics of mobile devices. For example, in [35] and [36], Rahmati et al. studied how smartphone users interact with their limited battery capacity of their devices, and found, among other things, that many recharges happen according to the user's context, with significant remaining battery, challenging the conventional wisdom that a longer battery lifetime is always desired. These finding motivated the design and implementation of context-driven energy management in [8], by Banerjee, Rahmati, et al, where the phone adapts its energy management policies based on each user's charging behaviors and remaining battery levels. It takes advantage of the often unused battery capacity to increase the performance and usability of the mobile device. Similarly, in [9], Ravi et al. study the use of a wider range of context information for adapting the energy management policies of mobile phones.

2.1.3. **Determining user state using context**

There has also been considerable research on determining user state from context information. While device usage can clearly be influenced by the user's state, this dissertation abstains from extracting user state as an interim stage, and instead focuses on the direct relationship between device usage and context information.

One example of extracting user state from context information include extracting physical activity, such as the work by Bouten et al. in [37], where they extract daily physical activity from accelerometer readings. Another recent example if the work by LiKamWa et al. in [38] where they extract the users emotional mood in terms of pleasure and activeness from context information, with over 90% accuracy compared to self reported mood.

User state can be itself be the goal, or may be further utilized for *implicit* user interaction with computers. Implicit user interaction, as defined by Schmidt in [5], as input based on users actions that are not primarily aimed to interact with the computer. Accordingly, Schmidt champions for an XML language to enable implicit interaction with computers. Schmidt also presents a sample app, a context aware notepad that adapts its display according to a user's context, at times increasing the contrast, font size, and/or hiding the text for privacy.

2.1.4. **Temporal analysis of context**

There has been considerable research on utilizing a time series (stream) of context, often using semi-Markov models. The stream of context is often used to predict the next state of the context itself, or the future usage or state of another system resource. Typically, these efforts can be seen as extracting hidden states of the user's mind based on a history of context. This dissertation is orthogonal to these efforts, focusing on the current context, defined as the last known context of the user, irrespective of prior context states.

The most common of such work focuses on wireless networks. For example, in [30], Lee and Hou utilize semi-Markov models to predict the next Wi-Fi access points users will associate to, and the duration they will remain associated, based on their history of Wi-Fi associations. Similarly, in [39], Laasonen predicts the order of future cellular associations using semi-Markov models. Along the same lines, in [40], Rathnayake et al. compare semi-Markov models with Dynamic Bayesian Networks (DBN), and show that DBN outperforms semi-Markov models by 20% for the prediction of Wi-Fi availability five minutes into the future.

As network conditions and location are strongly correlated, many other similar work focus on the intersection of mobile user mobility models and predicting future wireless network associations. In [41], Song et al. focus on methods to predict the order of appearance of Wi-Fi access points (irrespective of actual time) using the same traces, and show the efficacy of semi-Markov models.

Using Markov models has its drawbacks, for example it requires exponential processing when predicting the time for future events [42]. Consequently, alternatives to Markov models have been proposed and evaluated for the prediction of stream states. In particular, Peddemors et al. in [43] offer a through treatment of that area. In that work, the authors calculate the conditional probability of events, in particular the time and access point of future Wi-Fi associations, according to sensor (context) predictors. Note that processing large stream data sets of higher orders of depths is itself a challenge. The work in [44] by Aggarwal et al. addresses this issue, and provides an in-depth analysis.

2.2. System implications of context-awareness

2.2.1. **Frameworks for sensing and processing context**

The usefulness of context has been so significant that many researchers have designed and implemented frameworks for the specific task of sensing and processing context. These work attest to the significance of context. One such example is the work by Kranenburg et al. in [11], where they present a context-processing framework that offers a single heterogeneous interface to multiple, hierarchic context sources. ContextPhone by Raento et al. [45] also focuses on simplifying the development of context-aware applications and services by providing a single, uniform interface to context information available on Symbian smartphones. Another example is the work by Wood et al. in [46], where they present a context processing middleware tailored for networked entities, such as in sensor networks. Their protocols allow distributed queries while supporting power management. Another more recent example is the work by Lee et al. [12], where they present a generic framework for both gathering and processing context information from a variety of sensors. For a survey of frameworks for sensing and processing context information, see Baldauf et al., “A survey on context-aware systems” [13].

2.2.2. **Energy efficiency in sensing context**

A number of recent research have focused on reducing the cost of acquiring context. These work attest to the challenge of energy efficiency in context awareness, but typically focus on single applications and/or static configurations. They use one or more of the following three techniques to reduce energy cost, while retaining acceptable performance. First, *frequency reduction* reduces the sampling frequency of energy hungry context sensors. Second, *sensor substitution* utilizes lower energy cost context instead of energy hungry ones. Third, *sensor elimination* attempts to use a subset of sensors. Chapter 6 of this dissertation takes the third approach, but provides a generic framework for system designers to dynamically or statically optimize the cost-accuracy tradeoffs of context awareness.

To this end, in [17], Wang et al. attempt to reduce the frequency of sensing for phone motion detection using accelerometer readings, and for discovering nearby phones. They show that while user state changes are not fully Markovian, by assuming such and using an optimal Markov policy, they can reduce sampling frequency by 20% compared to uniform sampling.

Both Zhuang et al. in [15] and Paek et al. in [16] propose to reduce the energy cost of GPS localization by reducing the frequency of GPS sensor readings as well as employing network based positioning. Additionally, Paek et al. utilize network localization to detect when the GPS is unable to acquire a lock, and disable it entirely in those situations to avoid wasting energy. On the other hand, Zhuang et al. utilize the accelerometer to detect device motion, disabling the GPS when no motion is detected.

A number of recent researches have focused on sensor on sensor elimination for specific applications. For human physical activity detection, typically using a number of accelerometers placed on various parts of the body, in [18], Panuccio et al. attempt to reduce the cost of using sensors with different but fixed costs. They sequentially eliminate sensors with a high cost vs. weight for a specific activity the system is interested in classifying, while maintaining a preset average performance, and show a different subset of sensors are sufficient depending on what activity the system wants to detecting. In [19], Zappi et al. also automatically select a subset of accelerometers, but attempt to minimize the number of utilized accelerometers instead. They can select a different set of sensors depending on the activity the system wants to detect, and can cope with a changing available pool of accelerometers to account for battery depletion , etc. In [20], Kang et al. attack a similar problem, but instead of focusing on one activity of interest, use a state transition model for human activities and only enable sensors that are necessary for identifying transitions out of the human's current state. For the purpose of user state and activity detection using a wider range of sensors available on mobile devices, in [21], Wang et al. present a framework for energy efficient, hierarchical sensor access while maintaining high accuracy. In contrast to the application specific approaches employed in the above work, and their static sensor selection for each classification event, this desertion methodologically addresses a more general and fundamental problem for energy efficient context awareness, to dynamically inform the system designer of the expected cost-accuracy tradeoffs of classification events, and allow them to dynamically choose their tradeoffs requirements, even during each classification event.

A more general problem of selecting the most effective subset of sensors, also referred to as observations or predictors, has been the focus of decades of research in the artificial intelligence and operations research communities. These work focus on the optimization of information, typically defined as either joint entropy or information gain (delta entropy). It is common practice to use the greedy (myopic) solution towards this selection problem [47-50], due to the processing complexities of finding the optimal solution [51-53]. Fortunately, based on the submodularity property of the information gain from multiple observations [54], existing work prove that greedy solutions are near-optimal, with provable constant factor performance guarantees for both unit cost [54] and different cost sensors [47]. SmartContext, in Chapter 6, is built upon the greedy solution provided by Krause et al. [47], but is adapted according accordingly, in particular using estimation accuracy instead of information gain.

2.3. Studying Mobile Usage in-situ

2.3.1. **Phone Logging**

There has been several research reported in literature that has focused on, or utilized phone logging, Compared to LiveLab, they collect very limited context information, or can only be used for shorter studies, due to privacy concerns and battery lifetime limitations that LiveLab overcomes.

In [55], Froehlich et al. present MyExperience, a framework for collecting sensor and survey data on Windows Mobile devices. The framework supports collecting data and in-situ surveys according to context triggers, such as time and phone use, through an XML interface. They focus on evaluating the efficacy of MyExperience for in-situ surveys, and show a battery life reduction of 12% with 20 surveys per day without collecting energy consuming context data such as from Bluetooth and GPS sources. A more limited form of in-situ surveys have been proposed earlier by Intelle et al. in [56]. Banerjee, Rahmati, et al. have used in situ surveys in field studies in [8]. In contrast, and orthogonal to those efforts, LiveLab focuses on enabling long term field logging with an unprecedented amount of context information, possibly reducing the need for in-situ survey data for some studies.

The Reality Mining project, presented in [57] by Eagle and Pentland is one of the largest phone data logging efforts to date, covering 100 Symbian phones over the course of nine months. Reality Mining is in part based on ContextPhone [45], described prior, and periodically records call logs, visible cell IDs, visible Bluetooth devices, and charging status. The same authors later show that device usage patterns are indeed structured and predictable [58].

In [59], Falaki et al. deploy a custom logging utility on the phones of 33 Android users, capturing detailed usage data regarding screen status (on/off), app usage, network traffic, and energy drain. They also use another, closed, dataset from 222 Windows Mobile users containing screen status and app usage only. They find that mobile usage is indeed extremely diverse. However, lacking further information, and in particular context measurements, they are unable to extract order among the diversity or capture the context dependency of mobile usage.

In [60], Rahmati et al. present a four month field study with 14 participants from Pecan Park, an underserved neighborhood of Houston, TX. The Pecan Park Study was based on a prior pilot study with the same phone and logger [61]. The logging recorded comprehensive Wi-Fi, cellular, and battery traces, but only minimal usage information for privacy reasons. Instead, the authors relied on qualitative self reporting through focus groups for usage information. The Pecan Park study further highlighted the challenges and importance of long-term, holistic studies of mobile devices and services under real-life settings. By addressing privacy concerns, building upon the lessons learned from the Pecan Park study, and taking advantage of the continuous improvement of smartphone hardware in terms of sensors and energy efficiency, the LiveLab methodology records detailed usage data along with even more context data.

Furthermore, to the best of the author's knowledge, all existing smartphone loggers in literature, including the ones reported above, employ a static installation method and are therefore difficult to update or maintain. However, as highlighted previously in [60], new research hypotheses develop when data are collected from mobile users in the field. This requires the in-device logger to be updated frequently to collect new data. LiveLab, on the other hand, is updates itself automatically every day through an encrypted connection, if necessary.

Finally, the author highlights that there have recently been reports of closed source loggers, namely CarrierIQ [62] that log an unspecified amount of usage details without obtaining users consent in multiple phone platforms, including the iPhone. While the exact details of CarrierIQ are still unclear, the logger apparently has been installed with the cooperation of US phone carriers [63]. The sensitivity caused by such reports highlights the privacy challenges of long-term logging, while the scale that it has been used highlights the value of such data logs.

2.3.2. **Effects of Social Status on Mobile Usage**

Understanding user diversity has been a central tenet of human-computer interaction (HCI) research [64]. Previous studies have shown that Socioeconomic Status (SES) differences are important to consider for the design of other technologies. In [65], Goel et al. revisited the digital divide and found, among other differences, that SES difference drove how frequently web pages were accessed. Individuals in lower SES brackets accessed the web more than their higher SES peers. Similarly, Tossell, C., along with the author, found that lower SES users of iPod Touch devices who had less access to other technologies used their iPod Touch substantially more for activities commonly used on PCs [66].

In contrast, Chapter 7 of this dissertation investigates social context to measure its effect on mobile usage. It analyzes and compares the apps and websites used by two carefully selected distinct socioeconomic subgroups, collectively. These findings provide researchers and system designers with a better understanding of how users vary, resulting in better support of a broader range of individuals with different backgrounds, capabilities, skills and interests.

Chapter 3

Background and Motivation

Context has in the past been widely exploited for individual applications and system services, providing improved efficiency and performance. In this section, as a motivation, this dissertation presents the author's prior work that employs context and limited long term user studies. It then presents LiveLab, our solution to collecting long-term usage and context traces from real-world user studies.

The author's prior work attests to the value of context by calculating and measuring significant, quantified, performance gains by employing context information. Yet, the fundamental question of quantifying and measuring the context dependency remained. Furthermore, obtaining context information often incurs a significant energy cost. This has encouraged many researchers to substitute lower cost context for energy hungry ones, or propose various mechanisms, sometimes themselves context based, to reduce the frequency of accessing context.

On the other hand, studying context dependency directly itself can be extremely challenging, as it needs a large trace collected in real life user studies. While extensive logging of PC usage has been reported in past literature, privacy concerns, and more importantly, battery lifetime limitations, have limited the scope of mobile phone based studies in the past, making such a study on context dependency previously impossible. The LiveLab methodology demonstrates that through the it's careful design and implementation, it can overcome the privacy and energy challenges on modern mobile devices, enabling long term logging of unprecedented data.

3.1. Context Dependency of Network Resources

In Context-for-Wireless [10, 33], the author explored the context dependency of network resources. This would, in turn, allow the system to automatically select between cellular and Wi-Fi for any given data transfer, without having to power up and probe the Wi-Fi network, significantly improving the energy efficiency of mobile Internet. The context information included prior Wi-Fi conditions, time of day, cellular network conditions, and device motion.

Using real-life network and context traces the authors collected in a two month user study, they devised effective algorithms to learn the conditional probability distribution of Wi-Fi network conditions, given context information. With the conditional probability distribution, the authors formulated wireless data transfer through multiple interfaces as a statistical decision problem, and showed that Context-for-Wireless can improve the average battery lifetime of a commercial smartphone by 30 to 40% for Electrocardiogram health reporting, close to the theoretical upper bound of the traces at 42%. The authors further demonstrated that it is possible to predict Wi-Fi availability for 1 and 10 hours into the future with 95% and 90% accuracy, respectively. The authors' experimentation using a proof-of-concept device confirmed these findings, showing a 35 percent improvement in battery lifetime.

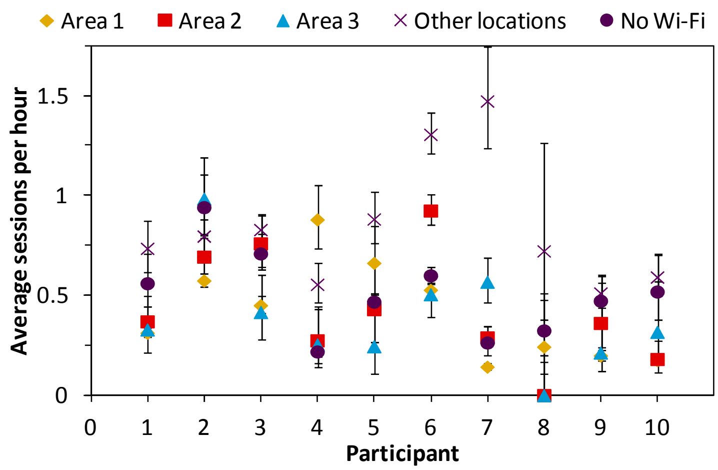
3.2. Context Dependency of Smartphone Usage

In addition to the traces the author collected through a field trial for Context-for-Wireless, the author carried out a longitudinal field study over four months with 14 teenagers from Pecan Park, an underserved community in

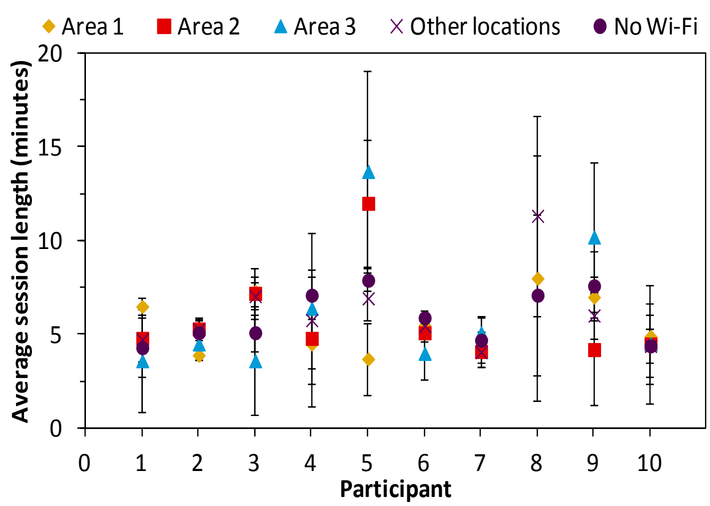
Houston, TX, where an open-access Wi-Fi network is available [60]. The authors used lessons learned from their previous user studies to devise a logger with minimal usability and privacy issues. Consequently, the findings and traces, while groundbreaking, were severely limited in scope. For example, in order to satisfy privacy concerns, the logger only recorded display on/off status as a measure of usage, instead of detailed recordings. The authors resorted to qualitative focus groups and other forms of self-reporting to amend the logs and enable groundbreaking results. From a then-unprecedented amount of qualitative and quantitative data, including focus groups, interviews, and in-device logging, the research suggested that smartphone usage is context-dependent.

In addition to self-reports of increased usage (or the non-usage) of certain apps in certain places and contexts, the Pecan Park study traces provided quantitative evidence for the location dependency of app usage. The logging software recorded visible Wi-Fi access points, and they were used to cluster the most visited access points into unique physical areas while maintaining user privacy. Figure 3.1 shows the average number of non-voice sessions per hour and the average session length at each location, respectively, for ten participants with sufficient data. It shows that most of users had significantly different usage characteristics at each participant's top locations, in terms of average session lengths and/or sessions per hour, hence their usage was location dependent.

The Pecan Park study also highlighted the importance, and the implementation, organization, and privacy challenges of long-term, holistic studies of mobile devices and services under real-life settings. The findings and lessons learned from the Pecan Park study, as well as the continuous improvement of smartphone hardware in terms of sensors and energy efficiency, enabled the LiveLab methodology, as will be presented next.



(a) Number of sessions (average = 0.42 / hour)



(b) Length of sessions (average = 6.0 minutes)

Figure 3.1 – Usage patterns of the Pecan Park participants. Areas 1 to 3 denote the top three location clusters where each participant spent their time, and were calculated for each participant separately. Locations that could not be classified due to lack of visible Wi-Fi access points are shown collectively as 'No Wi-Fi'. Whiskers indicate 95% confidence intervals.

3.3. LiveLab: Collecting long term context in-situ

This dissertation is enabled by the LiveLab methodology, which enables us to collect unprecedented usage and context data from 24 iPhone users continuously for one year. LiveLab addresses the technical and human factors that limited prior user studies of this duration and scope. The key component of LiveLab is a reprogrammable in-device logger designed for long-term field studies with minimum energy, usability and privacy intrusions.

3.3.1. **Participants**

The 24 LiveLab participants were young college students with an average age of 19.7 years (*SD* =1.1 years), and the study lasted from February 2010 to February 2011. These 24 balanced participants were recruited from two distinct socioeconomic status (SES) groups from a small private university at a major metropolitan area in the USA. They all lived on campus in dormitories. Because the university provides needs-based scholarships to lower-income students, this information was used as a proxy to separate the participants into two groups. 13 received need-based scholarships, and their household income was under $60’000. 11 users did not receive scholarships and their household income was over $80’000. Other factors, including their major, gender, race, PC access, and game console ownership, were balanced across groups. All had a PC or laptop at their residence, in addition to access to the university's computing labs. 11 of the low SES and all high SES participants had a personal laptop.

Every participant received a free iPhone for their participation, and was required to use the outfitted iPhone as their primary device for the entire year. Additionally, each participant received free service coverage, including 450 voice call minutes per month, unlimited data, and unlimited SMS during the study. All participants were assisted in porting their phone numbers to the iPhones. The participants were not given specific instructions on how to use the device, other than to use it as they would normally use their phones.

3.3.2. **Logger Design and Implementation**

The key component of the LiveLab is an in-device, field modifiable logging software that collects usage and context *in situ*.

To run the iPhone logger in the background continuously, it was necessary to jailbreak the iPhone 3GSs and exploit a setting provided by the iOS to start the daemon process on boot, as well as restart the daemon process anytime it is killed. The main logger daemon is written as a bash shell script and utilizes components written in various languages, including C, Perl, awk, SQL, and objective C, altogether comprising ~2000 lines of code. Furthermore, the logger daemon is able to call built in functions, manage child processes, install and use programs from repositories, run custom programs, and add new features. The LiveLab logger is implemented in a modular and robust fashion, thus new iOS releases may break individual components, but the main functionality will not be affected. In order to monitor and update the logger, it is programmed to report data and, if necessary, update itself every day through an encrypted connection, via rsync [67], to a lab server. Over the course of the study over 20 automatic updates were deployed to fix bugs, add functionality, and conduct temporary experiments.

The logger records a plethora of context information, as shown in Table 3.1. However, this dissertation focuses on logs regarding usage and context. The visited websites, apps used, and phone calls are recorded by the phone's operating system, and the logger piggy-backs on the phone's logs by periodically recording them. Further, whenever the phone's CPU is not asleep, at 15 minute intervals, the logger records the iPhone reported location, the cell tower the phone is associated to, and a 15 second recording from the accelerometer at 25hz. The GPS location data is collected using Apple's provided framework, which reports the GPS location if available, and, if not, the estimated location based on visible cell towers and strengths; both methods provide an estimated accuracy. The logger attempts to retrieve the location until the accuracy is less than 100m, or the location has been updated (by the framework) 3 times, in order to avoid running continuously when a GPS lock is unavailable, yet still retrieve accurate location data. For cell ID location, the logger queries the phone's GSM modem using the AT command set, returning the single, currently associated cell ID. The data is recorded on the phones, and is transferred nightly to our servers in a secure fashion. The logger has recorded thousands of usage samples through the year-long study, as presented in Table 3.1 and Table 3.2.

Table 3.1 – Usage and context data collected through LiveLab

Item	*Privacy method*	*Logging method*	*Energy per measurement*
Call logs and Address Book	1-way hashing	Piggy-back	0
SMS, Email	feature extraction, hashing	Piggy-back	0
Web History	1-out-of-n	Piggy-back	0
Installed programs and media	1-out-of-n	Piggy-back	0
Captured media, e.g. photos, videos, and voice	1-out-of-n	Piggy-back	0
battery / charging status	1-out-of-n	Interrupt	negligible
App. launches; changes to foreground app.	1-out-of-n	Interrupt	negligible
GPS Location	1-out-of-n	Interval (15 min.)	50 J to 300 J
Accelerometer reading	1-out-of-n	Interval (15 min.)	1.65 J
Currently running processes	1-out-of-n	Interval (15 min.)	negligible
Cell ID, signal strength	1-out-of-n	Interval (15 min.)	1.2 J
Wi-Fi Access Points	1-out-of-n	Interval (15 min.)	1.5 J
Network bandwidth, latency	1-out-of-n	Interval	variable

Table 3.2 – Number of usage data samples collected through LiveLab

Type of usage	*Total samples*	*Mean samples per user*
Websites visited	17,000	700
Phone calls	54,000	2,300
Applications launched	508,000	21,200

3.3.3. **Privacy Challenges**

Collecting data from smartphones in the field naturally incurs privacy issues. In order to collect valid data, these issues must be investigated and addressed in an acceptable manner. To this end, the particular concerns of the participants were investigated prior, and traditional as well as novel methods were employed to address the participants' concerns, in particular the novel method of *partitioning*.

Before even beginning the deployments, we conducted several interviews with all of the potential participants. We discussed the logger with the participants in depth at a formal meeting in order to develop a better understanding of their particular privacy concerns and the information they are unwilling to have logged. Not surprisingly, we found that participants' biggest concern was regarding their identity. That is, participants do not want researchers to be able to associate their identities with their data. Surprisingly, they are not concerned about some potentially sensitive data being collected as long as the data is not directly linked to their identity. The participants were fine with a 1-out-of-n anonymity for much of the data we originally considered private, including GPS location and Web access history. In contrast, they wanted more privacy regarding other items, such as the contents of emails and messages, as well as the identities of their phone contacts. However, we found that they are comfortable with the on-device feature extraction and/or hashing performed by the logger, as long as the raw data is never stored on disk.

Based on these finding, we designed and employ the following methods to protect user privacy while retaining relevant information for research. First, we *partitioned* the research team so that the data analysis and logger development team do not know or directly interact with the participants, in order to avoid linking data to the actual users. A separate human factors team acts as the interface with the participants but does not deal directly with the logger or access the raw data. This enables us to contact the participants in a privacy sensitive manner, which we have found to be necessary on numerous occasions, e.g., to schedule impromptu interviews with users who exhibit a drastic change in behavior. To the best of the author's knowledge, knowledge, LiveLab is the first to propose and implement partitioning to address the privacy challenges of data logging in user studies. Second, LiveLab performs feature extraction and one-way hashing on the phone itself, to preserve the important information and the uniqueness of a data entry without even storing its content. For example, by hashing phone numbers, we can construct call statistics without knowing the actual phone numbers. As another example, the logger extracts and stores statistics regarding text messages, such as their length, number of words and emoticons, without storing their actual content.

3.3.4. **Energy Challenges**

Collecting data from smartphones in the field naturally incurs power overhead and reduces battery lifetime. Significantly reduced battery lifetime is likely to impact usage, thus the usage data would not accurately reflect user behavior in real life [35, 60]. Therefore, we need to carefully consider and mitigate the energy impact of the logger. The LiveLab design simultaneously aims for improving the energy efficiency of logging, while improving the accuracy and usefulness of the collected data. We utilize three main concepts in this regard:

First, LiveLab minimizes the energy overhead of logging by using built-in logs and interrupt driven data collection whenever possible. For all data collection, we have attempted to utilize, in order, 1) built-in, always running, logs, 2) interrupt driven logging, and 3) polling based logging. By using built-in logs we incur only the power overhead of reporting the data. For events that are not logged automatically by the platform, such as power state changes and app launches, we leverage interrupt based logging. For items that we can't log continuously, due to the power overhead, such as location data and accelerometer data, we poll at regular intervals.

Second, for items that are infeasible to continuously monitor, such as location and accelerometer data, LiveLab attempts to maximize the utility of the collected data by only logging them when the phone's CPU is active, i.e. the phone is being used. This further reduces the overhead of running the logger, since the CPU is running anyway.

Third, since the periodically logged items are energy-expensive, e.g. GPS readings and bandwidth measurements, LiveLab optimizes the logging interval in order to maintain a reasonable battery life. In order to find a reasonable tradeoff, we utilize the energy measurements in Table 3.1. Based on the iPhone battery capacity of 1200 mAh at 3.7V, we calculate the energy capacity of the iPhone battery at 16 kJ. Consequently, every time that the logger collects the energy hungry polled data, 0.3% – 1.8% of the battery capacity is consumed. We target an average battery consumption of 10% per day. Therefore, we chose the 15 minute interval to achieve that target for 3-hours per day phone usage duration.

Note that, for many purposes, such as network performance measurements, the long duration of the field trial reduces the need for frequent measurements. The rationale is that a longer duration will allow the data to be collected at more times of the day and more locations. While such data will not be able to catch certain temporal dynamics, e.g. the location trace of a user in a given day, they still retain key statistics regarding the user and network behavior.

3.4. Collected Data

3.4.1. **Usage**

For all three principal types of mobile usage (phone calls, web usage, and app usage), the number of usage categories that are considered are limited to 100. This simplifies data processing, and can even increase accuracy by reducing usage cases with too few samples. The number 100 was chosen based on the CDF of usage, covering 87%, 93%, and 99% of web, phone, and app usage.

This dissertation considers web usage as the independent websites a user visits, as presented by their domain names. Each visited domain is one entry, irrespective of the number of web pages under that domain. This would include the top level domain (TLD) and the first hierarchical subdomain, e.g. *www.example.com/url/* would be counted as *example.com*. For phone calls, while the logger does not record actual phone numbers, it records a one-way hash that uniquely identifies each phone number. All phone calls are considered, including ones that have a length of zero, indicating no conversation. For app usage, this dissertation considers all apps the user utilizes, including built-in ones and those obtained from the App Store. The home screen, however, is not considered as an app, even though it is implemented as an app on the iPhone platform. Note that this dissertation uses the word *app* and *application* differently. App is used to refer to applications that are installed on the iPhone, either built-in or obtained from the App Store. Application is used to refer to its more general meaning, i.e. use case. Similarly, application agnostic refers to multipurpose, not dedicated to a single service or purpose.

3.4.2. **Context**

This dissertation considers several sources of context in two broad categories; *sensor context* that is sensed through the phone sensors, and *usage context* that is last known usage state of the phone. The sensor context this dissertation utilizes are time&day, movement (accelerometer power), cell ID location, and GPS location. The usage contexts this dissertation utilizes are the prior visited website, phone call, and app.

For time&day, this dissertation **s**eparates weekends and weekdays, but otherwise treat days as the same. Separating weekends from weekdays not only makes intuitive sense, but testing on the LiveLab dataset indicated that it performed better than treating all days the same. Therefore, with a one-minute resolution, time&day is a continuous number between 0 and 2880, to account for a two day period (a weekday and a weekend).

For movement, this dissertation calculates the log of the power of the accelerometer readings. The rationale of utilizing the log of power, instead of absolute power, is the distribution of power readings that is close to the power law. More than 99% of the $log(p)$ entries fall between 0.1 and 10000, and the range is therefore limited accordingly.

For GPS location, this dissertation utilizes the most accurate location provided by the iPhone API, which is provided in the geographic coordinate system, i.e. latitude and longitude. For cell ID location, this dissertation utilizes the (single) cell ID reported by the iPhone's GSM modem.

3.5. Data Processing

Due to the extremely large size of the LiveLab logs, it is often necessary to process them sequentially. Therefore, the tools to process them were mostly developed by the author using the Perl language and Bash scripts. The author also took advantage of several open source tools to help process the logs, as follows.

The popular database, PostgreSQL, was utilized to store, access, and preprocess the LiveLab data. PostgreSQL is an object-relational database management system (ORDBMS) developed by the PostgreSQL Global Development Group, consisting of community volunteers and employees throughout the world [68].

The popular clustering package, Cluster 3.0 [69] was utilized for k-mean clustering throughout this dissertation. Cluster 3.0 has been developed in C by members of the Laboratory of DNA Information Analysis, Human Genome Center of the University of Tokyo. Cluster 3.0 is based on prior efforts by Michael Eisen of Lawrence Berkeley National Laboratory [70]. The Perl wrappers used to interface with Cluster 3.0 were developed by John Nolan of the University of California, Santa Cruz [70].

Chapter 4

Quantifying Context Dependency

As previously highlighted, context dependency can be broadly defined as a set of strict or probabilistic rules and relations between two often discrete variables, context(s) and outcome [2]. Multiple theoretical, application agnostic metrics exist for measuring the relationship of such variables. These include entropy as a measure of uncertainty, and Pseudo R Square as a measure of correlation [71]. This dissertation presents *estimation accuracy*, based on maximum a posteriori probability (MAP) estimation as a formal yet practical, application agnostic definition for context dependency. In contrast with other theoretical metrics, such as entropy and pseudo R, estimation accuracy provides practical insight into the performance of many potential applications and services, while remaining application agnostic.

Based on this definition of context dependency, this dissertation provides practical methods to calculate the posterior probability of an outcome (g) given context (x), or $P(g|x)$, from one or multiple continuous or discrete (multinomial) context sources. In order to calculate $P(g|x)$, it is necessary to discretize continuous context into a limited number of (multinomial) cases. Further, discrete context may have too many possible values to efficiently calculate $P(g|x)$, producing too few samples per category and inaccurate posterior probability calculations. In these cases, it is necessary to group some categories together to reduce the number of categories. This is referred to as binning. Clearly, the performance of the analysis will depend on how context is discretized and/or binned. This dissertation presents and compares several characteristic methods for binning data.

4.1. Formal Definition of Context Dependency

In order to formally define context dependency of mobile usage, this dissertation presents mobile usage as a random variable that one is interested in estimating, g. Without any other information but the observations of g itself, one can only estimate g with $\hat{g}$ that minimizes the expected estimation error $E[L(g, \hat{g})]$. $L(a,b)$ is a loss function containing the penalty for an incorrect estimate. Similarly, this dissertation defines estimation accuracy as $1 - E[L(g, \hat{g})]$.

For clarity and without loss of generality, this dissertation assumes g takes value from a finite set $\{g_1, g_2, \ldots, g_k\}$, and the loss function is a zero-one function, i.e., 0 when $g = \hat{g}$, and 1 otherwise. In this case, the optimal estimation would be

$$\hat{g} = \underset{i}{\operatorname{argmax}}\, P(g = i)$$

and the corresponding expected estimation accuracy is

$$P(g = \hat{g})$$

The use of context information can help reduce the expected estimation error. Assume one can observe not only g but also concomitant contextual information presented as a vector $x=(x_1, x_2, \ldots, x_n)$. To estimate g, one has an additional piece of information: the current observation of x. Assuming the same loss function as used above, from statistical machine learning text, the optimal estimation would be

$$\hat{g}(x) = \underset{i}{\operatorname{argmax}}\, P(g = i|x)$$

where $P(g|x)$ is the a posteriori probability. Now the expected estimation accuracy is $\sum_{\mathrm{x}} P(g = \hat{g}(x)|x)P(x)$, which should always be higher than $P(g = \hat{g})$, the expected estimation accuracy without the contextual observation *x*. The estimation is known as the maximum a posteriori (MAP) estimation. When g is continuous value and loss function is squared error, the optimal estimation can be similar calculated as the expectation of g, $E[g]$, without the contextual observation, and $E[g|x]$, the conditional expectation of g when the contextual observation is available.

In applications and services that the cost of a false negative is considerably higher than false positive, providing multiple responses (or best guesses) can be beneficial. For example, for application preloading, the system could preload multiple applications instead of one. This would, in turn, reduce the miss rate, i.e. not having the next application preloaded. One can use the same definition to allow multiple responses. In this case, the expected estimation accuracy would be $\sum_{\mathrm{x}} P(g \in \hat{g}(x)|x)P(x)$.

4.2. Calculating Posterior Probabilities

The key to MAP estimation is the accurate calculation of the a posteriori probability distributions. Posterior probability is the conditional probability of the outcome, g, given certain context, *x*. In other words:

$$P(g|x)$$

It may initially appear straightforward to calculate $P(g|x)$, by simply dividing the number of times each possible outcome, g_i, has occurred under context condition x, by the number of times x has been observed in total. Recall that g takes value from a finite set, $\{g_1, g_2, \ldots, g_k\}$.

$$P(g_i|x) = \frac{count(g_i|x)}{count(x)}$$

However, if the number of times x has been observed is small, the estimates of $P(g|x)$ are unreliable [72]. Due to the large number of possible context combinations, and the possibility of having few or no prior samples in a given context, posterior probability estimates may become inaccurate or impossible. This is true even for individual context sources, but can become significantly worse if multiple context sources are treated as multiple dimensions, due to the curse of dimensionality [14].

In this section, this dissertation examines several prominent methods to address this challenge. It first examines the case when dealing with only one context source; it employs Laplace Correction to reduce the negative impact of too few observations under some contexts. This dissertation then examines the case for dealing with multiple context sources. Instead of treating the context space as multi-dimension, each context source determining one dimension, it explore several prominent classifier combination techniques in order to treat each context source as a separate single-dimensional predictor.

4.2.1. **Individual Context**

Under context conditions that occur infrequently, the simple estimates of $P(g|x)$ becomes unreliable. To reduce the negative impact of too few observations under some context conditions, this dissertation employs *Laplace Correction*. There are multiple methods in literature for Laplace correction. For example, [73] calculates it as

$$P(g_i|x) = \frac{count(g_i|x) + 1}{count(x) + c}$$

where c is the number of possible outcomes. In contrast, to smooth the posterior probability towards $P(g_i)$, the prior probability of outcome g_i, [74] calculates the Laplace correction as

$$P(g_i|x) = \frac{count(g_i|x) + m.P(g_i)}{count(x) + m}$$

The parameter m controls the degree of regularization of the estimate. A larger m will result in an estimate closer to $P(g_i)$. This dissertation uses a combination of the two methods by setting m as the number of possible outcomes (c in the prior formula). Note that Laplace Correction only smoothes out the estimate $P(g|x)$ when there are a small number of context samples. The effect of Laplace Correction is negligible if there are significantly more samples than outcomes, i.e., *count*(x) >> m.

4.2.2. **Multiple Context**

As presented earlier, treating the context space as a multi dimensional space, with each context source determining one dimension, will result in an unacceptably sparse data set due to the curse of dimensionality. In order to address this challenge for calculating posterior probability, i.e., $P(g_i|x)$, where $x = \{x_1, x_2, \dots, x_n\}$, this dissertation employs classifier combination techniques. Classifier combination techniques enable one to treat each context source as a separate one-dimensional predictor, and combine multiple $P(g_i|x)$ into $P(g_i|x_1, x_2, \dots, x_n)$. In other words,

$$P(g_i|x_1, x_2, \dots, x_n) = combination(P(g_i|x_1), P(g_i|x_2), \dots, P(g_i|x_n))$$

This dissertation explores three prominent classifier combination techniques. The first is the *Naïve Bayesian Rule*, which works under the assumption that different sources of context are conditionally independent. That is, $P(x_i|g, x_k) = P(x_i|g)$, for all $i \neq k$. The assumption that different sources of context are conditionally independent does not necessarily hold. However, Simple Bayesian is known to often perform well even without this condition [75, 76]. We therefore evaluate the performance of Simple Bayes alongside other methods. The Bayesian rule calculates $P(g_i|x)$ as

$$P(g_i|x) = \frac{P(x|g_i)P(g_i)}{P(x)}$$

Assuming individual context sources are conditionally independent, $P(x|g_i)$ can be calculated as

$$P(x|g_i) = P(x_1, |g_i). \dots . P(x_n \, |g_i)$$

and $P(x)$ can be simply treated as a fixed constant, or can be calculated as

$$P(x) = \sum_i P(g_i). P(x_1|g_i). P(x_2|g_i). \dots . P(x_n|g_i)$$

Using the collected usage traces, it is straightforward to calculate $P\left(x_j, \middle| g_i\right)$, either directly, or, again according to the Bayesian rule according to the posterior probabilities of each individual usage:

$$P\left(x_j \middle| g_i\right) = \frac{P(g_i|x_i)P(x_i)}{P(g_i)}$$

This dissertation calculates $P\left(x_j \middle| g_i\right)$ directly, using the training data. Similar to Chapter 4.2, Laplace Correction is utilized to reduce the ill effects of too few samples, as shown below. Again, *c* is the number of possible categories for *x*.

$$P(x_j|g_i) = \frac{count(x_j|g_i) + c.P(x_j)}{count(g_i) + c}$$

The second combination technique this dissertation explores is the *Maximum Rule*. The probability of each outcome is reported proportional to the maximum probability of that outcome among all classifiers, so that the sum of probabilities remains equal to one. Formally,

$$P(g_i|x_1, x_2, \dots, x_n) \propto max(P(g_i|x_1), P(g_i|x_2), \dots, P(g_i|x_n))$$

For example, if one classifier selects outcome A with a 80% posterior probability, and two other classifiers select outcome B with 70% and 60% posterior probabilities, outcome A will be selected, and a posterior probability proportional to 80% is reported. The Maximum Rule is known to be highly sensitive to noise [77], when one classifier may be producing a high confidence due to noisy data or too few samples.

The third combination technique this dissertation explores is the *Mean Rule*, also known as the *Average Rule*. It calculates the probability of each outcome as the average of the reported probabilities by each of the classifying methods. Formally,

$$P(g_i|x_1, x_2, \dots, x_n) = mean(P(g_i|x_1), P(g_i|x_2), \dots, P(g_i|x_n))$$

The Mean Rule is known to be especially resilient to noise [78], and most useful when the classifiers are highly correlated. There exist variations to the Mean Rule, such as the weighted Mean Rule where the classifiers are combined according to a weight assigned to each classifier. The weighted mean is reasonable when classifiers are not of equal accuracy, where a higher weight is assigned to more accurate classifiers. This dissertation explores the equal weight Mean Rule.

4.3. Discretization and Binning of Context

Based on the definition of context dependency described in Section 4.1, in order to calculate $P(g|x)$, it is necessary to abstract context into a limited number of discrete (multinomial) cases. However, often, context is available in continuous form. In such cases, one needs to discretize the data. Other times, context may be available in discrete form, but with too many possible values to efficiently calculate $P(g|x)$. In such cases, it is necessary to limit the number of categories to prevent too few samples per category producing inaccurate posterior probability calculations. This is referred to as binning categorical data. Clearly, the accuracy of a posteriori probability calculations will depend on how the data is discretized and binned. In particular, the number of bins chosen for any context involves an inherent tradeoff; more bins can allow finer molding of bins and more accurate posterior probability calculations, but at the same time would result in fewer samples per category, increasing noise and reducing the accuracy of posterior probability calculations.

Note that some texts use the term binning for the discretization of continuous data. In contrast, this dissertation distinguishes between the categorization of continuous and categorical data by calling them discretization and binning respectively. Obviously, after discretization of continuous data, they can be binned as well, similar to discrete data.

This section presents and compares several common methods for discretization and binning of context data. This dissertation evaluates the efficacy of outcome-based Supervised Binning for both categorical and continuous context, and compares them to the more simple approach of using the top-n categories for categorical data and simple equal frequency categorization for continuous data. Note that there are significant variations on the number of categories required to make efficient use of each context source, as will be presented later in Chapter 5.

4.3.1. **Discretization of Context**

Continuous sources of context inherently have a single or multidimensional structure to them, where samples close to each other are related. Such a structure allows for the use of unsupervised clustering techniques to discretize them. There have been many methods in literature for unsupervised clustering of single and multi-dimensional entries, and consequently creating the discrete cases. Furthermore, there is often an option of adding an *equal frequency* constraint, as opposed to *equal width* discretization, as shown in Figure 4.1.

Both equal width and equal frequency discretization is straightforward for one dimensional context, such as time and movement (accelerometer power). Equal width discretization divides the range of possible values into an arbitrary number of equal width segments. For example, to divide time-of-day (00:00 – 23:59) into three bins, the segments would be 00:00 – 07:59, 08:00 – 15:59, and 16:00 – 23:59. Equal frequency binning divides the range of possible values into an arbitrary number of segments (bins) so that there are an equal number of samples in each bin. For example, if there are six samples and three bins, the boundaries are selected to fall between samples in such a way to have two samples per bin.

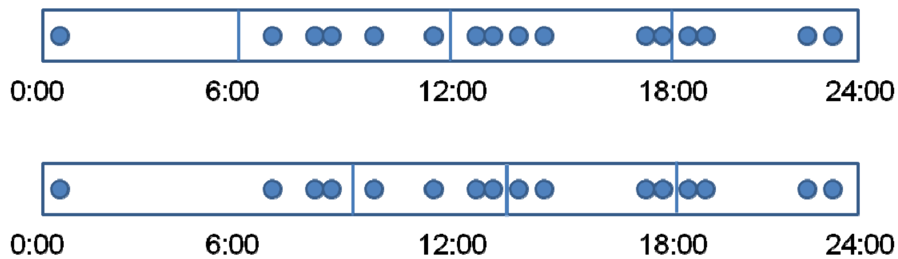


Figure 4.1 – Equal width discretization (top) would result in more natural boundaries. Equal frequency discretization (bottom) would use bins more efficiently and prevent too few samples in some bins.

An equal number of samples per cluster would guarantee efficient use of clusters, and prevent a too few number of samples in some clusters producing inaccurate results. On the other hand, the equal sample constraint may artificially limit the boundaries in the clustering algorithm, resulting in inefficient clusters.

For continuous context in multiple dimensions, this dissertation refers to clustering literature to find a suitable unsupervised clustering algorithm. The resulting clusters would in turn become the categories. There are two requirements for such an algorithm. First, obviously, samples close to each other must be in one cluster. Second, in order to evaluate the effect of the number of categories and the number of samples per category, the number of clusters must be specifiable by us. This dissertation utilizes the popular k-mean clustering since satisfies its two constraints; it partitions entries into k clusters while minimizing the Within-Cluster Sum of Squares (WCSS).

For equal frequency discretization of multidimensional context, this dissertation proposes and evaluates an extension to k-means. Assuming m entries and a cluster size of n ($m=k.n$), the algorithm works as follows. First, a regular k-mean clustering is performed on the entries. The biggest cluster, i.e. the one with the most entries, is then selected, and the closest n entries to the mean are assigned to that cluster. The remaining entries are clustered again, using the same method (i.e. $m-n$ entries into $k-1$ clusters). This is repeated until all entries are clustered. To retain meaningful results and prevent overlapping entries to be placed into different clusters, the equal size constraint is relaxed when two entries overlap or the resulting cluster radius is too small (e.g. for location clustering, closer than 5 meters to the centroid of a location cluster).

4.3.2. **Binning Categorical Context**

Categorical context samples can be either ordinal, where there is an inherent structure or order in the samples, or nominal, when there is no such structure. For *ordinal categorical context*, it is possible to utilize this structure for binning. Similar to continuous context, both equal frequency and equal width binning would be applicable, and clustering based on a distance metric can be applied to multidimensional context. Another approach to binning is to utilize supervised learning based on the outcome. This could be applied to both ordinal and nominal categorical context.

Supervised binning provides a best of both worlds solution to the inherent tradeoff of the number of bins. Recall that more bins can allow finer molding of bins and more accurate posterior probability calculations, but at the same time would result in fewer samples per bin, increasing noise and reducing the accuracy of posterior probability calculations. Supervised Binning can identify and bin together categories that have similar outcomes. Therefore, supervised binning can allow finer molding of bins, increasing accuracy, without reducing the number of samples per bin that would increase noise and reduce accuracy.

Supervised Binning can be performed on both categorical and continuous context. For the latter, it must first be discretized into a larger number of categories, and then binned. One must note that in order to preserve the integrity of results in supervised binning, it is imperative to separate the training data that is used for creating the bins, from the testing data.

There are two common approaches to supervised binning:

The first approach is to find or create a distance metric that can calculate the distance between any two predictors using the outcome. Consequently, one can then utilize traditional clustering algorithms e.g. k-mean, to heuristically bin the categories, similar to structured context. Calculating the distance between context cases would mean calculating the 1-norm, 2-norm, cosign, etc. distance between the outcome (usage) vectors $\{P(g_1|x), P(g_2|x), \ldots, P(g_n|x)\}$ of any two context entries, denoted as x.

This dissertation evaluates the efficacy of Supervised Binning using this method. To this end, this dissertation uses k-mean clustering based on the 2-norm distances of the normalized Laplace-corrected usage vectors. Note that each usage vector is defined as the posterior probability of each of the possible outcomes, i.e. $\{P(g_1|x), P(g_2|x), \ldots, P(g_{100}|x)\}$. The result is a 100 dimension vector, to account for each of the top 100 usages. The effectiveness of Supervised Binning for the context dependency of mobile usage is presented in Chapter 5.

The second approach is enabled by building a classifier tree. From the resulting classifier tree, it is possible to use the derived partitions to bin the categories. An optimal classifier tree is in fact the Supervised Binning of context. Note that in building the classifier tree, one is free to use any gain or loss function they desire, such as prediction accuracy, as presented in [14].

Unfortunately, building the optimal classifier tree may be computationally impossible when dealing with categorical predictors and outcomes. There are methods to simplify the calculation, as long as one of two conditions are met. Specifically, 1) if the input (predictor) categories are ordered, or 2) if the output (outcome) is binomial, i.e. 0 or 1. For this dissertation, neither of these conditions hold. In the absence of such conditions, one has to resort to heuristic methods to build (a suboptimal, but reasonable) decision tree, such as CHi-squared Automatic Interaction Detector (CHAID), a recursive tree classification method [79].

For the purpose of comparison, and as a baseline, this dissertation defines Simple Binning as follows. For binning categorical context into *n* bins, we simply choose the most popular *n-1* categories, and group all other categories as the *n*'th bin. This is especially reasonable if the distribution of context follows the power law, which is often the case. For consistency, for continuous context, this dissertation defines Simple Binning into *n* bins as discretizing the context into *n* categories.

Chapter 5

Context Dependency of Mobile Usage

Using the LiveLab dataset presented in Chapter 3.3, and the definition of context dependency and methods presented in Chapter 4, this Chapter presents a series of interesting findings regarding the context dependency of web, phone, and app usage. These findings have significant implications on context aware application and services, several examples of which are discussed in this Chapter. Note that context-based prediction of usage is extremely challenging due to the distribution of all three usage types that closely resemble the power law (Figure 5.1): the most popular usage of each user constitutes a major proportion of their usage, and much higher than the

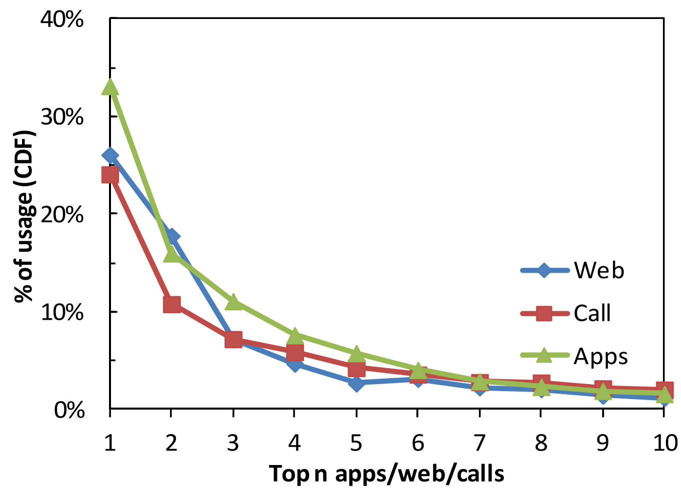


Figure 5.1 – Distribution of usage resembles the power law. Therefore, estimating usage is challenging (average for all participants)

second most popular usage, and so forth. Consequently, even given context based evidence, it is unlikely for the posterior probability of any usage to rise above the more common usage.

As mentioned prior, for applications and services in which the cost of a false negative is considerably higher than false positive can benefit from multiple responses in the form of $\hat{g} = \widehat{g_1} \cup \widehat{g_2} \cup \ldots$. For example, for application preloading, the system might preload multiple applications to reduce the chance of a miss. Accordingly, this dissertation considers the case for 3 and 10 responses as well as the single response case.

5.1. Individual Context

This dissertation studies the performance of individual context by measuring the estimation accuracy of web, phone, and app usage versus the number of context bins, as shown in Figure 5.2, Figure 5.3, and Figure 5.4, respectively. One context bin would mean no context, i.e. always returning the most likely result(s). This section utilizes leave-out-one cross-validation (LOOCV) to preserve the integrity of results. LOOCV removes the test sample from the training data set used to calculate posterior probabilities. This section also utilizes Simple Binning to keep the meaning of bins easy to understand. Recall from Chapter 4.3.2 that for categorical context, Simple Binning utilizes the top n-1 categories and an 'other' category. For continuous context, it simply discretizes it into n bins.

The findings presented in this section are regarding the performance of each context, as well as the effect of the number of bins. Note that different context have a widely diverse range of effectiveness, depending on usage and the number of acceptable responses. Also, it can be seen that more bins initially improves performance, but after a point will hurt performance. The reason is that even though more bins can allow finer molding of the model, hence a more accurate calculation of $P(g|x)$, it would result in fewer samples per bin, increasing noise and reducing accuracy.

An important finding not inherently obvious in the figures is that an increase in the number of context bins is useful only as long as there are a reasonable number of samples to reliably calculate the posterior probability of each bin. As a rule of thumb, there should be more than ten samples per bin, even though the ill effects of too few samples are mitigated to an extent using Laplace correction. As shown in Table 3.2, there are on average 700 website visits per user. Therefore, it is unsurprising that in particular for equal cluster size contexts, there are diminishing results in going over 10-50 context bins. On the other hand, since there are over two thousand phone call samples, increasing the number of context bins is fruitful up to 100-200 bins, where the returns are diminished. This shows that the number of context bins should be not preset, but dynamically adjusted by the system to ensure a reasonable number of samples per category.

This dissertation next provide findings specific to each context type:

5.1.1. **Time&Day**

Recall that separating weekends from weekdays increases performance, compared to treating all days as the same. This dissertation also found that equal frequency discretization of time&day performs better than equal width discretization. The effectiveness of time&day levels off early, when the number of bins is extended beyond ~20.

5.1.2. **Movement**

Similar to time&day, equal frequency discretization of accelerometer power performs slightly better than equal width discretization. Interestingly, a relatively high number of bins (e.g. 100) are most effective here. This is in contrast with original expectations that a small number of bins, e.g. to account for moving and non-moving states, would be sufficient. This finding suggests that accelerometer power can and should be used *as a signature to classify a user's detailed state*, and not merely as an indicator for whether they are moving or not. Previous research has shown a similar phenomenon with ambient sound for the purpose of room level localization, i.e. SoundSense [80].

5.1.3. **GPS Location**

In contrast to the single dimension contexts, location performed best without the equal frequency cluster constraint. The author believes this is due to the equal frequency constraint artificially breaking down meaningful clusters in order to satisfy the sample size constraint. Interestingly, without the equal frequency constraint, a larger number of bins do not reduce performance.

The author believes this is due to the extra clusters mostly absorbing outliers, instead of breaking down meaningful clusters.

5.1.4. **Cell ID Location**

As each cellular cell spans a large coverage area, most of a user's life is under a small number of cell IDs. Therefore, there is little to gain from increased number of context bins. Note that as cell ID is already discrete, discretization doesn't apply here.

5.1.5. **Prior usage**

All three types of usage are most dependent on the prior usage of the same type, versus other forms of prior usage. For example, phone calls are more dependent on the prior phone call. Yet, other types of usage are also more or less good indicators. The only exception was that prior app usage seems to have no effect on web usage.

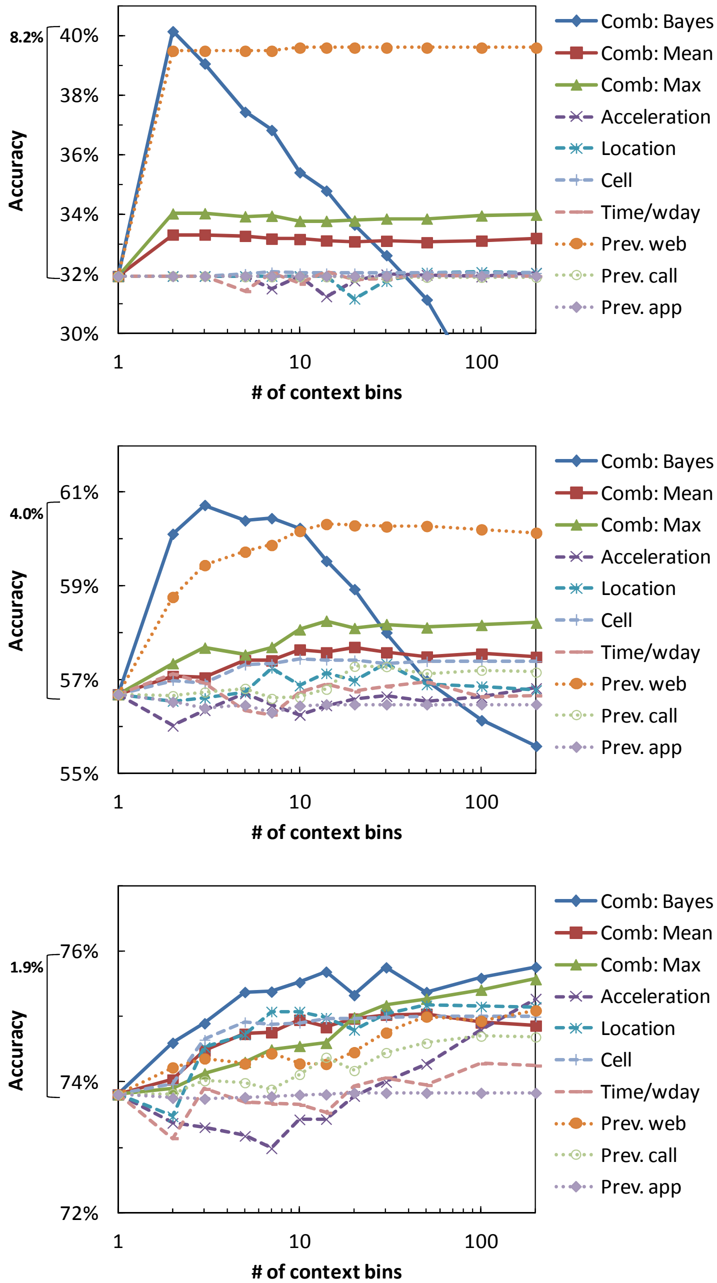


Figure 5.2 – Context dependency of web usage, presented as the estimation accuracy, for 1 (top), 3 (middle), and 10 (bottom) accepted responses

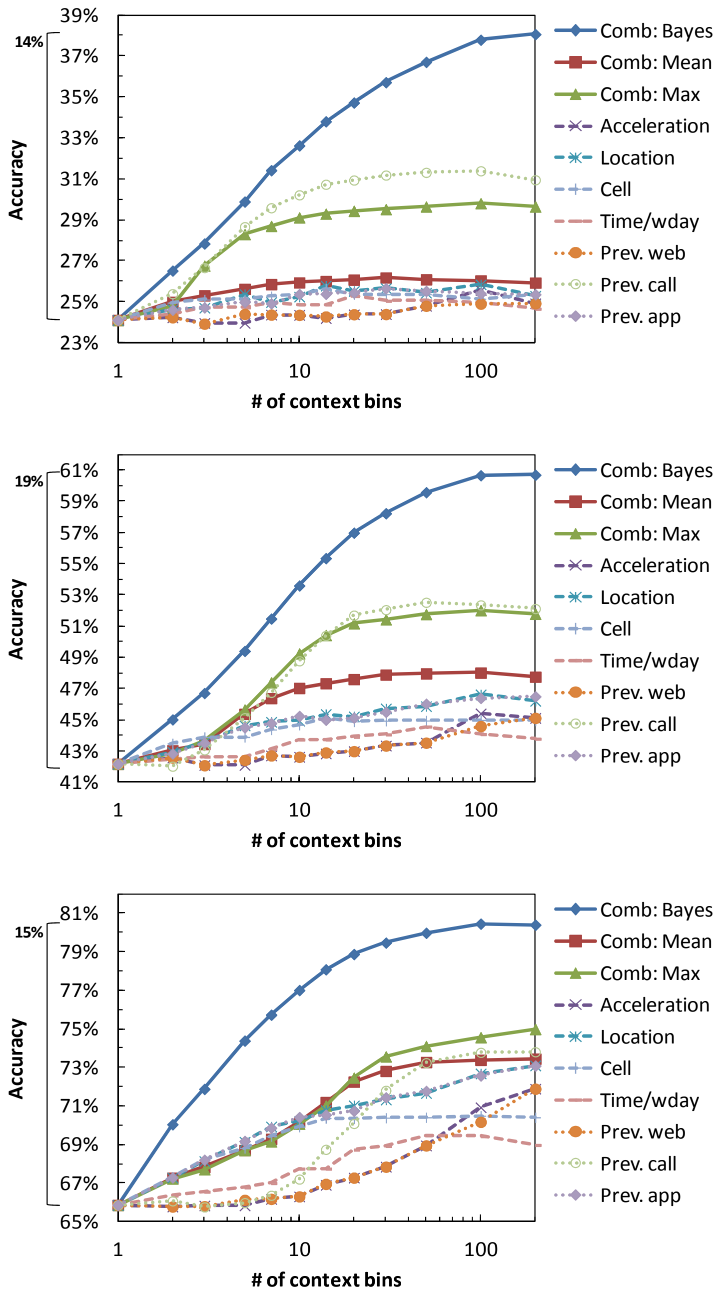


Figure 5.3 – Context dependency of phone calls, presented as the estimation accuracy, for 1 (top), 3 (middle), and 10 (bottom) accepted responses

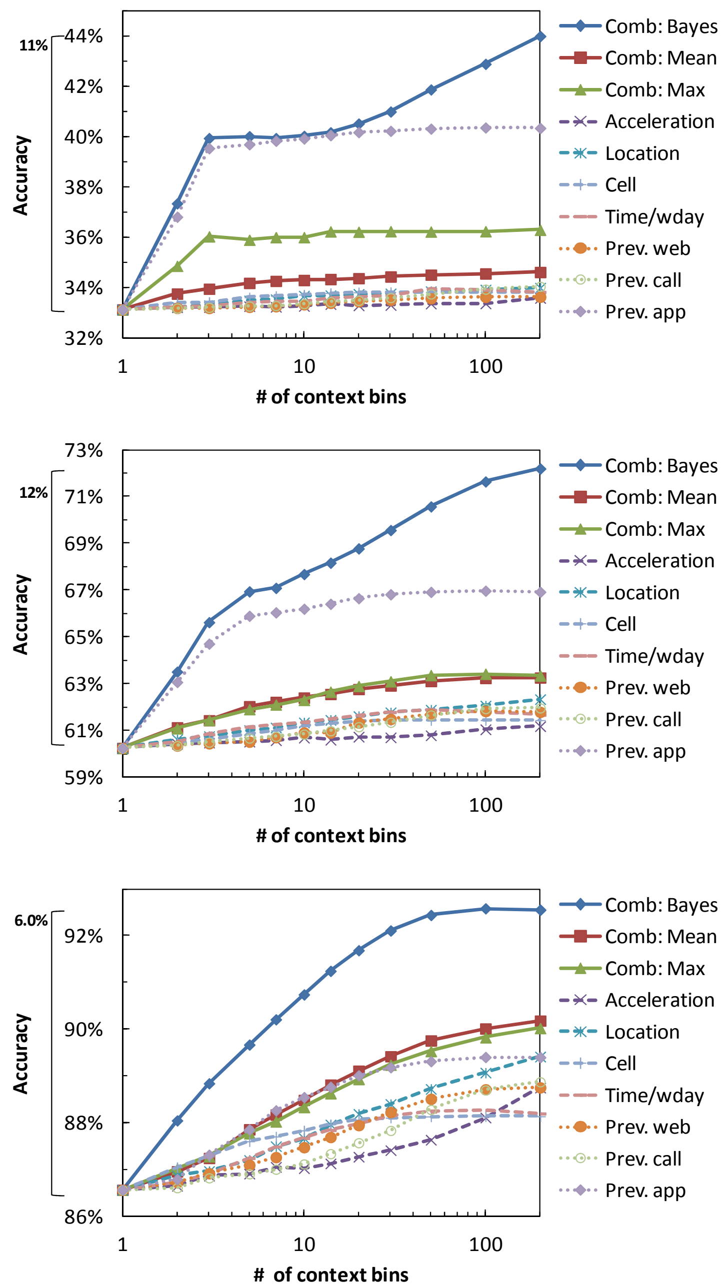


Figure 5.4 – Context dependency of app usage, presented as the estimation accuracy, for 1 (top), 3 (middle), and 10 (bottom) accepted responses

5.2. Combinations of Context

By observing the performance of the three types of usage, it can be sees that combination methods are very useful when a number of meaningful context sources are present. This dissertation finds that the max-rule consistently outperforms the mean-rule. The mean-rule is known to perform well when classifiers are noisy and highly dependent. Therefore, this dissertation concludes that that different context sources are not noisy or highly dependent. The surprising fact that Bayes often outperforms other combinations is another indicator that our context sources are not highly dependent and/or their dependencies are distributed evenly [76]. On the other hand, even though Laplace Correction is utilized to reduce the impact of data sparseness, the performance of Bayes is reduced when there are more bins (with fewer samples and therefore more noise in each category). Indeed, it is well known that Bayes is highly susceptible to noise.

These findings suggest that classifier combination must not be used on sources that show little or no context dependency, and that the max-rule should be used when there is a small context data set, but the system should switch to Bayes when the training data set grows. This can be identified by the system automatically, as when Bayes starts performing better than the max-rule.

Note that treating multiple context sources in a multidimensional manner resulted in practically no improvement in estimation accuracy, over the non-context case. Considering the number of samples per user in Table 3.1, the poor performance of multidimensional treatment of context sources is unsurprising. Even with only 10 context categories, there were significantly less than 1% of samples for any given context. More importantly, most samples belonged to categories with less than 0.1% of samples.

5.3. Seasonal Variation

The long-term LiveLab traces allow this dissertation to answer an important hypothesis regarding context dependency: how significant is the effect of the user's seasonal variation on the prediction accuracy. In other words, whether shorter durations may show higher context dependency, due to significant seasonal changes in user behavior, or would fewer samples in shorter durations reduce estimation accuracy.

The LiveLab traces indicate that while there are significant seasonal changes in device usage, as shown by the author's prior work [81] and highlighted in Figure 5.5, the context-dependency of usage remains relatively stable. To this end, this dissertation presents the same analysis as the prior section for the Naïve Bayes combination method, but instead of only calculating it for the entire 12 month duration of the study, it is calculated over one, three, and six month durations as well. For each duration, the best number of bins is selected, and results from the multiple periods are averaged, e.g. the results from the four 3-month periods are averaged together. The findings presented in Figure 5.6 indicate that such variations are small. Yet, the fact that the performance is not significantly reduced for shorter durations indicates that a smaller data set, e.g. one month or more duration, would have been sufficient for context-awareness.

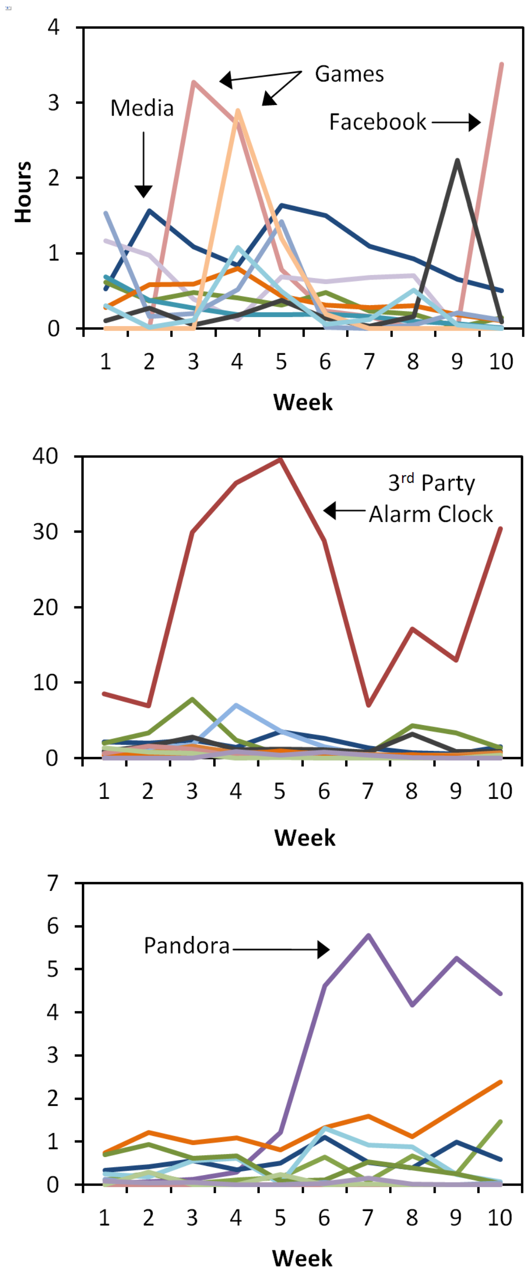


Figure 5.5 – App usage per week for three sample users shows a significant change in application usage amounts.

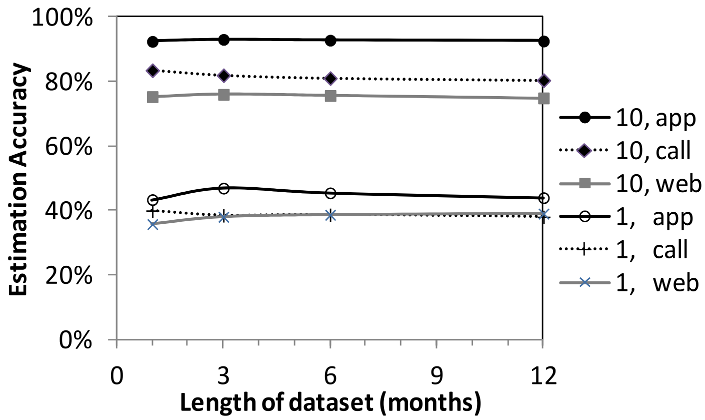


Figure 5.6 – Affect of seasonal variation of mobile usage on its context dependency is small, and one to three months of training logs is sufficient. Estimation accuracy for with one and ten responses, calculated on trace durations of one, three, six, and twelve months, using the Bayes method. (averaged among all users and periods).

5.4. Supervised Binning

As presented in Chapter 4.3.2, there is in inherent tradeoff present in choosing the number of context bins; more bins can allow finer molding of the bins, and more accurate posterior probability calculation, but would at the same time result in fewer samples per bin, increasing noise and reducing accuracy. Supervised Binning has the potential to make the best of both worlds, by identifying and binning together categories that have similar outcomes. This dissertation studies the efficacy of Supervised Binning, and shows it can greatly increase the accuracy of context based usage estimation.

This dissertation applies Supervised Binning as follows. For continuous context, it first discretizes it into ten times the categories, to allow for sufficient freedom for the binning algorithm while avoiding overfitting. It is computationally prohibitive to apply LOOCV for binning, as it would require recalculation of the binning for each test case. Therefore, two-fold cross validation is utilized, splitting the data into two six-month durations, and using the first six months for training and the second for testing, followed by the opposite.

Figure 5.7 shows that there is significant performance increase from Supervised Binning. The results are calculated using the Bayes combinatory method, since it produced the best results (Chapter 5.2). For consistency, the Supervised Binning results are compared to the simple binning case calculated using the same two-fold cross validation. The Supervised Binning results are presented using the Bayes combinatory method, since it produced the best results as shown in the previous sections.

One interesting question that is answered by this dissertation is whether it is necessary to perform supervised binning individually per user, or are there inherent features in context that are common between users, and can allow the supervised binning to be performed once for all users, i.e. using data from all users. The results in Figure 5.7 confirm the former: even for the small, relatively homogeneous population of this study, Supervised Binning using data from all users fails to improve accuracy over simple binning.

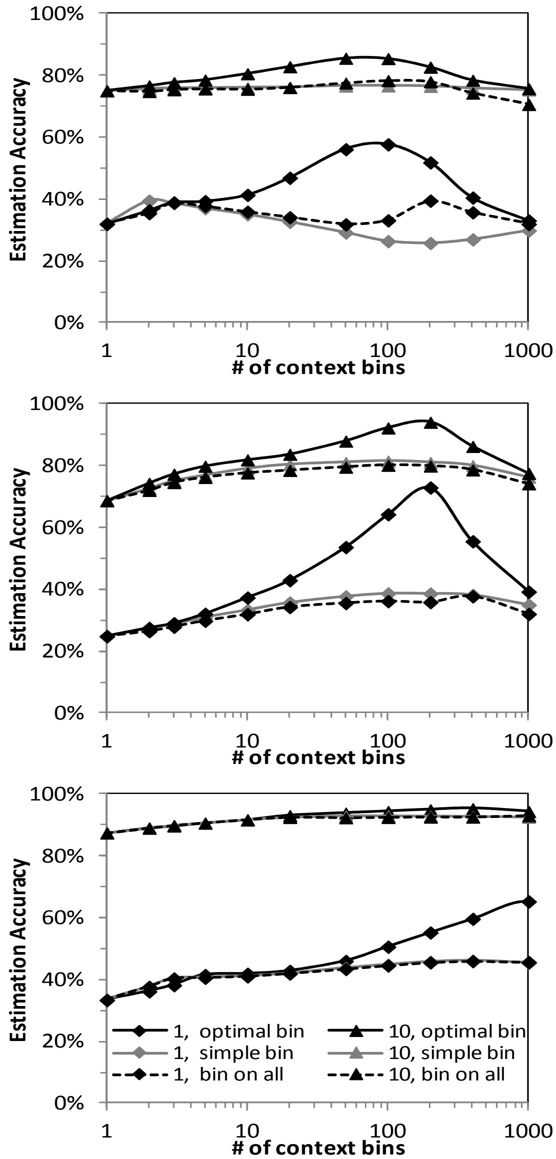


Figure 5.7 – Supervised Binning, individually for each user, can greatly increase the accuracy of context-based usage estimation. Estimation accuracy of web (top), phone call (middle), and app usage (bottom), calculated using the Bayes method, for one, three, and ten acceptable responses. Compared to Supervised Binning on all users' data, and individualized simple binning. One bin means no context information was used.

5.5. User Diversity

The LiveLab traces allow this dissertation to analyze the diversity in the context dependency of different users, i.e. whether some users have more diverse usage, and whether some users' usage is more context dependent.

To this end, Kernel Density Estimation (KDE) is utilized to present the distribution of estimation accuracy among the participants, and compare it to the case without context information (i.e. only one bin), for one and ten acceptable responses, as shown in Figure 5.8. The estimation accuracy is calculated using the best methods, i.e. Bayesian combination and Supervised binning, and the KDE bandwidth is empirically set to 0.05.

Figure 5.8 shows that while there is considerable diversity among participants' usage patterns, all of them show context-dependency in their usage. Furthermore, the top one and ten usage cases, which serve as the baselines for context dependency, constitute a significant share of usage for all participants. Finally, note that among the three principal usage, web usage had a much higher diversity of context-dependency, compared to its non context case. This shows that there is more diversity in the context dependency of web usage, compared to phone and app usage.

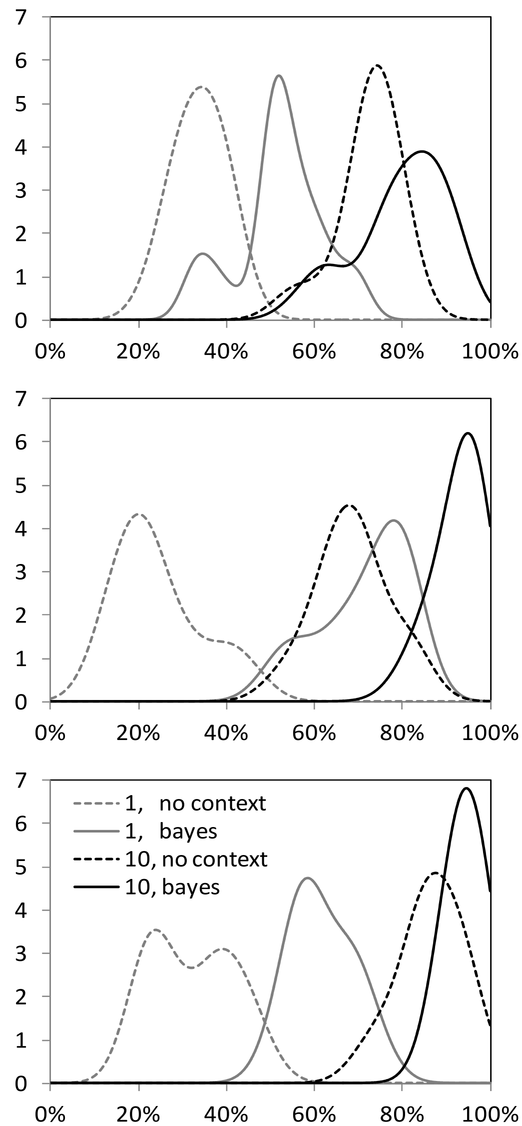


Figure 5.8 – Kernel Density Estimation (KDE) with bandwidth = 0.05 shows the distribution of estimation accuracy for one and ten acceptable responses among the 24 participants. Web (top), phone (middle) and app usage (bottom)

5.6. Prior Usage Context

The methodological classifier combination approach to the data sparseness challenge of context awareness, presented in Chapter 5.2, allows the use of prior usage context in addition to sensor context. Recall from Chapter 3.4.2 that usage context refers to the last used websites, phone calls, and apps, and sensor context refers to the measurements of the device's sensors, i.e. time&day, cell ID, motion, and GPS.

This section examines the effectiveness of prior usage context in increasing estimation accuracy. The depth of prior usage context is defined as how many prior usages are considered and combined along with the sensor context. A zero depth means no usage context is utilized. For the depth of one, as was presented in the prior sections, the last used prior usage for web, phone, and apps is used, similar to other sensor context. However, higher depth usage context can be treated in two ways. First, n-th depth usage context can be simply presented as yet another single dimension usage context, which will be handled and combined similar to other context, as described in Chapter 4. Second, is to treat each n-depth usage context as an n-dimensional vector. Compared to the first approach, this method suffers from sparseness, due to the curse of dimensionality.

Figure 5.9 shows the best estimation accuracy achieved through Bayesian combination and Supervised Binning, for different depths of prior usage context for both of these methods. It shows that after the depth of one, the gains start to diminish. This is due to the limitations of classifier combination; the additional information provided by depths higher than one cannot offset the error induced by their dependence across multiple depths. Also, for the multidimensional treatment of n-depth context, due to its sparseness for most of the samples, Laplace Correction simply returns the average a posteriori probability as the conditional a posteriori probability. In turn, this adds dependency and reduces performance. Based on these results, this dissertation concludes that its methodology may not be directly applied to historical usage context with a depth of greater than one. This limitation stems from inherent limitations of classifier combination for dependent data.

In order to better support historical usage context, for depth of greater than one, it is obvious that the dependency challenge needs to be addressed. Therefore, for thesis proposes and evaluates the following method for incorporating historical context: Treat usage context with a depth higher than one as multidimensional, but for each classification event and each type of usage, choose the highest depth that has more than *m* training samples. This method is referred to as auto-depth. The constant *m* is the minimum number of training samples deemed sufficient for an accurate a posteriori probability calculation. For the purpose of evaluation, this dissertation sets *m* to 10, based on the findings Chapter 5.1 that specified it as the rule of thumb minimum necessary number of samples per bin. The findings presented in Figure 5.9 show that the automatic depth selection approach works well, in terms of avoiding a performance drop, i.e. more information doesn't hurt. However, there is no measurably significant performance increase after the depth of one for the measured data.

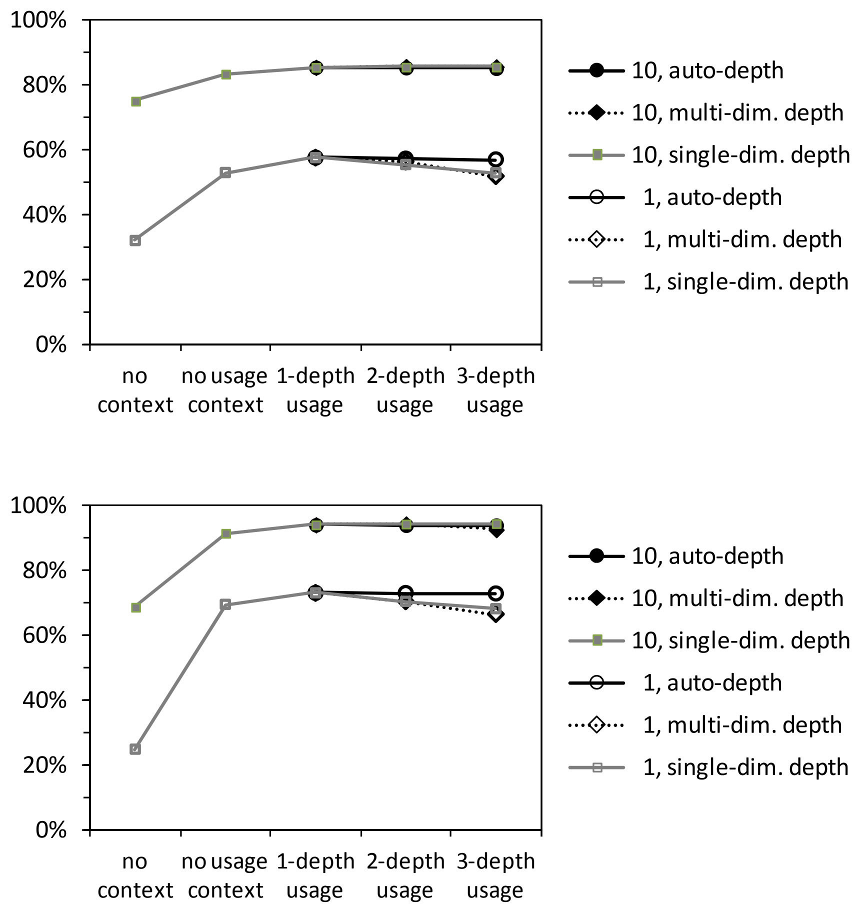


Figure 5.9 – Automatic depth selection is necessary to efficiently utilize prior usage context with depth greater than one. Effectiveness of prior usage context in increasing estimation accuracy, for web (top) and phone (bottom) usage. Best estimation accuracy achieved (through Baysian combination and Supervised Binning), vs. depth of prior usage context for one and ten acceptable responses, averaged among the 24 participants.

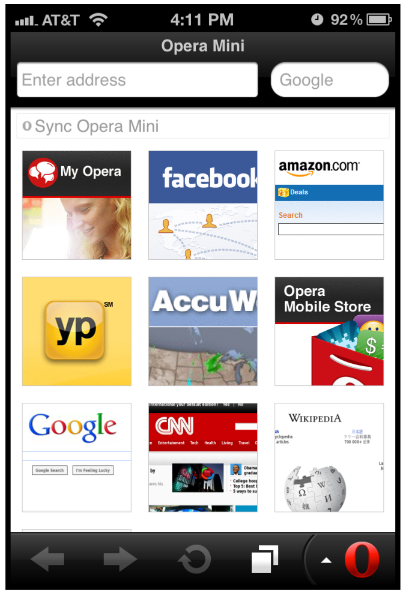
5.7. Sample Context-Aware Applications

The context dependency of mobile usage not only provides insight to the social behavior of humans, but can be utilized in many applications and services. In this section, a brief description of several such applications are provided, and their expected performance gains are evaluated based on the best performing methods presented prior, i.e. using Supervised Binning and Bayesian combination. These applications attest to the practical merits of context dependency, and the choice of estimation accuracy as the metric for context dependency.

5.7.1. **Web bookmarks**

It is known that a few websites account for most of the typical user's usage [66]. Accordingly, some browsers, e.g. Opera, offer a list of favorites or home screen that is configurable by the user or automatically generated, as shown in Figure 5.10 (top). This would provide with access to the user's most common websites. Others provide a user configurable home page that is automatically loaded when the browser is run.

A context-aware web favorites list can present more likely choices of websites to the user, according to their context. The findings presented in Figure 5.10 (bottom) show that a context-based solution for providing the user with either a single default home page, or a list of ten websites, significantly outperforms an ideal static selection, with a miss rate of 15% vs. 25% for a list of ten websites, and 42% vs. 68% for the single home page, and less than half the ideal static solution. Interestingly, the ideal static list of ten favorite websites outperforms the ten most recently visited.



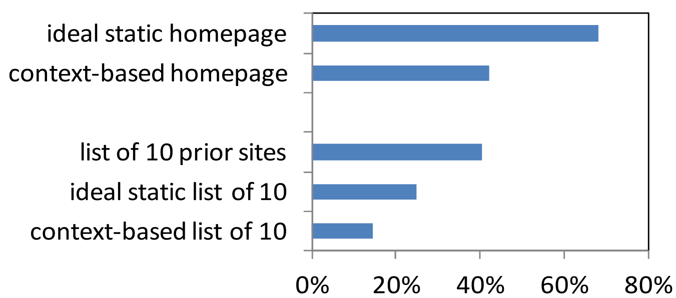


Figure 5.10 – Many browsers, e.g. Opera (top), display a list of bookmarks when launched. Others, show a single page, i.e. the home page. The performance of a context-aware web favorites list that presents the user with a list of most likely list of websites, or a single most likely homepage, is presented (bottom) as miss rates.

5.7.2. **Phone favorites list**

In order to assist users in making phone calls, phones typically provide the user with a redial button, a list of recent phone calls, and/or a user configurable favorites list, as shown in Figure 5.11 (top). For example, the iPhone used in the study provided a list of recent phone calls as well as a user configurable favorite contacts list. A context-aware phone favorites list can present a list of contacts the user is most likely to call, according to their context.

On average, a static list of each user's top ten contacts has a miss rate of 32%, and a recent call list has a miss rate of 28%. On the other hand, a context aware favorites list can reduce the users' need to go through their phonebook by approximately five fold, to 6%. Furthermore, the miss rate of a redial button is 64%, but the context-based dial button has a miss rate of 27%, as shown in Figure 5.11 (bottom).

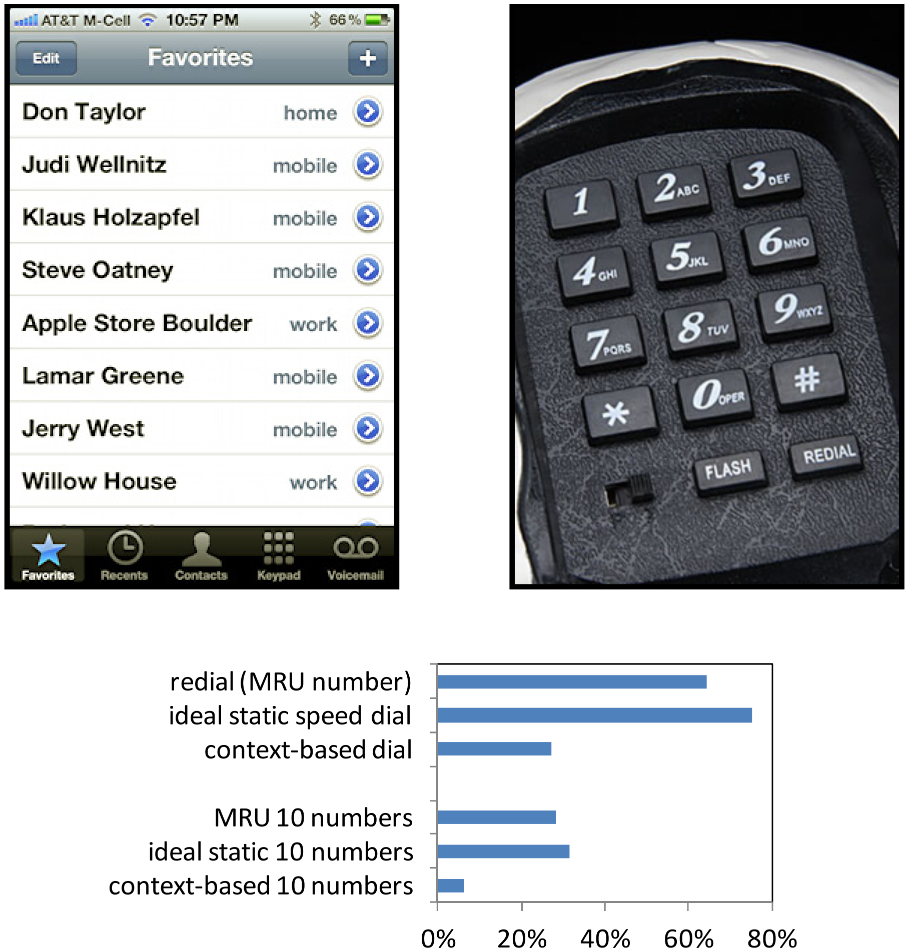


Figure 5.11 – A favorite phone contacts list is a modern development (top left), but redial functionality is much older (top right). The performance of a context-aware phone favorites list that presents one or a list of most likely phone numbers is presented (bottom) as miss rates.

5.7.3. **App Quicklaunch and Preloading**

Most phones often have a list of apps that are more readily available for users to run, i.e. quicklaunch, as shown in Figure 5.12 (top). The iPhone provides room for four such apps, which are readily available on any page of the home screen, and users can also organize their apps so the most common are placed in the first page. A context-aware quicklaunch list dynamically updates the quicklaunch list according to the users' context to contain the most likely launched apps. This dissertation shows that it would have a miss rate of 16%, compared to the 39% miss rate of the ideal static quicklaunch, an improvement of three times. For ten apps, the miss rate is just 4%, compared to 13% for the static case, as shown in Figure 5.12 (bottom).

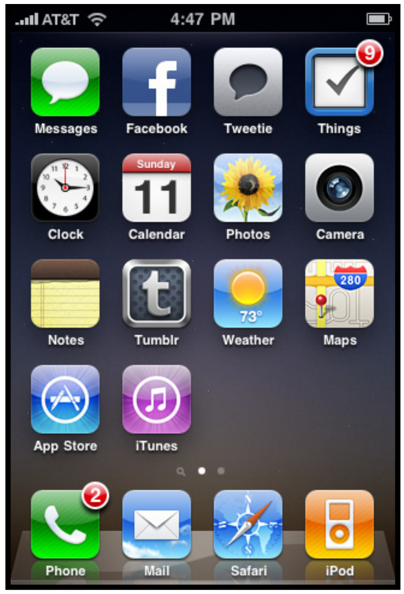
Preloading is another possible use of context, where context-based estimation of the app to be used can enhance performance. Preloading, including context-based methods have been widely studied in the past [82]. To this end, this dissertation has measured the app launch times on the substantially faster iPhone 4, presented in Table 5.1. The measurements were repeated three times for each app, and content load times are excluded when applicable. For the case without preloading, each app's process was manually terminated between the runs. For the preloaded case, each app was

Table 5.1 – Launch time for the 10 most popular applications (seconds), measured on the faster iPhone 4

Application	*Preloaded (in memory)*	*Without preloading*
SMS	0.6	2.2
Phone	0.6	always preloaded
Email	0.6	
Facebook	0.6	2.9
Safari	0.6	2.3
iPod	0.6	2.3
Alarm Clock	0.6	1.3
Settings	0.6	2.0
Maps	0.6	1.7
Notes	0.6	1.5
Average	0.6	2.0

started and closed before the measurement run. In such a case, the iPhone OS would keep the app in memory, i.e. preloaded.

Without preloading, the average load time was 2.0 seconds (median = 2.1, deviation = 0.5). With preloading, the load times were 0.6 seconds for all the measured apps. These measurements show that, on average, preloading can improve app load times over three fold. Note that the iPhone and many other platforms utilize a most recently used algorithm to keep multiple apps in memory, given memory limitations. This dissertation compares the performance of context-based estimation to the MRU algorithm, and shows that the miss rate for ten preloaded apps is improved from 9% to 4%, as shown in Figure 5.12 (bottom). Furthermore, for three preloaded apps, the miss rate is reduced from 31% to 17%, almost half.



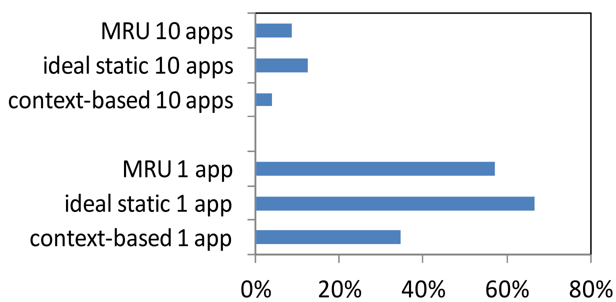


Figure 5.12 – For easier access, a number of apps can be placed on the quicklaunch home screen of the iPhone (top), or even better the always-visible bottom row. Similarly, the phone can pre-load a number of apps for faster launch times. The performance of a context-aware app list, or applcation preloading, that estimate the most likely launched apps is presented (bottom) as miss rates.

Chapter 6

Cost-Aware Context Combination

The findings presented so far attest to the performance and usefulness of context-based usage estimation. However, obtaining and processing context information often incurs significant energy costs. Many ad-hoc methods, sometimes themselves context based, have been employed to reduce energy cost of context awareness, while satisfying the system designer's cost-accuracy tradeoff. These methods typically reduce the frequency of accessing energy hungry context sources, or to avoid them altogether, substituting them entirely by lower cost context sources.

In this chapter, a methodological, application agnostic framework, SmartContext, is presented that addresses this challenge. SmartContext takes advantage of the methodology presented in Chapter 4, and builds upon the general problem of budgeted observation selection in the operations research community, to automatically optimize the energy cost of context-based estimation, while satisfying the accuracy requirements and tradeoffs that the designer sets for every estimation event.

6.1. Cost of Context

This dissertation focuses on energy costs of obtaining context information, and measures them independently. However, the SmartContext framework is applicable to other forms of costs as well, and the costs can be dependent or independent of each other. Furthermore, the energy costs can be measured automatically by the system in the software, as in [83].

The additional energy consumption incurred due to acquiring different forms of context are measured and presented in Table 3.1. This thesis defines the additional energy consumption of an activity as the extra energy consumption of that activity, compared to that of an idle phone. The measurements were performed on the iPhone 3GS, using the FTA22D Power Monitor, by Monsoon Solutions, Inc. [84]. To eliminate interference from the battery charging circuitry, we measured the power transferred to the phone

Table 6.1 – Ranking of context sources, and their energy cost

Type of context	*Rank*	*Energy cost*
Prior usage, time&day	0	negligible
Accelerometer	1	1.65 J
Cell ID	2	1.2 J
GPS location	3	50 – 300 J

from its battery connection, instead of between the charger and the phone. Each measurement was performed at least three times, and averaged.

6.2. Requirements and Operation

6.2.1. **Requirements and Assumptions**

SmartContext takes advantage of the methodological classifier combination approach presented in this dissertation. It is compatible with any classifier combination method, as long as the combination method can provide the MAP estimate as well as the estimation accuracy, using a) any combination of context sources, and b) with negligible processing overhead for *n* calculations, where *n* is less than or equal to the number of context sources. All three classifier combination methods evaluated in this dissertation have these features. However, this chapter utilizes the Bayesian combination method, as it performed best.

SmartContext selects context sources in order to meet an estimation accuracy set by context-aware applications and services for every estimation event. SmartContext requires that the training data set be available for all context sources, i.e. $P(\hat{g}_i|x_\mathrm{n})$ for all *x* and *i*. SmartContext requires the cost, or the expected cost, of utilizing each context source to be known in advance as well, as presented in Chapter 6.1. Note that the costs can be independent or dependent on each other. Further, the costs of some context sources are negligible. Therefore, they will be always utilized, limiting selection to context sources with non-negligible costs.

6.2.2. **Operation**

SmartContext's operation consists of two main steps. The first is determining the ranking of context sources. In order to keep processing costs in check, this ranking must be pre-calculated, but can be always static, or can be dependent on the context information gained at any step. In the next section we show that a static solution is both practical and performs well. In this case, the ranking needs to be performed only once. The second step is the energy aware combination of context. This has negligible overhead, and is performed dynamically for every estimation event according to the requirements and tradeoffs of the context-aware application or service.

Once the ranking is determined, the energy aware combination of context works as follows. For each classification event, SmartContext starts combining multiple sources of context information one by one, in the ordering determined in step one. Note that this can be done with minimum processing overhead, and for any combination of context sources, as explained in Chapter 4.2.2. After running the classifier combination with each additional context source, it checks the criteria of the requesting application or service, for that estimation event. In the evaluation presented here, a fixed minimum estimation accuracy for every estimation event is utilized. However, the application or service may set a different accuracy requirement for each estimation event, or even consider the expected cost of accessing the next context source, in determining when to settle with the current estimation accuracy and stop accessing more context sources. SmartContext assures the target estimation accuracy for each estimation event, as long as it is possible to reach that accuracy, while spending no excess cost in acquiring unnecessary context. In other words, in some conditions, no additional costly context is used, while in more uncertain conditions, SmartContext may use

```
DetermineCostPerformanceRanking(context_sensors)
ForEach (sensor) in (sorted_free_context) do {
    accuracy, usage = CombineNextContext(sensor)
}
ForEach (sensor) in (sorted_costly_context) do {
    If AppConditionMet(accuracy, usage, costs[]) {
        Exit Loop
    }
    accuracy, usage = CombineNextContext(sensor)
}
Return (accuracy, usage)
```

Figure 6.1 – Pseudo-code for SmartContext.

up to all the available context sources. The pseudo-code description of SmartContext is shown in Figure 6.1.

6.3. Ranking of Context Sources

The ranking of context sources is analogous to a well studied problem in artificial intelligence and operations research, which can be defined as follows:

How to select a subset, X, of possible observations (i.e. predictors or information sources) V, that most effectively reduces uncertainty and maximizes information gain?

6.3.1. **Review of existing methods**

Solutions toward this challenge are based on submodularity, an important property of the information gain from multiple observations [54]. Submodularity is also intuitively named as the diminishing returns property. It states that the information gain from an observation helps more if one has a smaller set of observations so far. Vice versa, the information gain from an observation helps less if it is added to a larger set. This can be formally presented as follows. The set function $F : V \rightarrow \mathbb{R}$ is submodular if

$$F(A \cup X) - F(A) \geq F(A' \cup X) - F(A')$$

for all $A \subset A' \subseteq V$, $X \notin A$, i.e. adding X to a smaller set, A, helps more than adding it to a larger set, A'. The general problem of maximizing submodular functions is NP-hard [51], and general algorithms are unable to provide guarantees in terms of processing time [52], unless there are certain assumptions, e.g. selecting a subset among a fixed tree ordering of possible observations [53]. However, artificially imposing such a dependency tree is heuristic in nature and can reduce performance, e.g. in [85]. Further, calculating the maximum-likelihood dependence tree requires assumes pre-measured mutual information between unit cost observations are available [86], neither of which are applicable to the case of this dissertation.

Therefore, it is common practice to use the greedy (myopic) solution towards this selection problem [47]. The submodularity property ensures that such greedy solutions are near-optimal, typically with provable constant factor performance guarantees. The greedy solution, assuming a unit cost for all observations, selects the observation with the most information at every step, i.e. the marginal increase $F(A \cup X) - F(A)$ is maximized. For this case, in [54], Nemhauser et al. prove that any set of equal-cost observations selected in this manner performs, at worst, a factor of (1 – 1/e), compared to the optimal set. More recently, the operations research community has proved the same bound when observations have different costs. Krause et al. prove that for independent costs, the greedy solution selects the observation with the maximum *cost-effectiveness* at every step, i.e. the marginal increase divided by cost of observation, $(F(A \cup X) - F(A))/cost(X)$, is maximized [47]. Furthermore, the same work proves that approximation algorithms are unable to provide guarantees better than a constant factor of (1-1/e), i.e. (k . (1-1/e)). We therefore base our work on the solution provided by Krause et al. [47].

6.3.2. **Ranking Mobile Context**

SmartContext is based upon the greedy method described in the earlier section, guaranteeing [47] a performance bound of $(1 - 1/e)$. However, the performance guarantee requires two assumptions. First, the costs of context sources (observations) must be independent from each other. Indeed, mobile context sources typically have independent energy costs, as was the case of this dissertation. Second, the submodularity or diminishing returns property must hold for our utility function (estimation accuracy). While this appears a reasonable assumption, it is necessary to verify it.

Existing work typically use entropy as their utility function, and either assume that it is submodular [54], or prove that it is submodular under an assumption of independence [47]. This dissertation experimentally verifies the overall submodularity of estimation accuracy. For this purpose, it is necessary to show that the estimation accuracy gain resulting from adding (combining) any context source decreases if more context sources were known (combined) beforehand. Note that since SmartContext assumes that free context is always utilized, it is necessary to verify the submodularity only among costly context. Figure 6.2 shows the estimation accuracy gain for Cell ID, acceleration, and GPS location.

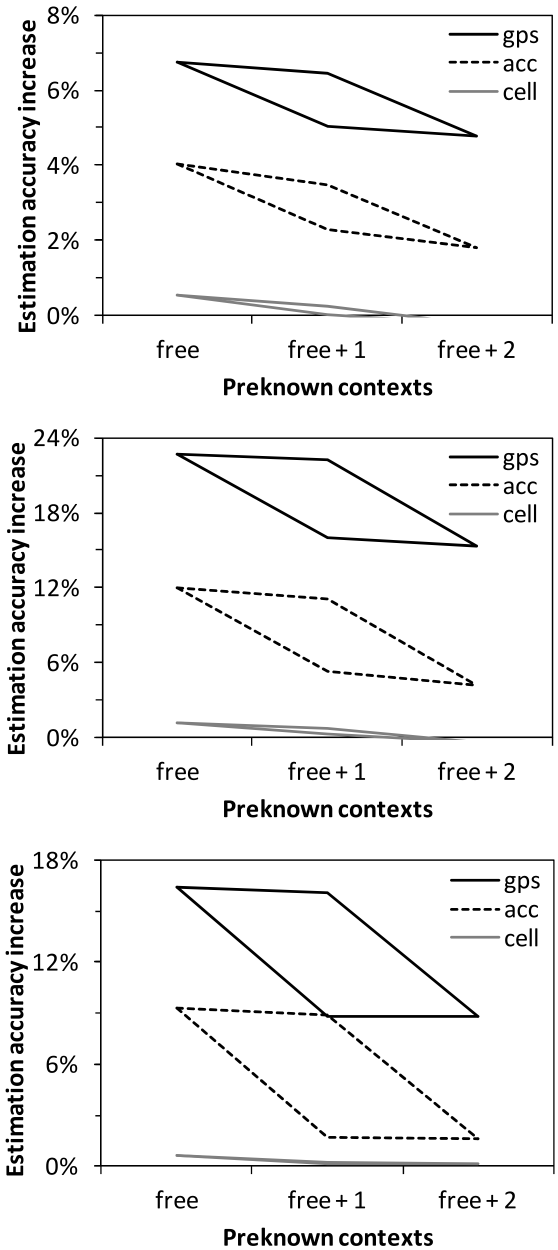


Figure 6.2 – Submodularity of estimation accuracy; the average estimation accuracy gain resulting from combining a certain context decreases if more contexts were known (combined) beforehand. Free indicates time&day and previous usage. Free + 1 and free + 2 indicate one or both of the remaining two contexts (except the one to be added). Top: web usage. Middle: phone call usage. Bottom: app usage.

Therefore, this dissertation concludes that the greedy approach works well for context awareness. In this case, the best ordering is obtainable by ranking the context sources according to their cost effectiveness. In this case, the cost effectiveness of each context source is the marginal estimation accuracy increase divided by its expected energy cost, i.e., $(F(A \cup X) - F(A))/cost(X)$.

The expected energy cost can be pre-measured by the system designer, as in the case of this dissertation, or can be measured automatically in the software as in [83]. The energy costs of acquiring context on the iPhone 3GS are presented in Chapter 6.1 and Table 3.1. This desertion measured the estimation accuracy of each context source individually in Figure 6.2. Note that assuming x_1, … x_a are context sources that have been acquired so far,

$$
\begin{aligned}
&F(A \cup X) - F(A) \\
&\qquad = \operatorname*{argmax}_i P(g = \mathrm{i} | x_1, \dots, x_a, x_X) P(x_1, \dots, x_a, x_X) \\
&\qquad - \operatorname*{argmax}_i P(g = i | x_1, \dots, x_a) P(x_1, \dots, x_a)
\end{aligned}
$$

The resulting ranking is shown in Table 6.1. Note that due to the often significant difference in the energy cost of context sources on mobile devices, their ranking becomes close to, or according to their energy cost.

Finally, we note that due to the relatively limited number of costly context sources on mobile devices, it is also possible to simply perform a thorough search, calculating the performance of SmartContext under all possible orderings of context sources. For the case of this thesis, there are three costly context sources, resulting in a total of $3! = 6$ possible rankings. Unsurprisingly, the rankings obtained using the through search under each of the experimental cases of Chapter 6.4 are, in fact, the same as the greedy ranking calculated in this section.

6.4. Evaluation of SmartContext

SmartContext has been evaluated using the LiveLab traces. Figure 6.3, Figure 6.4, and Figure 6.5, show, for web, phone, and app usage, how often each context source is utilized for different (minimum) target accuracies, and how often the target accuracy is achieved, as well as the overall average achieved estimation accuracy. Note that SmartContext always uses time&day and previous usages, as they are available without cost. The additional accuracy provided by each additional context source is incremental and diminishing. This is expected, as each individual source of context has a small incremental value, as shown in Chapter 5.1, and as submodularity ensures diminishing returns, as shown in Chapter 6.3 and Figure 6.2.

Yet, this dissertation shows that significant energy savings are possible with very little sacrifice of accuracy. For example, for web, phone, and app usage respectively, for one acceptable response, setting the (minimum) target accuracy to 25%, 50%, and 50%, achieves 89%, 67%, and 61% energy saving, while providing overall estimation accuracy within 1% of the case using all context sources. For ten acceptable responses and 75%, 80%, and 85% (minimum) target accuracies, respectively, the energy savings are 71%, 65%, and 89%, while again achieving overall estimation accuracy within 1% of the case using all context sources.

Note that as the ordering of context, and the posterior probabilities are pre-calculated using the training data, e.g. during charging, they do not add to the overhead of SmartContext during regular operation. Further, the combination algorithms require little processing, therefore SmartContext has negligible overhead.

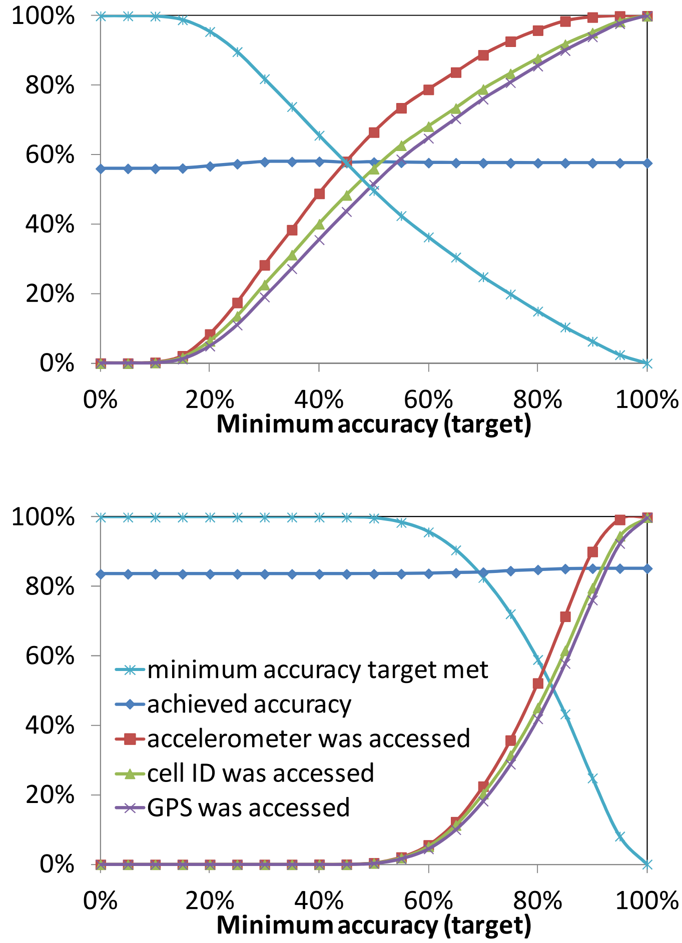


Figure 6.3 – Performance of SmartContext for web usage, for 1 (top), and 10 (bottom) responses. The figure shows, for a range of minimum accuracy targets, how often costly context is accessed, how often the minimum accuracy target is met, and the overall average estimation accuracy.

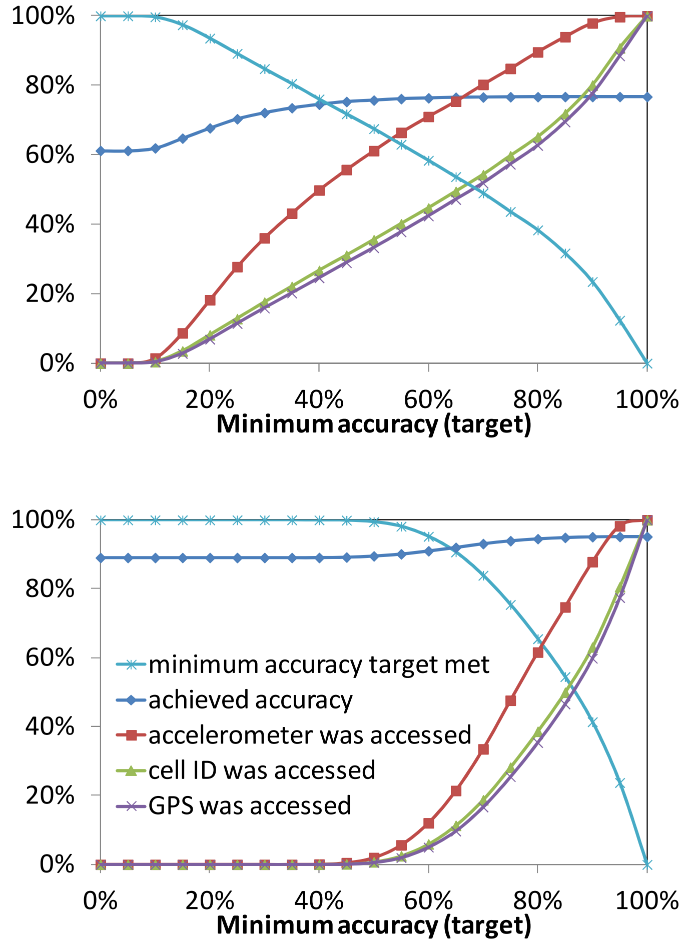


Figure 6.4 – Performance of SmartContext for phone usage, for 1 (top), and 10 (bottom) responses. The figure shows, for a range of minimum accuracy targets, how often costly context is accessed, how often the minimum accuracy target is met, and the overall average estimation accuracy.

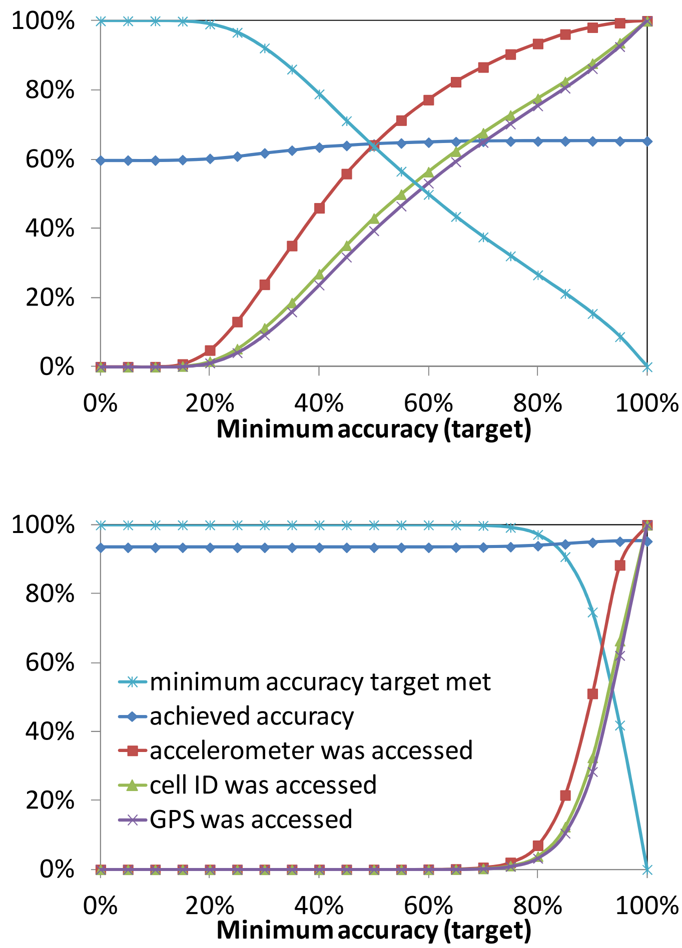


Figure 6.5 – Performance of SmartContext for app usage, for 1 (top), and 10 (bottom) responses. The figure shows, for a range of minimum accuracy targets, how often costly context is accessed, how often the minimum accuracy target is met, and the overall average estimation accuracy.

6.5. Energy Aware Training

The findings presented so far attest to the effectiveness of SmartContext for methodologically managing the energy cost of context awareness, while satisfying the application or service designer's dynamic accuracy requirements and tradeoffs by selecting a subset of context sensors to acquire and combine, as described in Chapter 6.2. Often, gathering training data also encounters a significant energy challenge as well. In this section, this dissertation discusses the both the advantages and limitations of its methodology for managing the energy costs of context awareness during the training stage. In other words, whether and how it is possible to expand the methodology of this thesis, and in particular, SmartContext, to selective gathering of training data from a subset of context sources, in order to minimize training cost while satisfying specific performance constraints.

6.5.1. **Training During Classification**

One of the fundamental benefits of the context combination methodology presented in this thesis is that it inherently supports the opportunistic collection of training data during context-based classification events, without additional cost. This includes classification events that utilize a subset of context sensors to save energy, i.e. SmartContext.

The reason that such a feat is possible is that each training sample does not require readings from all context sources to be present. Instead, the context combination methodology of this dissertation learns the a posteriori distribution of each context independently from other contexts, using classifier combination methods (i.e. Bayes) to calculate the overall a posteriori probabilities. Accordingly, as long as the ground truth outcome (usage), g, becomes available any time after a classification attempt, it can be used alongside the m measured contexts, x , … x_m, out of a total of n contexts, as training samples, $\{g, x_1\}$ … $\{g, x_m\}$ to improve the training data set for each of the m contexts that were utilized by SmartContext for the respective estimation. Contexts $x_{m+1}, \ldots, x_n$ are neither measured nor utilized for training.

As an example, recall the context-aware phone favorite contacts list of Chapter 5.7.2. Assume that at one particular time that the favorites list is launched, SmartContext utilizes only Cell ID and the free context sources to generate the smart favorites list. After the user dials their intended party, g, this ground truth usage is stored along with the recorded context $x_1, \ldots x_m$,. The a posteriori distributions of phone calls are later updated accordingly. In fact, after increased training observations, there is a possibility that SmartContext may not even need Cell ID to satisfy the system designer's accuracy requirements for an identical case in the future.

6.5.2. **Creating the initial training set**

The previous section shows that every classification event that can be associated with an actual outcome (ground truth usage), can be utilized as a training sample. However, this is under the assumption that the binnings of all context sources has already been calculated. On the other hand, the Supervised Binning of context sources, shown in Chapter 5.4 to greatly improve estimation accuracy, requires a reasonably sized training data set of that context source. In other words, at the very least, there should be complete training data until an accurate binning can be formed. Chapter 5.3 suggests that as little as one month of training data can be sufficient. After sufficient training data is obtained to form bins for each context source, training can continue opportunistically, as described in Chapter 6.5.1.

In retrospect, the authors acknowledge that collecting readings from all context sensors for every single classification event may be too costly for the system. In this case, there is always the option of limiting the energy costs of training by trading training frequency with duration. For example, by collecting training samples only half of the times, but over twice as long a training duration. Again, one of the important benefits of the methodology presented in this thesis is that the system can operate, albeit with a lower estimation accuracy, on only a subset of context sources. For example, training data for binning all contexts except GPS can be collected over one month, but GPS training and binning can occur after three months, with a third of the energy consumption. In the mean time, classifier combination and SmartContext can simply operate using context sources other than GPS, with slightly reduced estimation accuracy.

Chapter 7

Social Context of Mobile Usage

The previous chapters of this dissertation focused on context information acquired from built-in sensors and the users' prior usage. This chapter investigates a different form of context, *social context*, to measure its effect on mobile usage. To this end, the LiveLab users are separated into two carefully selected distinct socioeconomic groups, and their collective usage patterns are analyzed and compared.

The findings of this dissertation, as well as previously reported research confirm that smartphone users are extremely diverse, e.g. by orders of magnitude [59]. Understanding user diversity has been central tenet of human-computer interaction (HCI) research [64]. Yet, little research has moved towards understanding these differences in more precise ways. These findings provide researchers and system designers with a better understanding of how users vary, resulting in better support of a broader range of individuals with different backgrounds, capabilities, skills and interests.

7.1. Diversity and Dynamics of Usage

This section takes a deeper look into the diversity of mobile usage by analyzing the diversity and dynamics of app usage, in terms of adoption and usage for both built-in and App Store apps.

7.1.1. **App Usage**

The 24 LiveLab participants installed over 3400 apps during the yearlong study, of which over 2000 were unique. They also purchased almost 750 apps, of which 500 were unique, from the Apple App Store, spending over $1300. Surprisingly, more than half (62%) of the 3400 apps installed by the users were uninstalled during the study.

This dissertation defines the *lifespan* of an app as the time between its installation and its uninstallation. Many apps have a short lifespan, e.g., 20%

Table 7.1 – The following categories were manually assigned to applications, in part based on genres reported by the App Store

Category	*Genres*	*Notes*
Games	Games, Entertainment, Media	Entertainment and media consumption
Utilities	Utilities and Productivity	Calculators, alarm clocks, to-do lists
Reference	Books, Education, and Reference	Information resources
News	News, Sports, Travel, Weather	Contemporaneous information resources
Commerce	Business, Finance, Lifestyle (shopping)	Shopping or financial apps
Social Networking	Social Networking	Facebook, MySpace, Twitter
Other	Health, Navigation, Medical, Photography	Only a few (162) applications

uninstalled within a single day and 31% within two weeks, indicating that the user tried the apps but disliked them.

In order to analyze and more importantly present such a huge data set for behaviors and trends, it is necessary to assign categories to apps, as shown in Table 7.1. The App Store already reports 20 genres for apps, but to the inconsistency of App Store genres and the fact that a certain app may be tagged by multiple genres, this dissertation had to carefully and manually categorize them.

The most popular app category was games, accounting for over 50% of app installs and over 50% of money spent, and approximately 5% of app usage. In contrast, social networking apps, mostly being free, only accounted for less than 2% of money spent, but accounted for 8% of app usage. As expected, there was a wide variation between users in adopting paid and free apps. The users spent a median of $25 on 14 apps, as shown in Figure 7.1, and all but two users purchased at least one app. Figure 7.2 shows the total number of adopted apps during the study, broken in to built-in, free and paid apps. The ratio of paid to free apps stays relatively constant over time, at around 20%. Interestingly, and somewhat surprisingly, paid apps had a shorter lifespan overall, as shown in Figure 7.3. The large number of paid apps with one day lifespan shows that users frequently purchase apps which they quickly determine they dislike, losing money in the process. The larger number of paid app uninstalls in the next months can be attributed to the large number of paid games

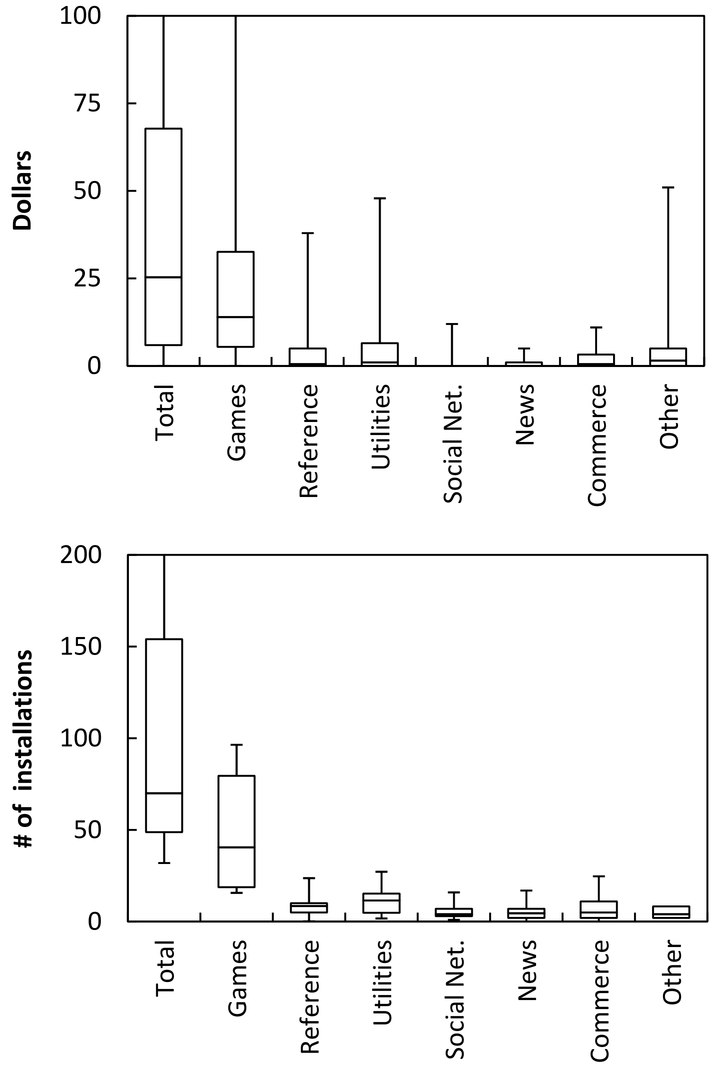


Figure 7.1 – There is significant user diversity in app installation in terms of number (top) and price (bottom). Broken down by category. Boxes: 2nd / 3rd quartiles. Whiskers: maximum / minimum. Horizontal lines: median

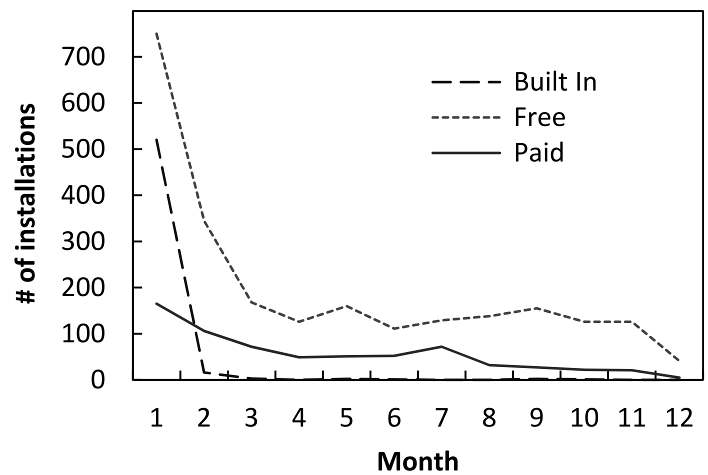


Figure 7.2 – The ratio of paid to free app installations remained steady through the study, at ~ 20%.

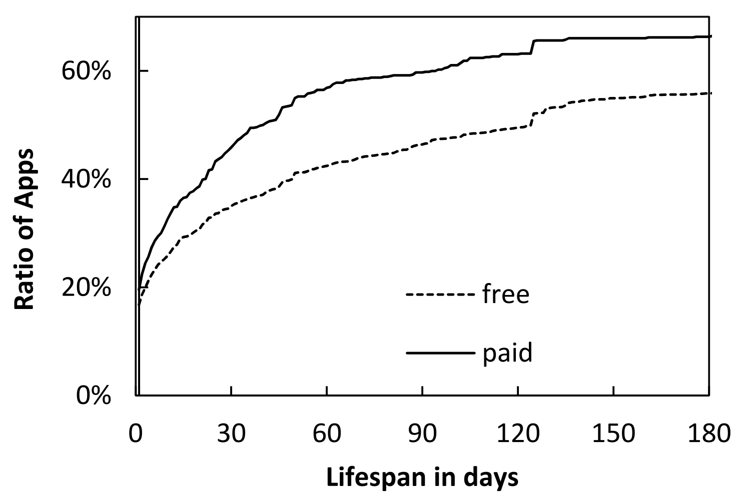


Figure 7.3 – Paid apps have a shorter lifespan compared to free apps

Even though categorizing apps allows the analysis of app usage of the users, there is still a significant variation between app usage duration and frequency among different users. The significant differences between users, even among the second and third quartiles, highlight the fact that the average or median user alone is unable to serve as a benchmark for mobile usage. Instead, it is necessary to consider a wide variation of users and usage. Figure 7.4 shows the boxplot of app usage by different users, for both frequency (Left) and duration (Right). The box indicates the second and third quartiles among users. The whiskers indicate the maximum and minimum values among users. The horizontal lines inside the boxes indicate the users' median.

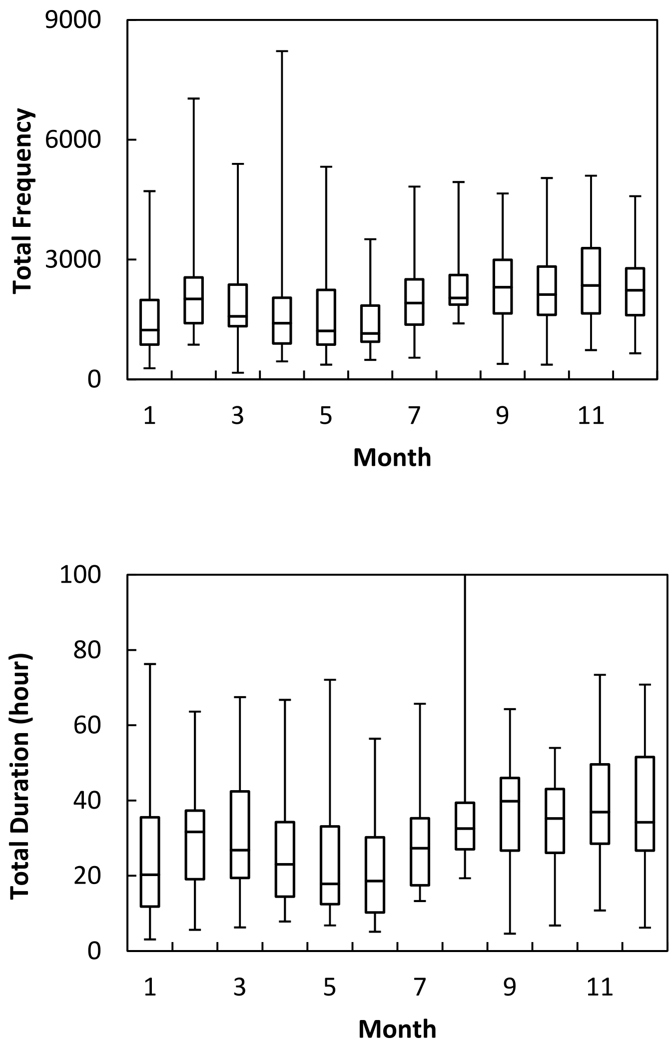


Figure 7.4 – Ap usage very diverse throughout the study, in terms of both frequency (top) and duration (bottom). Boxplots show 2nd / 3rd quartiles. Whiskers: maximum / minimum. Horizontal lines: median

7.1.2. **Web Usage**

Similar to app usage, each user's web usage converges to a small set of websites. As shown in Figure 5.1, the top website of a user accounts for 28% of web usage (median); and the user's top ten websites accounts for 87% of usage. Compared to app usage, users were more inclined to explore new websites than apps, which is intuitive since visiting a new website requires much less commitment and time than installing an app. The key supporting evidence is the median month-to-month Cosign Similarity of web usage was significantly lower than that of app usage, at (0.73 – 0.94), compared to (0.85 - 0.97) respectively. To calculate the Cosign Similarities, this dissertation defines a multidimensional web usage vector so that the magnitude of each element corresponds to the frequency a specific website or app is launched. The Cosine Similarity is the Cosine function of the angle between the two usage vectors, and its output is always between 0 and 1. In mathematical terms

$$A.B = \|A\|\|B\| \cos\theta$$

Therefore, Cosine Similarity ($S_{A,B}$) can be calculated as

$$S_{A,B} = \cos\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i . B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} . \sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

where A_i and B_i is the amount of usage type i for the users A and B respectively.

While iPhone apps are developed for a smartphone environment, and are often tailored to the specific features of the smartphone platform, it is expected web browsing to be an extension and supplement to users' regular browsing. The findings in this chapter support this hypothesis, but strongly suggest users are disappointed with their web browsing experience; contrary to app usage, there was a significant decrease in participants' web usage throughout the study. This dissertation hypothesizes that the disappointment was due to the shortcomings of the browsing experience, and the fact that the participants had regular access to PC for web browsing. Indeed, compared to PCs, the phone's screen is much smaller and the web browsing experience is significantly slower (caused by higher connection latencies [87]).

In order to further study the decrease in web usage, this dissertation analyzes the web content browsed by the users. The participants access both mobile and non-mobile websites. This dissertation classifies web pages based on URL keyword matching, e.g. URLs that "m.", "mobile.", "iphone.", etc. are classified as mobile. Some popular websites, such as google.com, use the same URL for both mobile and non-mobile versions. In those cases, it is assumed the mobile version was used. Mobile web pages are less content rich than their non-mobile counterparts, in terms of styles (CSS), scripts (JS), multimedia content (IMG), and HTML size, as shown in Figure 7.5. Overall, the phone had to download 120KB for the typical mobile page and 3 times more, or 360KB, for the non-mobile page.

The results in Figure 7.6 indicate that the drop in web usage was due to a drop in non-mobile website usage, while mobile website usage, presumably better fit for mobile devices remained relatively stable through the study. The clear message for designers is to develop a mobile-version of any content that could be accessed through a browser. Indeed, this is more important for users in lower SES brackets and new smartphone users as they transition and learn how to install apps.

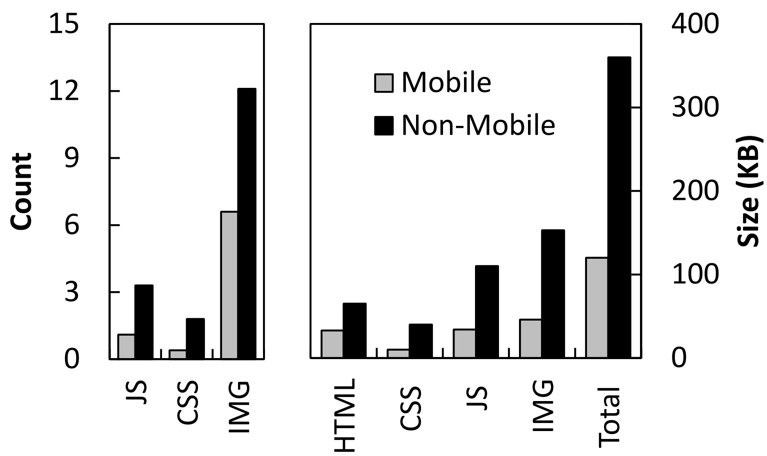


Figure 7.5 – Mobile web pages are less content rich, in terms of the number of resources (right) and their sizes (right)

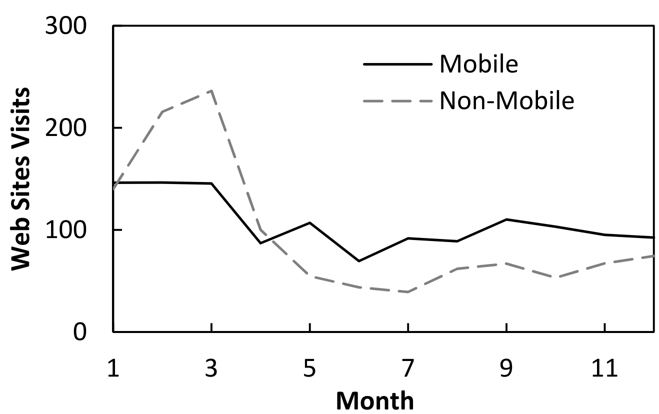


Figure 7.6 – Median visits to mobile and non-mobile website per month

7.2. Effect of Socioeconomic Status (SES)

This section assesses the differences between SES groups in overall usage of their iPhones. The author had expected differences to be minimal, e.g. how much they spent in App Store purchases, as both groups lived in the same dormitories on campus, and had no significant bias in their gender, major, PC access, or game console ownership. Surprisingly, the findings suggest stronger and broader differences in how they used their devices.

7.2.1. **App Usage**

App usage was consistently higher in the low SES users, approximately 40% more than high SES users in terms of both frequency and duration, as shown in Figure 7.7. For control, the differences between the low and high SES groups were assessed over the entire study period of one year. Visits to apps and the web were combined for each user within each quarter. A 2 (SES: Low vs. High) x 4 (Time: quarters) analysis of variance (ANOVA) revealed that low SES users visited apps on their smartphone 40% more than their higher SES peers ($F(1, 22) = 9.73$, $p = .01$). A main effect for Time or interaction effect was not found. The low SES users also consistently used a more diverse set of apps throughout the study, as shown in Figure 7.8 by the top ten apps' smaller fraction of usage. The diversity is in part due to the low SES participants' higher variety of games used. Overall, the higher device usage and app variety in low SES users suggests that the iPhones are used for hedonic and utilitarian reasons by the low SES group and just the latter for the high SES group. It appears this may be due to low SES users having fewer or less interesting *outside options*, including those for entertainment or otherwise.

Figure 7.9 is a radar chart showing app usage for each SES group, for the top ten apps or app categories, normalized to the overall average usage of each app or app category. Four of these apps or app categories revealed how SES groups differed: Facebook, phone, games, and utilities. Logistic regression confirms this statement; this desertion compared the standardized logistic regression coefficients of each app or app category, as suggested in [88] to find the dominant predictors of SES. The results show that the top 3 dominant apps in frequency are utilities, games and phone; and top 3 in duration are Facebook, games and utilities, which comprise the exact same four apps.

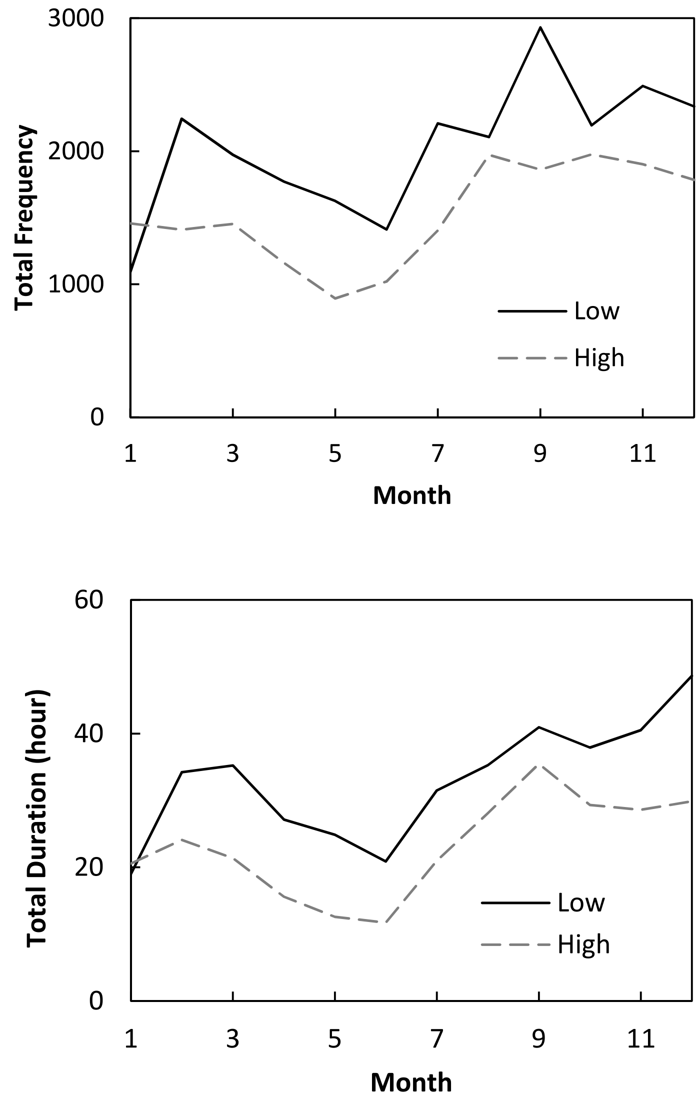
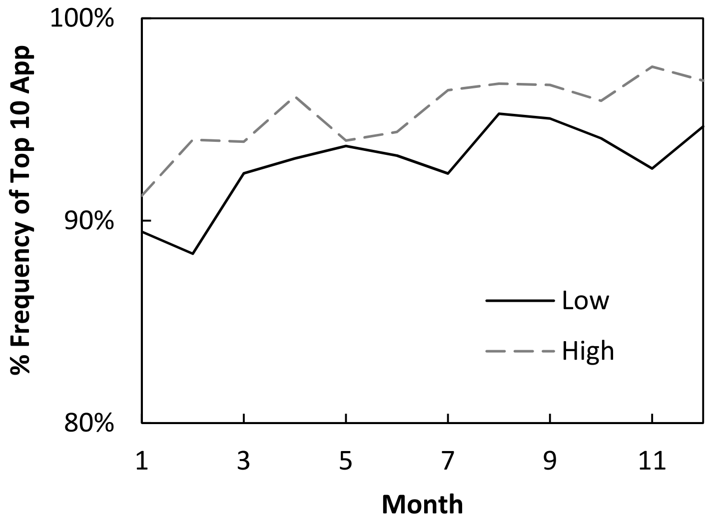


Figure 7.7 – Median app usage was higher for low SES participants, in terms of both frequency (top) and duration (bottom)



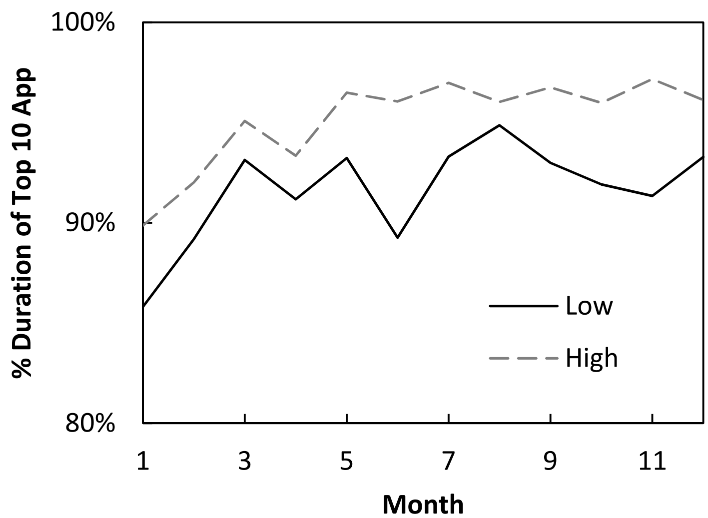


Figure 7.8 – The top ten apps contributed to a larger fraction of usage for high SES groups, in terms of both frequency (top) and duration (bottom)

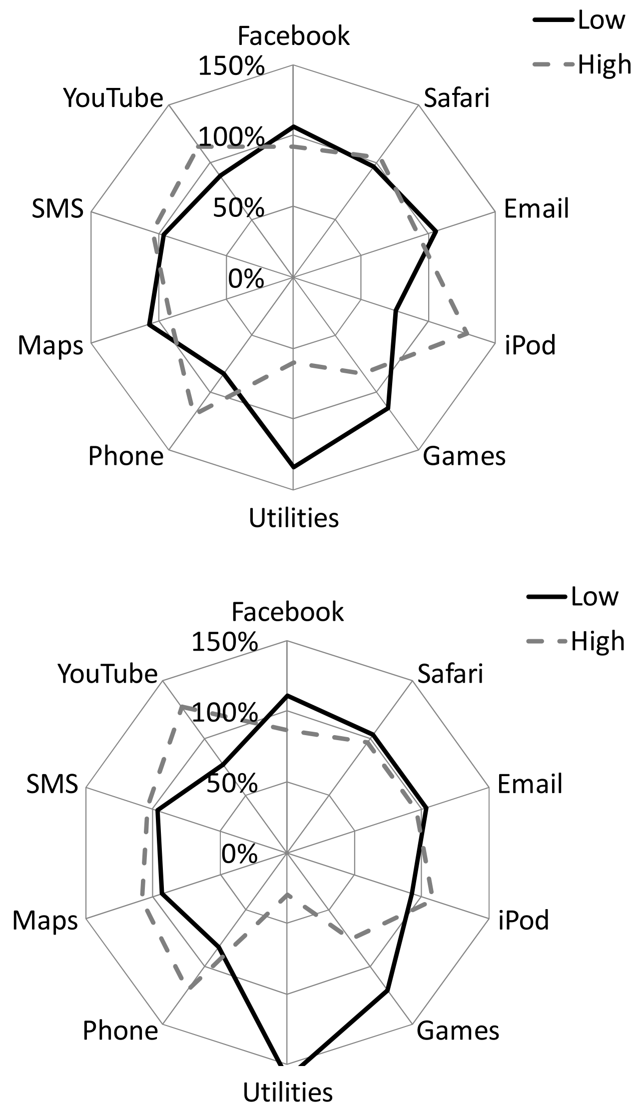


Figure 7.9 – App usage, relative to each app's average usage, for both SES groups in terms of frequency (top) and duration (bottom)

7.2.2. **Web Usage**

By comparing the SES groups, it can be seen that Web usage was initially much higher in the low SES group. However, the usage of both groups dropped, and their differences disappeared through the course of the study, as shown in Figure 7.10. Similar to the prior subsection, ANOVA was used for URL visits. Interestingly, it revealed that there were no main effects for SES or Time. However, a significant interaction showed that the lower SES group accessed the web more at the beginning of the study; over time, however the differences between SES web use attenuated, $F(3, 66) = 4.60$, $p = .01$. Correspondingly, duration of use followed similar patterns. Low SES users spent more time on the web early on; however, differences between SES groups diminished as a function of experience.

While iPhone apps are developed for a smartphone environment, and are often tailored to the specific features of the smartphone platform, it is expected web browsing to be an extension and supplement to users' regular browsing. The higher initial usage for low SES participants shows that, for them, smartphone web usage is more of an extension to their PC-based web access. Indeed, user interviews with both groups suggest that even though they both had access to personal and university PCs, the lower SES group owned older and lower-quality computers.

In contrast with app usage, both SES groups had similar diversity in web usage, as shown in Figure 7.11. The similarity can be attributed to the participants' previously established web browsing habits.

7.2.3. **App Store Purchases**

The author had expected high SES participants to spend more on paid apps, but was surprised to find the opposite. Low SES users spent a median of $31 on a median of 17 apps, compared to $15 on 6 apps for the high SES users. In other words, they spent approximately twice the amount of money on three times as many apps.

Looking deeper into the data, it can be seen that low SES users were more money conscious and presumably more careful in their purchases compared to high SES users. This is shown by their significantly different prices paid per hour usage of paid apps. By dividing the total each user spent in the App Store by the total paid app usage duration, one can calculate the cost per hour for paid apps (i.e. price divided by duration). The low SES users had significantly lower prices paid per hour (median: $1.0 vs. $2.6), which is substantial even considering the increased overall usage of the low SES users.

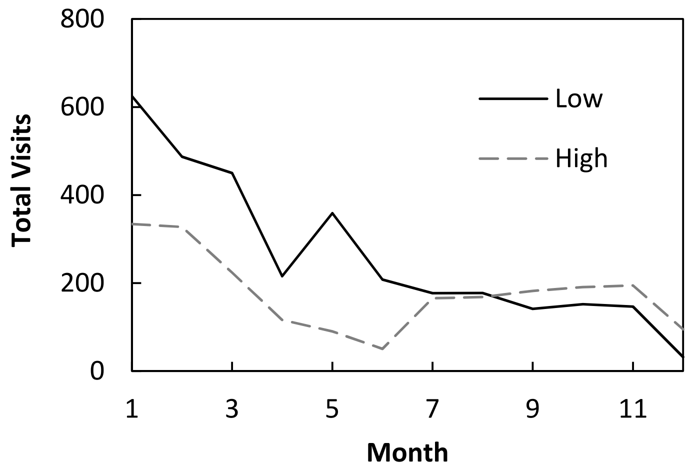


Figure 7.10 – Median web usage was initially higher for low SES users, but became similar to high SES users

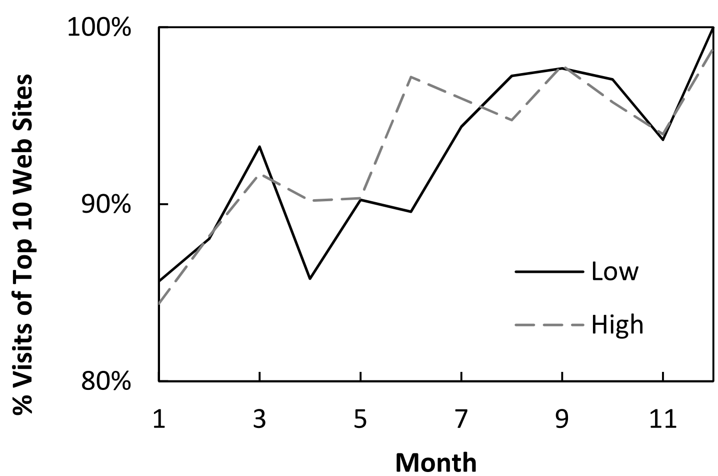


Figure 7.11 – The top 10 websites contributed to a similar fraction of usage for both SES groups

7.3. Implications of Socioeconomic Differences

The past section showed that there are clear usage differences based on SES levels, even by controlling the user experience, i.e. the type of device, temporal context, and other demographic factors (e.g., all students at the same university, age, gender, etc.). This section elaborates on the implications of the usage differences on the design and evaluation of mobile devices and apps.

7.3.1. **App Development**

The findings regarding app lifespan (Figure 7.3) provide insights into promoting third-party smartphone apps. They show that users often try out apps for short periods, e.g. a day. Unfortunately, neither the Apple App Store nor the Android Market offers try-before-you-buy as a universal feature. Instead, users are typically expected to purchase apps based on reviews and word of mouth. However, this dissertation clearly indicates that users would benefit from a try-before-you-buy feature, such as the one introduced by the recent Windows Phone 7 platform. This would enable users to waste less money, as well as potentially explore and purchase more apps. Additionally, real estate on iPhones is important and a try-before-you-buy store can facilitate users to quickly "clean house" if an app is not useful or engaging

Some operating systems, such as Windows Phone have already developed this feature. The usage traces showing higher month-to-month diversity in web usage (Figure 7.11), highlights the fact that smartphone users are more comfortable exploring websites and web applications than downloading apps. Indeed, it is natural for users to be more adventurous in accessing different websites than using apps; visiting a website takes much less commitment than installing an app. This suggests that an application developer could reach a larger audience by providing a web service similar to its installation-based app when appropriate, so that first-time users can more easily assess them.

7.3.2. **Designing Mobile Content**

Many have envisioned feature-rich smartphones that provide cost-effective access to information technologies and entertainment, especially for users from underserved communities. This was one of the key motivations for focusing on socioeconomic status (SES). The results of this chapter support this vision: users with low SES tend to use smartphones more frequently and for more time than high SES users (>40% more). Clearly, the web browser is more central to supporting low SES users adopt smartphone technology. When they first received their smartphones, low SES users seemed to use mental models developed through PC or laptop use. This manifested in an increased reliance on their mobile browsers.

Over time, however, this reliance diminished in favor of app use which was adopted earlier by their higher SES peers. In other words, low SES users, in addition to apps, require access to the mobile web to do things that could once only be done on PCs. Because many of these pages were not optimized for mobile use, it appeared they relied less on their browsers as a function of experience. Recall that low SES users accessed more non-mobile sites which required more resources to load. The resulting page loading delays [87] have been noted as a primary cause of web usage declines on PCs [89]. Clearly, this is also a primary problem for the mobile web and this is especially problematic for low SES users. Note that since only a few top websites are most commonly used, it can be expected that predictive capabilities can be leveraged to preload their most common resources and improve performance.

7.3.3. **Smartphone Design**

Based on the results of SES comparisons, this dissertation identifies several key groups of users that phones must cater to. This dissertation acknowledges the observations were made from a very narrow demography of smartphone users (college students), and that a broader user population likely has many more and different groups. Nonetheless, the significant differences in such a narrow demographic strongly suggest that the one-size-fits-all phone design paradigm fails to serve the best interest of users. Instead, multiple mobile platforms with appropriately selected features are more likely to compliment the needs of different user groups. While some features can be achieved through customizations of software and/or the OS, others require hardware modifications, e.g. hardware keyboard, game controller buttons, and small form factor.

The low SES users had significantly higher overall usage, which places greater requirements on the device's battery. Their mobile web browsing was also shown to be more of an extension to their PC experience, increasing the value of larger screen sizes for them. Since both battery capacity and screen size come as a tradeoff to device compactness, it can be hypothesized that everything else equal, different users would significantly benefit from different choices in terms of these tradeoffs, e.g. higher capacity battery and larger displays for low SES users.

7.3.4. **Field Evaluations**

The findings in this chapter provide important insights into how the field evaluation of smartphone and its service should be designed and carried out.

First, the findings demonstrate the importance of controlling for demographic factors to understand differences between users. Prior work on smartphone usage was not particularly prudent in participant selection and, not surprisingly, failed to reveal any difference [59], or failed to provide conclusive evidence for speculated differences [60].

Second, the findings demonstrated that extraordinary care must be taken in drawing conclusions from data collected by giving out devices and studying them in field for a short period of time (e.g. shorter than three months). The results show that the first months see a significantly different degree of exploration and diversity in usage than in the remaining months (e.g. Figure 7.2, Figure 7.6, and Figure 7.8). Moreover, because usage continues to evolve even one year into the study, conclusions drawn using data collected from a short period of time should be generalized with care. Examples include the seasonal variation in usage, and apps losing appeal, as is often the case with games.

Third, the findings demonstrated the value of following the same users for a long period of time and of being able to interview them for insights into their behavior. This is shown both by the significant usage changes in the later months of the study. However, this method is expensive financially and administratively. Therefore, it can only be applied to relatively few participants. As a result, this method is complementary to those that gather data from a large number of users but only sporadically, e.g., [90].

Chapter 8

Last Thoughts

8.1. Conclusion

The focus of this dissertation has been the methodological quantification and measurement of the context dependency of mobile usage, in an application agnostic yet practical manner. The context dependency of three principal types of mobile usage have been measured; web, phone, and app usage. However, the methodology of this dissertation can be readily applied to other forms of mobile usage as well as system resources. This dissertation shows that there is considerable context dependency in mobile usage, but it needs to be carefully and efficiently extracted. This work is distinguished from prior context aware work, as they typically provide ad-hoc and application specific solutions. In contrast, the methodological approach of this dissertation liberates system designers from having to design and evaluate a new solution for every context-based system service or application.

This dissertation has identified estimation accuracy as the application agnostic yet practical measure of choice for context dependency, and has provided a methodological approach to calculate the posterior probabilities for MAP estimation. To this end, it has addressed the challenges of sparseness when dealing with multiple sources of context information, and the challenge of energy consumption for acquiring context from energy hungry sensors. In particular, by utilizing energy hungry context only at uncertain times, SmartContext can achieve an estimation accuracy within 1% of the maximum possible accuracy, while significantly reducing energy costs by half or more. The traces from LiveLab have allowed the measurements and resulting findings to be based on real-life in-situ usage of mobile devices. Furthermore, the LiveLab traces allow this dissertation to evaluate several sample context-aware applications. Most importantly, these applications highlight the practical value of estimation accuracy as a measure of context dependency, and attest to the effectiveness of context for estimating usage. As expected, the best performing context-aware methods consistently outperform common non-context-based methods.

Finally, this thesis investigates the role of social context in mobile usage, by collectively comparing two carefully selected socioeconomic groups among the participants. These findings highlight the significant role of social context, even when other factors are kept similar, and provide researchers and system designers with a better understanding of how users vary, and how to design mobile devices, services, applications, and content.

8.2. Future directions

The methodology and findings of this dissertation opens the path for numerous context-aware applications and research, and provide for a better understanding of a broader range of individuals. However, there are four specific directions for future work that are inspired by this dissertation.

First, is providing device independent system support for context awareness on mobile devices. This dissertation provides methodologies for practical context awareness. However, it is necessary for mobile devices to have system-wide support for context awareness, including APIs for context-aware applications and services to interact with. Such an implementation has significant benefits; it can unlock additional energy savings by combining sensor readings and training data from all context-aware applications and services, while at the same time relieving their designers from the burden of individually implementing data collection, training, and context-based estimation.

Second, this dissertation highlights that in many practical applications, MAP based estimation accuracy may be a better measure for useful information, compared to entropy. Therefore, operations research communities may want to extend work using MAP based estimation accuracy as a more complicated utility function compared to information entropy In particular, while this dissertation showed the overall submodularity property of the LiveLab data, it does not prove it for all conditions. A better treatment of MAP based estimation accuracy appears necessary. For example, identifying the mathematical requirements that can ensure the submodularity of MAP based estimation accuracy, can help mobile designers in their choice of context sources and their methods for handling them.

Third, the unexpected findings regarding social context highlights the need for more carefully designed and conducted research on the human factors of mobile computing, and on a wider range of demographics. LiveLab leads the way to enable and simplify such research in the future. Therefore, the author is hopeful that the hypothesis presented in this paper, as well as other effects of social context, will be more thoroughly evaluated in the future. In turn, this would assist mobile and application designers and researchers from all disciplines.

Fourth, this dissertation found that for accelerometer power readings, a relatively high number of bins (e.g. 100+) can improve estimation accuracy of usage. This was in contrast with the author's original expectations that a small number of bins would be sufficient to account for the users' physical state, e.g. walking, running, etc. Prior research on extracting users' physical or activity states using accelerometer readings also typically focus on a relatively small number of states, e.g. as in [37]. The fact that a hundred or more accelerometer power bins can improve estimation accuracy suggests that accelerometer power can and should be used as a *signature* to classify a user's detailed state, and not merely as an indicator for whether they are moving or not. Previous research has shown a similar phenomenon with ambient sound for the purpose of room level localization, i.e. SoundSense [80].

References

[1] Schilit, B., Adams, N. and Want, R., Context-aware computing applications. in *Proc. Workshop on Mobile Computing Systems and Applications,* 1994.

[2] Gu, T., Pung, H.K. and Zhang, D.Q., A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 2005.

[3] Lemlouma, T. and Layaida, N., Context-aware adaptation for mobile devices. in *Proc. Int. IEEE Conf. on Mobile Data Management*, 2004.

[4] Kane, S.K., Wobbrock, J.O. and Smith, I.E., Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. in *Proc. ACM Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*, 2008.

[5] Schmidt, A., Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 2000.

[6] Marmasse, N. and Schmandt, C., Location-Aware Information Delivery with ComMotion. in *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, 2000.

[7] Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C., Developing a context-aware electronic tourist guide: some issues and experiences. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2000.

[8] Banerjee, N., Rahmati, A., Corner, M.D., Rollins, S. and Zhong, L., Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. *Proc. Int Conf. Ubiquitous Computing (Ubicomp)*, 2007.

[9] Ravi, N., Scott, J., Han, L. and Iftode, L., Context-aware Battery Management for Mobile Phones. *Proc. Int. Conf. on Pervasive Computing and Communications (PerCom)*, 2008.

[10] Rahmati, A. and Zhong, L., Context-Based Network Estimation for Energy-Efficient Ubiquitous Wireless Connectivity. *IEEE Transactions on Mobile Computing*, 2011.

[11] van Kranenburg, H., Bargh, M.S., Iacob, S. and Peddemors, A., A context management framework for supporting context-aware distributed applications. *Communications Magazine, IEEE*, 2006.

[12] Lee, Y., Iyengar, S.S., Min, C., Ju, Y., Kang, S., Park, T., Lee, J., Rhee, Y. and Song, J., MobiCon: a mobile context-monitoring platform. *Commun. ACM*, 2012.

[13] Baldauf, M., Dustdar, S. and Rosenberg, F., A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2007.

[14] Hastie, T., Tibshirani, R. and Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[15] Zhuang, Z., Kim, K.-H. and Singh, J.P., Improving energy efficiency of location sensing on smartphones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[16] Paek, J., Kim, J. and Govindan, R., Energy-efficient rate-adaptive GPS-based positioning for smartphones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[17] Wang, Y., Krishnamachari, B., Zhao, Q. and Annavaram, M., Markov-optimal sensing policy for user state estimation in mobile devices. in *Proc. Int. Conf on Information Processing in Sensor Networks (IPSN)*, 2010.

[18] Panuccio, P., Ghasemzadeh, H., Fortino, G. and Jafari, R., Power-aware action recognition with optimal sensor selection: an AdaBoost driven distributed template matching approach. in *Proc. ACM Workshop on Mobile Systems, Applications, and Services for Healthcare*, 2011.

[19] Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., Tröster, G. and Verdone, R., Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection. *Wireless Sensor Networks*, 2008.

[20] Kang, S., Lee, J., Jang, H., Lee, H., Lee, Y., Park, S., Park, T. and Song, J., SeeMon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2008.

[21] Wang, Y., Lin, J., Annavaram, M., Jacobson, Q.A., Hong, J., Krishnamachari, B. and Sadeh, N., A framework of energy efficient mobile sensing for automatic user state recognition. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2009.

[22] Schilit, B.N. and Theimer, M.M., Disseminating active map information to mobile hosts. *Network, IEEE*, 1994.

[23] Schilit, B., Adams, N. and Want, R., Context-aware computing applications. in *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, 1994.

[24] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M. and Steggles, P., Towards a Better Understanding of Context and Context-Awareness. in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999.

[25] Schwinger, W., Grün, C., Pröll, B., Retschitzegger, W. and Schauerhuber, A., Context-awareness in Mobile Tourism Guides. *Technical Report, Institute for Applied Knowledge Processing, University of Linz, Austria*, 2005.

[26] Chen, G. and Kotz, D. A survey of context-aware mobile computing research, Citeseer, 2000.

[27] Asthana, A., Cravatts, M. and Krzyzanowski, P., An indoor wireless system for personalized shopping assistance. in *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, 1994.

[28] Schmidt, A., Aidoo, K., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W. and Gellersen, H.-W. Advanced Interaction in

Context. in *Handheld and Ubiquitous Computing*, Springer Berlin / Heidelberg, 1999, 89-101.

[29] Motorola Mobility. November 17, 2011 Press Release - Motorola Smart Actions: Uncover the secret super powers of your phone *http://www.motorola.com/blog/2011/11/17/motorola-smart-actions-uncover-the-secret-super-powers-of-your-phone/*, 2011.

[30] Lee, J.-K. and Hou, J.C., Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application. in *Proc. ACM int. symp. on mobile ad hoc networking and computing (MobiHoc)*, 2006.

[31] Kotz, D., Henderson, T., Abyzov, I. and Yeo, J. CRAWDAD data set dartmouth/campus (v. 2009-09-09) *http://crawdad.cs.dartmouth.edu/dartmouth/campus*, 2009.

[32] Nicholson, A.J. and Noble, B.D., BreadCrumbs: Forecasting Mobile Connectivity. *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*, 2008.

[33] Rahmati, A. and Zhong, L., Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2007.

[34] Cugola, G., Migliavacca, M., Roedig, U. and Sreenan, C., A Context and Content-Based Routing Protocol for Mobile Sensor Networks. *Wireless Sensor Networks*, 2009.

[35] Rahmati, A., Qian, A. and Zhong, L., Understanding human-battery interaction on mobile phones. in *Proc. ACM Int. Conf. Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2007.

[36] Rahmati, A. and Zhong, L., Human-battery interaction on mobile phones. *Pervasive and Mobile Computing*, 2009.

[37] Bouten, C.V.C., Koekkoek, K.T.M., Verduin, M., Kodde, R. and Janssen, J.D., A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily Physical Activity. *IEEE Trans. Biomedical Engineering*, 1997.

[38] LiKamWa, R., Liu, Y., Lane, N.D. and Zhong, L., Can Your Smartphone Infer Your Mood? *Proc. int. wrkshp. on Sensing for App Phones (PhoneSense)*, 2011.

[39] Laasonen, K., Jorge, A., Torgo, L., Brazdil, P., Camacho, R. and Gama, J.o. Clustering and Prediction of Mobile User Routes from Cellular Data Knowledge Discovery in Databases: PKDD 2005. in, Springer Berlin / Heidelberg, 2005, 569-576.

[40] Rathnayake, U., Ott, M. and Seneviratne, A., A DBN approach for network availability prediction. in *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, 2009.

[41] Libo, S., Kotz, D., Ravi, J. and Xiaoning, H., Evaluating Next-Cell Predictors with Extensive Wi-Fi Mobility Data. *Mobile Computing, IEEE Transactions on*, 2006.

[42] Kijima, M. *Markov processes for stochastic modeling*. Chapman & Hall/CRC, 1997.

[43] Peddemors, A., Eertink, H. and Niemegeers, I., Predicting mobility events on personal devices. *Pervasive and Mobile Computing*, 2010.

[44] Aggarwal, C.C., Han, J., Wang, J. and Yu, P.S., A framework for clustering evolving data streams. in *Proceedings of the 29th international conference on Very large data bases - Volume 29*, 2003.

[45] Raento, M., Oulasvirta, A., Petit, R. and Toivonen, H., ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing*, 2005.

[46] Wood, A., Stankovic, J., Virone, G., Selavo, L., Zhimin, H., Qiuhua, C., Thao, D., Yafeng, W., Lei, F. and Stoleru, R., Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 2008.

[47] Krause, A. and Guestrin, C., Near-optimal nonmyopic value of information in graphical models. in *Proc. of Uncertainty in AI*, 2005.

[48] Van Der Gaag, L. and Wessels, M., Selective evidence gathering for diagnostic belief networks. *AISB Quarterly*, 1993.

[49] Dittmer, S.L. and Jensen, F.V., Myopic value of information in influence diagrams. in *Proc. conf. Uncertainty in Artificial Intelligence (UAI)*, 1997.

[50] Bayer-Zubek, V., Learning diagnostic policies from examples by systematic search. in *Proc. conf. on Uncertainty in Artificial Intelligence (UAI)*, 2004.

[51] Ko, C.-W., Jon, L. and Queyranne, M., An Exact Algorithm for Maximum Entropy Sampling. *Operations Research*, 1995.

[52] Goldengorin, B., Tijssen, G.A., Tso, M. and A. The maximization of submodular functions : old and new proofs for the correctness of the dichotomy algorithm, University of Groningen, Research Institute SOM (Systems, Organisations and Management), 1999.

[53] Krause, A. and Guestrin, C., Optimal nonmyopic value of information in graphical models-efficient algorithms and theoretical limits. *Proc. IJCAI*, 2005.

[54] Nemhauser, G.L., Wolsey, L.A. and Fisher, M.L., An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 1978.

[55] Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B. and Landay, J.A., MyExperience: a system for in situtracing and capturing of user feedback on mobile phones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2007.

[56] Intille, S.S., Rondoni, J., Kukla, C., Ancona, I. and Bao, L., A context-aware experience sampling tool. in *CHI '03 extended abstracts on Human factors in computing systems*, 2003.

[57] Eagle, N. and Pentland, A., Reality mining: sensing complex social systems. *Personal Ubiquitous Computing*, 2006.

[58] Eagle, N. and Pentland, A., Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 2009.

[59] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R. and Estrin, D., Diversity in Smartphone Usage. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[60] Rahmati, A. and Zhong, L., A Longitudinal Study of Non-Voice Mobile Phone Usage by Teens from an Underserved Urban Community. *Rice University Technichal Report RECG-0515-09*, 2009.

[61] Rahmati, A. and Zhong, L., Usability evaluation of a commercial Pocket PC phone: A pilot study. *Proc. ACM Int. Conf. Mobile Technology, Applications and Systems (Mobility)*, 2007.

[62] Eckhart, T., AndroidSecurityTest.com: CarrierIQ. *http://androidsecuritytest.com/features/logs-and-services/loggers/carrieriq/*, November 12, 2011.

[63] Kravets, D., Wired.com: Researcher's Video Shows Secret Software on Millions of Phones Logging Everything. *http://www.wired.com/threatlevel/2011/11/secret-software-logging-video*, November 29, 2011.

[64] Shneiderman, B., Universal usability. *Commun. ACM*, 2000.

[65] Goel, S., Hofman, J.M. and Sirer, M.I., Who Does What on the Web: Studying Web Browsing Behavior at Scale. *under review*, 2011.

[66] Tossell, C., Jardina, J.R., Kortum, P.T., Peres, S.C., Shepard, C.W., Rahmati, A. and Zhong, L., Effects of Socioeconomic diversity on iPod Touch Device Use in Real-World Environments. *Proc. Human Factors and Ergonomics Society (HFES)*, 2011.

[67] rsync. an open source utility that provides fast incremental file transfer *http://samba.anu.edu.au/rsync/*.

[68] PostgreSQL. PostgreSQL object-relational database management system (ORDBMS) *http://www.postgresql.org/*.

[69] Hoon, M.J.L.d., Imoto, S., Nolan, J. and Miyano, S., Open Source Clustering Software. 2004.

[70] Cluster 3.0. Open Source Clustering Software *http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm*.

[71] Scott Long, J. and Freese, J. *Regression models for categorical dependent variables using stata*. Stata Press, 2006.

[72] Kuncheva, L. *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.

[73] Provost, F. and Domingos, P., Tree Induction for Probability-Based Ranking. *Machine Learning*, 2003.

[74] Zadrozny, B. and Elkan, C., Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *Proc. Int. Conf. on Machine Learning (ICML)*, 2001.

[75] Domingos, P. and Pazzani, M., Beyond independence: Conditions for the optimality of the simple Bayesian classifier. in *Proc. Int'l Conf. Machine Learning*, 1996.

[76] Zhang, H., The optimality of naive Bayes. *Proc. Int Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2004.

[77] Duin, R. and Tax, D. Experiments with Classifier Combining Rules. in *Multiple Classifier Systems*, Springer Berlin / Heidelberg, 2000, 16-29.

[78] Kittler, J., Hatef, M., Duin, R.P.W. and Matas, J., On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[79] Hill, T. and Lewicki, P. *Statistics: Methods and Applications*. StatSoft, Inc., 2005.

[80] Lu, H., Pan, W., Lane, N.D., Choudhury, T. and Campbell, A.T., SoundSense: scalable sound sensing for people-centric applications on mobile phones. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2009.

[81] Shepard, C., Rahmati, A., Tossell, C., Zhong, L. and Kortum, P., LiveLab: measuring wireless networks and smartphone users in the field. *SIGMETRICS Perform. Eval. Rev.*, 2010.

[82] Yan, T., Chu, D., Ganesan, D., Kansal, A. and Liu, J., Fast App Launching for Mobile Devices using Predictive User Context. *Under Review*, 2012.

[83] Dong, M. and Zhong, L., Self-constructive high-rate system energy modeling for battery-powered mobile systems. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2011.

[84] Monsoon Solutions Inc. Monsoon FTA22D Power Monitor http://www.msoon.com/LabEquipment/PowerMonitor/.

[85] Guestrin, C., Krause, A. and Singh, A.P., Near-optimal sensor placements in gaussian processes. in *Proceedings of the 22nd international conference on Machine learning*, 2005.

[86] Chow, C. and Liu, C., Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 1968.

[87] Wang, Z., Lin, X., Zhong, L. and Chishtie, M., Why are Web Browsers Slow on Smartphones? in *Proc. ACM Int. Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2011.

[88] Menard, S., Six approaches to calculating standardized logistic regression coefficients. *The American Statistician*, 2004.

[89] Sears, A. and Jacko, J.A. *The human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications*. Lawrence Erlbaum, 2003.

[90] Huang, J., Xu, Q., Tiwana, B., Mao, Z., Zhang, M. and Bahl, P., Anatomizing Application Performance Differences on Smartphones. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.