Community-guided Mobile Phone Sensing Systems

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Nicholas D. Lane

DARTMOUTH COLLEGE

Hanover, New Hampshire

June, 2011

Examining Committee:

(co-chair) Andrew T. Campbell, Ph.D.

(co-chair) Tanzeem Choudhury, Ph.D.

Deepak Ganesan, Ph.D.

Reza Olfati-Saber, Ph.D.

Brian W. Pogue, Ph.D.

Dean of Graduate Studies

Abstract

Smartphones with embedded sensors have become commonplace items carried by millions of people. Mobile phone sensing is now on the cusp of going mainstream. Two critical ingredients, the access to large-scale sensor data and robust mobile classification, underpin the majority of the emerging mobile sensing applications. Early research prototypes and small-scale deployments suggest these applications will revolutionize many aspects of our society, ranging from healthcare to energy-awareness.

However, the potential for mobile phone sensing to transform our world is threatened. Conventional approaches to training and performing inference using classification models are ill-suited to the conditions presented by this new domain. Making robust inferences regarding complex human activities and events is challenging due to the extreme diversity in both the contextual conditions and user characteristics encountered in the real-world. Existing methods for model training, requiring controlled experiments to collect labeled data, are unable to scale to large end-user populations.

We believe that solutions to many unsolved mobile sensing problems can be found in an approach we refer to as *community-guided mobile phone sensing systems*. Community-guided systems leverage not only individuals but are effective in exploiting the collective power of communities. These systems understand that people are part of hierarchies of densely connected communities, effected by group behavior and influenced by social networks. Through this understanding the strengths of communities and machines can be combined within a single sensing system.

Specifically, this thesis proves that the tight coupling of communities with the sensing systems they use can overcome many obstacles surrounding mobile classification. We investigate a variety of commonly occurring communities (e.g., opportunistic and social networks) and develop the algorithms and architectures required to extract the potential they contain. Two key thesis contributions are our proposals of Community-guided Learning (CGL) and Community Similarity Networks (CSN). CGL enables error-ridden, yet plentiful, labeled data produced from crowd-sourcing communities to train models of human behavior. CSN exploits the less obvious, but still ever-present, networks of similar people (e.g., sharing behavior or physical characteristics) to adapt generic classification models to specialize in subgroups within the broader user population. We finally present BeWell, a mobile health application which empowers individuals to manage their own wellbeing. This application senses multiple behavioral patterns which collectively influence the overall health of the user. BeWell is a case study into an application that requires the advances in mobile classification made by the community-guided techniques this thesis has pioneered.

Acknowledgments

First, I thank my co-advisors Andrew T. Campbell and Tanzeem Choudhury. More than anyone else they have influenced my view of the research process, and instilled in me the importance of aiming to produce high-quality research with the potential for impact. Most of all, I value their candid and honest opinions, their calmness and clarity of advice amidst difficult times, and their patience and understanding over the past several years. I am indebted to have had advisors that gave me all of the freedom, resources, guidance and support I could ever need during the period that lead up to this dissertation. They provide an environment where researchers can thrive and where the degree to which you succeed is ultimately in your own hands.

I feel fortunate to have had the opportunity to work closely with Feng Zhao, Dimitrios Lymberopoulos and David Chu during a five month internship that included time at MSRA in Beijing, China and MSR in Redmond, Washington. I throughly enjoyed the chance to work with not only Feng, Dimitrios and David but the many other exceptional researchers and interns at both locations.

As a member of the Mobile Sensing Group and the People-Aware Computing group I count myself lucky to have been surrounded by a number of outstanding individuals who, at different stages of my PhD, have been part of either lab (Emiliano Miluzzo, Shane Eisenman, Hong Lu, Ron Peterson, Gahng-Seop Ahn, Mirco Musolesi, Kristof Fodor, Mashfiqui Rabbi, Mu Lin, Andy Sarroff, Ye Xu, Xiaochao Yang, Shaohan Hu, Michela Papandrea, Cory Cornellius, Dan Peebles, Xiao Zheng). In particular, I must make special mention of Emiliano, Shane and Hong. Over the years we have shared many long hours working together in the lab, and in the case of Emiliano since the very first day of my degree. They are not only good colleagues, but good friends.

I am definitely lucky to have had support from some amazing people over these years. I would especially like to thank Malcolm Bailey, Kylie Akin, Salmaan Rashid, Sekar Velu, Jesse McMullen, Calhoun Wiseman, Deane Searle, Rachel Gonzalez, and Ushan Feng for being there. I apologize to all my family and friends for the past years. I appreciate your understanding of the unreasonably long delays in my replies to phone calls and emails. I thank you all for not giving up on me and I plan on keeping in closer contact in the future. Thank you all for your love and support.

Finally, I can not thank enough my mother for being accepting and loving irrespective of the unpredictable nature of my work schedule and the uncertainty it brings. Living half way across the world complicates many things for a family and she has always stood by me.

Dedicated to my father, Donald W. Lane

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Today's smartphone not only serves as the key computing and communication mobile device of choice but it also comes with a rich set of embedded sensors, for example, an accelerometer, digital compass, gyroscope, GPS, microphone and camera. Collectively, these sensors are enabling a new class of applications to emerge across a wide variety of domains, for instance, healthcare [28], social networks [73], safety, environmental impact [75] and transportation [6, 102], establishing a new area of research called mobile phone sensing.

Until recently mobile sensing research for example activity recognition where people's activity (e.g., walking, driving, sitting, talking) is classified and monitored required specialized mobile devices (e.g., the Mobile Sensing Platform (MSP) [25]) to be fabricated [97]. Mobile sensing applications had to be manually downloaded, installed and hand tuned for each device. User studies conducted to evaluate new mobile sensing applications and algorithms were small scale because of the expense and complexity of doing experiments at scale. As a result the research, which was innovative, gained little momentum outside a small group of dedicated researchers. Although the potential of using the mobile phones as a platform for sensing research has been discussed for a number of years now at industry led meetings [78] and in the literature [33, 94, 96] there has been little or no advances in the field until recently.

All that changed because of a number of important technological advances. First, the availability of cheap embedded sensors initially included in phones to drive the user experience (e.g., the accelerometer used to change the display orientation) is changing the landscape of possible applications. Now phones can be programmed to support new disruptive sensing applications including sharing the user's real-time

activity with friends on social networks like Facebook, keeping track of a person's carbon footprint, or monitoring their well-being. Second, smartphones are open and programmable. In addition to sensing, phones come with computing and communication resources that offer a low barrier of entry for third party programmers (e.g., undergraduates with little phone programming experience are developing and shipping applications). Third, and importantly, each phone vendor now offers an "app store" allowing developers to deliver new applications to large populations of users across the globe, which is transforming deployment of new applications and allowing the collection and analysis of data far beyond the scale of what was previously possible. Fourth, the mobile computing cloud enables developers to offload mobile services to backend servers providing unprecedented scale and additional resources for computing on collections of large-scale sensor data and supporting advanced features like persuasive user feedback based on the analysis of big sensor data.

The combination of these advances in embedded phone sensors, open programability, large-scale application delivery channels and mobile cloud support opens the door for new innovative research and development of sensing applications that are likely to revolutionize a large number of existing business sectors and ultimately significantly impact our everyday lives. Many questions remain to make this vision a reality. For example, how much intelligence can we push to the phone without jeopardizing the phone experience? How do we scale a sensing application from an individual, to a target community or even the general population? How do we use these new forms of large-scale application delivery systems (e.g., Apple AppStore, Google Market) to best drive data collection, analysis and validation? How can we exploit the availability of big data shared by applications but build water-tight systems that protect personal privacy? While this new research field can leverage results and insights from wireless sensor networks, pervasive computing, machine learning, and data mining, it presents new challenges not addressed by these communities.

This thesis makes contributions to many of these still open problems. However, we examine in greatest depth the various difficulties that arise when interpreting sensor data sampled by mobile phones. This process is at the heart of many emerging mobile sensing applications. In our study we take an approach grounded in the leveraging of user communities. We find that many of the challenges presented by mobile phone sensing to classification model training and inference can be overcome by closely integrating communities within these systems.

1.1.1 Mobile Classification

At this stage in the development of mobile phone sensing systems a number of the most critical technical challenges revolve around the interpretation of noisy sensor data. Although the applications of this emerging technology are broad the majority of them all require the classification of low-level raw data into the high-level complex human behaviors, events and contexts commonly found in the real world. In this thesis we refer to the conditions and problems of performing classification using mobile phones as *mobile classification*. We believe that mainstream usage of these systems will not occur until smartphones can understand sensor data using statistical models that have two key qualities, robustness and scalability.

Smartphones are exposed to an unpredictable range of environments (e.g., loud outdoor streets, quiet indoor offices) and used in unexpected ways (e.g., stored in bags and briefcases, placed in pockets and on belts). Accompanying the diverse range of contexts to which a phone is exposed are an equally diverse assortment of users. Users' gender, age, physical status and even lifestyles vary greatly from person to person. The existing statistical models used by mobile phones are supervised (example-based) and fail to generalize to the extreme diversity in context and users that are common in smartphone sensing. Further, the mobile device constraints of memory, computation, bandwidth, sensor-availability and finally the need for the phone to continue to operate successfully as a phone all contribute to complicating the development of effective solutions.

The difficulties of mobile classification only exacerbate a long existing bottleneck common in more general large-scale uses of classification – the effort and cost of acquiring labeled training data. The inability of supervised learning to cope with diversity could be addressed if equivalently diverse training data were readily available to provide examples of various contexts and users. However, conventional means of acquiring training data rely heavily on carefully controlled experiments. These experiments are costly to stage (and so are infrequent) and consider only a small number of different contexts or personal characteristics at any one time. This makes solutions based on solely on sourcing diverse training data impractical due to the shear number of different combinations of classes, contexts and types of people. Existing alternative approaches, for example, semi-supervised and unsupervised techniques, are still not yet well suited to most of the the classification tasks required by mobile sensing applications. Until breakthroughs occur in this area of machine learning then the availability of training data will remain pivotal to mobile classification.

1.1.2 Community-guided Mobile Phone Sensing Systems

This thesis advocates the deeper integration and tighter coupling of communities of everyday people and the sensing systems that serve them. We believe that this approach can be effective in overcoming many of the obstacles to smartphone sensing. The chapters of this dissertation prove the power of communities to overcome the challenges of mobile classification but we conjecture that further study will discover many other uses. We define a *community-guided mobile phone sensing system* as a system that either has one, or a combination, of the following two characteristics: i) the *direct* exploitation of a community of people (e.g., adopting a human-in-the-loop approach) where a group of users assume an active role; ii) the *indirect* leveraging of a community through the understanding of community behavior or structure (e.g., a social network or networks of similarity between communities); whereby the system benefits by being aware of communities but does not require conscious effort by users.

Today, the design and operation of mobile phone sensing systems correctly considers individual users very strongly. However, too often these design considerations have an overly narrow perspective. They imply people live in a vacuum and ignore the fact they are routinely part of a number of different communities. Instead, users of sensing systems are part of hierarchies of densely connected community groups and are effected by group behavior and influences from social networks. The combination of individuals, groups and the complex dynamics between them presents both many different types of communities and a variety of opportunities for these communities to be used. This thesis offers different perspectives on both of these issues. For instance, we consider ephemeral communities of co-located mobile devices, communities based on the social networks of users and crowd-sourcing communities which have been so effective in other domains. Our work also considers less obvious communities comprised of people who share common traits. We show such communities can assist in mobile classification when combined with crowd-sourcing communities, or be used by themselves to improve persuasive user feedback.

When considered collectively this dissertation poses an important question, are there a general set of principles or a re-useable design for binding communities and mobile sensing systems irrespective of the type of community or the mobile sensing problem being addressed? This thesis employes a variety of different techniques to overcome a range of what are largely mobile classification problems. However, we have focused primarily on a single category of problem and even with this constraint a unified model is not clear. One limiting factor to progress is the complexities of communities themselves, large-scale human networks and the interactions that occur

within them have been studied extensively for decades. As such the narrow waist in the design of these systems will take time to emerge, as researchers gain a more comprehensive understanding of this new interdisciplinary problem space. Our work represents a decisive step towards achieving this understanding.

1.2 Problem Statement

Having introduced the emerging field of mobile phone sensing, and discussed the challenges and opportunities it presents we now describe four specific problems addressed by this dissertation.

First, we study two key challenges to mobile classification, these being: i) heterogeneity in mobile devices and ii) the difficulty of acquiring large-scale training data. The problem of heterogeneity in mobile phones occurs as they commonly come in many different configurations, with sensing today still a secondary operation. This causes uncertainty as to the availability of primarily sensors and more generally computation, storage and network connectivity. We find this restricts the classification pipeline to a less than ideal design (i.e., the features and models employed), resulting in classification accuracy that is not as robust as would be otherwise possible. A lack of training data is a commonly encountered problem in many applications of machine learning. Under mobile classification training data is a critical resource required to build classifiers able to cope with diverse phone operating conditions and user characteristics. As a solution we envision a scenario where communities of users share labeled sensor data that they collected themselves. However, as a result secondary problems manifest, namely user burden and user disagreement, which we also investigate. User burden arises due to the additional effort necessary for individuals to manually provide labeled data. User disagreement is caused by inconsistencies between users during the labeling process, for example, due to semantic disagreement as to the textual description or the precise definition of an activity or event that is labeled. Absent from the state-of-the-art in mobile classification are processes which can allow cooperation to occur either between users or between co-located devices without noticeable drops in classification accuracy.

Second, we address a principle problem in enlisting members of a community to improve classification performance. The most direct way users can assist in more accurate classification is for them to provide training data themselves. Problems in doing this were highlighted by our earlier investigation into the sharing of labeled data within social networks of users. The process of collecting large-scale sensor data and

labeled activities from user using smartphones is itself technically simple. However, in practice learning classification models using this data is not nearly as easy due to the deficiencies in training data labeled using this approach. This is not surprising given even trained researchers make mistakes when labeling data. Such errors become much more pronounced, however, when labels are gathered by crowd-sourcing from inexperienced "low-commitment" users. In particular, users may give identical labels to activities with characteristically different signatures (e.g., labeling eating at home or at a restaurant as "dinner") or may give different labels to the same context (e.g., "work" vs. "office"). Further, they commonly make mistakes in the segmentation of classes (i.e., an activity or context), for example, by forgetting to indicate to the system an activity has stopped or changed. Under this scenario labels are unreliable but nonetheless contain valuable information for classification. The problem, however, is that existing machine learning techniques are unable to train models safely in the presence of error-prone labels as dirty as those sourced from crowds.

Third, we investigate a significant threat to scaling mobile phone sensing systems through our study of what we refer to as the *population diversity problem*. As the size of a user population increases not surprisingly so does the amount of diversity the population contains. Users can vary for a number of reasons, a clear example being physical differences, for instance, height, weight, sex, or the extent to which they are physically active. Other categories of dissimilarity include those based on lifestyle and background. People can live and work in different places, have different cultures and socio-economic origins, although they may even do the same basic set of activities (e.g., workout, employment, socialize) they can do these activities in much different ways. As we previously mentioned existing statistical models used in mobile sensing to recognize activities and events are supervised, example-driven. The robustness of inference using these models quickly degrades in the presence of such diversity. We demonstrate in this thesis that even with as few as 50 people classification accuracy can vary significantly from person to person. These findings motivate the need for approaches to mobile classification that recognize the individual differences between users and adapt classification models to compensate for such variation.

The final problem we address is how to design a persuasive mobile phone sensing system capable of helping individuals manage their own overall health and wellbeing. There exists numerous examples of persuasive systems [36] that have been successful in managing narrow aspects of health (e.g., stress [35], diet [82]), with the majority of work done on physical activity [28, 64]. However, the requirements of comprehensive wellbeing monitoring and persuasion are much more demanding. The system must

be able to not only continuously monitor multiple dimensions of behavior but also understand the implications of each behavior to the overall wellbeing of the user. This information must then be distilled into a stream of persuasive nudges that simultaneously balance the various behavioral goals required to achieve high levels of wellbeing. Effective persuasion under this scenario is particularly difficult. Considering multiple behaviors widens the scope of information that is useful to provide the user. However, the ability and inclination of people to absorb this information will likely remain the same. Persuasive signals must be developed that provide informative feedback to users on multiple dimensions of behavior yet are still not overly burdensome. The persuasive systems that have been developed so far fall well short in achieving this combination of demanding requirements.

1.3 Thesis Outline

In our exploration of the the open challenges just described we have taken an experimental approach. Where possible we have built prototype systems, conducted user studies, performed experiments and employed large-scale mobile sensing datasets to validate our ideas. In what follows we provide an outline of our study.

1.3.1 Cooperative Communities: Leveraging Social and Opportunistic Networks

In Chapter 3 we begin our exploration into the power of communities by considering two distinct, yet commonly occurring, communities within mobile phone sensing systems. These being: i) communities of co-located mobile devices that opportunistically form during everyday user activities (e.g., during a meeting, in a crowd, or at a coffee shop) and ii) communities based on the social networks that connect the users of mobile phones. We find that by leveraging these communities we are able to address two key challenges in performing classification in mobile phone environments, these being: i) heterogeneity in mobile device sensing and computational capabilities and ii) the time and effort necessary to acquire training data.

Leveraging these two very different communities presents specific sets of challenges unique to each community. Overcoming these challenges motivates the design of two complementary techniques, in-situ opportunistic feature vector merging and off-line social-network-driven sharing of training data and models between users. By sharing models and training data within groups of users defined by social connections we can

reduce the user effort and time involved in collecting training data, without reducing classification accuracy. The merging of features between nearby mobile phones can increase accuracy by enabling better performing models to be used even on devices without the requisite capabilities (e.g., sensors). We evaluate our proposed techniques with a significant places classifier [15] that infers and tags locations of importance to a user based on data gathered from sensor-enabled mobile phones. We report results from an experiment that ran for 12 days, involving 13 people.

1.3.2 Community-Guided Learning: Exploiting Crowd-sourced Labels

Chapter 4 continues our study by shifting attention to another common type of user community, large-scale groups of "low-commitment" contributors who participate in crowd-sourcing initiatives (e.g., Wikipedia). We examine how crowd-sourcing can be used to acquire training data. Training data is conventionally acquired through carefully controlled experiments that occur infrequently and involve a small number of people. By crowd-sourcing we can leverage potentially millions of people who continuously contribute a stream of labeled data. In this chapter we address one of the same fundamental bottlenecks to robust mobile classification that we considered in Chapter 3, the costly acquisition of labeled training data. However, we investigate an alternative but still complementary approach to the social-network-driven data and model sharing technique that we proposed in that earlier chapter.

Conventional learning algorithms assume labels provide perfect ground-truth however when crowd-sourcing the resulting labels are often noisy and prone to error, breaking this assumption. A key challenge in being able to exploit the crowd is overcoming this new difficulty to learning. To address this challenge we propose *Community-Guided Learning (CGL)*, a framework that allows existing classifiers to learn robustly from unreliably-labeled user-submitted data. Under this framework crowd-sourced data is intelligently re-grouped into classes by being guided by both the the underlying structure in the data and unconstrained free-form labels from users. CGL proposes the use of similarity measures to determine when and how to split and merge contributions from different labeled categories. Through experimental results we show this approach allows true classes to be identified from this data, overcoming errors and inconsistencies from users.

1.3.3 Community Similarity Networks: Scaling to Diverse Large-scale User Populations

The binding between community and sensing system in Chapter 4 required human attention and effort as we employed a "human-in-the-loop" approach to crowd-source labeled training data. In contrast Chapter 5 demonstrates communities can be exploited to solve critical problems in mobile sensing without directly involving the user. Specifically, we exploit the naturally occurring networks of similarity between communities of users; addressing the challenge of population diversity to performing mobile classification in large-scale sensing systems.

To overcome the population diversity problem we propose *Community Similarity Networks (CSN)*. CSN is an architecture and algorithms for mobile sensor-data classification that relies on a stream of crowd-sourced labels (made possible using CGL) and a learning process guided by various types of similarity networks (e.g., physical similarity, lifestyle similarity) that exist between users. Under CSN a new approach to training and inference for mobile classification is adopted, one in which generic classification models are specialized to be most accurate when used by a specific community of users. A mobile cloud infrastructure is employed to periodically retrain each specialized community classification model as new crowd-sourced labels accumulate, these updated models are downloaded by mobile phones to replace the previous version. We demonstrate, by building a prototype system and using mobile sensing datasets, that CSN not only improves classification robustness but is efficient and practical for use with existing sensor-enabled phones within large-scale deployments.

1.3.4 BeWell: Monitoring, Modeling, and Promoting Overall Wellbeing

The previous three chapters investigated the ability for communities to increase the scalability and robustness of mobile classification. We perform in Chapter 6 a case study of a mobile health application that requires such advances in classification for it to be effective. We describe our experiences in designing and deploying BeWell, a mobile phone based persuasive system [36] that enables people to not only closely monitor but improve their overall health and wellbeing.

BeWell continuously and automatically monitors user behavior in real-time along multiple dimensions, namely sleep, physical activity, and social interaction. The combination of these simple everyday behaviors (i.e., how we sleep, socialize and

exercise) is known to be closely connected to a wide range of individual health outcomes [37, 41, 79, 111], and can collectively shape the overall wellbeing of a person. BeWell is able to quantify the impact of user lifestyle choices on personal health by modeling the relationship between wellbeing and their own behavioral patterns. Users are made aware of these sensor-data based wellbeing assessments through an ambient display rendered on mobile phone wallpaper. We validate the design and effectiveness of BeWell with an implementation that uses commercial, off-the-shelf smartphones. Our detailed evaluation includes system benchmarks and controlled experiments, along with a 19 day, 27 person field trial. Results show that BeWell users are capable of digesting multidimensional wellbeing assessments and are responsive to feedback designed to promote improved health levels.

1.4 Thesis Contributions

The contributions of this thesis can be summarized as follows.

1. As the domain of mobile phone sensing matures it is becoming increasingly obvious that a key technical challenge for the field will be the enabling of robust and scalable mobile classification. In this thesis we are the first to clearly identify and then successfully leverage communities, of various forms, in meeting this challenge. As we describe in Chapter 3, we built the earliest mobile phone sensing system that embraced a "people-in-the-loop" approach within a sensor-data classification pipeline. In this system everyday users provided training data to boost the performance of a generic classification model. Our work into Cooperative Communities is the first to demonstrate the power of two distinct forms of frequently occurring communities: i) physically co-located communities of mobile devices that cooperate opportunistically and ii) communities formed by social networks of mobile device users who can cooperate off-line. During our study of these communities we proposed early, albeit now rudimentary, techniques which address a range of barriers to either mobile classification or community-guided sensing systems. These specifically were: i) population diversity and noisy crowd-sourced labels which we overcame by exploiting social networks, ii) the time and effort of acquiring training data, critical in specializing models for different mobile environments and communities of users, that we addressed by users providing labels themselves; and iii) heterogeneous mobile device capabilities either due to hardware or software limitations or even due to

phone placement on the user, which we coped with by sharing features between devices.

2. Crowd-sourcing has been a popular and successful approach in exploiting people to solve problems in a variety of domains (e.g., [105]). Chapter 4 presents Community-Guided Learning (CGL) which makes, for the first time, the crowd-sourcing of labels a viable approach to the problems of mobile classification. The CGL framework allows any supervised classifier to be trained using noisy labels contributed by untrained everyday people. With this framework we are advocating an entirely new paradigm for building models of human behavior which relies on the strengths of mobile phone sensing (i.e., the ability to tightly couple learning with the crowd) to overcome its challenges (e.g., the classification of complex human activities or noisy labels). Our experiments provided early insights as to the technical difficulties of crowd-sourcing labels, namely: inconsistent, subjective and error-prone labeling and segmentation by untrained low-commitment users. The result of our investigation was the development of an unconventional approach to learning that treats labels only as soft suggestions as to the actual class groupings within training data. Under CGL we use data similarity to compensate for the unavoidable unreliabilities of the labels provided.

3. Findings from smartphone sensing deployments [75] and activity recognition experiments [17] have hinted that existing classification models may not generalize to span common differences between people. We are the first to establish the importance of this problem, referring to it as the population diversity problem. In Chapter 5 we characterize the population diversity problem, showing it exists in even very different types of mobile classification and in user populations as small as 50 people. Community Similarity Networks (CSN) is currently the only mobile classification system able to cope with population diversity. Unless the techniques of CSN are used mobile phone sensing systems can not be deployed within a large-scale diverse user population without classification accuracy varying wildly from user to user. A major contribution of CSN is that it makes the personalization of classification models scalable to the extent models can be trained for unique communities of people, or even single individuals if so required.

4. The vast majority of mobile health systems target only a narrow slice of the wellbeing of an individual (e.g., [28, 35, 47, 64, 82]). As part of our Chapter 6

case study into an application that demands the advances in mobile classification that this thesis has delivered, we investigate how a mobile phone can monitor and promote overall wellbeing in users. BeWell is a new direction in persuasive mobile health. By focusing on holistic wellbeing it proposes solutions to new problems presented by what will become a common variety of mobile health application – comprehensive wellbeing management. Specifically, BeWell proposes approaches to: i) monitoring and promoting changes across multiple dimensions of everyday behavior and ii) increasing awareness to users of the wellbeing implications of different lifestyle choices. BeWell recognizes the need to interpret on behalf of the user the consequence to personal health and wellbeing caused by their own behavioral patterns. We investigate the use of an ambient display, rendered on smartphone wallpaper, that visualizes the wellbeing effects caused by multiple behavioral dimensions. Our experiments establish that not only are popular consumer smartphones a viable platform for *personal wellbeing applications*, but users are also able to understand and benefit from the multidimensional wellbeing feedback these application can provide.

The combination of Collaborative Communities, CGL, CSN and BeWell has expanded the boundaries of mobile phone sensing. We believe our findings open new avenues of research and will guide the design of future smartphone-based sensing systems.

Chapter 2

Background

2.1 Introduction

This chapter presents a snapshot of the progress made thus far by mobile phone sensing systems research. We begin by giving an overview of the sensors on the phone and their potential uses. We then discuss a number of leading application areas and sensing paradigms that have emerged in the literature recently. As a means to survey existing work and discuss important open challenges we formulate a simple architectural framework that serves as logical partitioning of what we consider to be important on the phone and in the cloud.

Our work focuses, principally, on how mobile classification can be improved by exploiting communities of smartphone users. The survey of mobile phone sensing found in this chapter provides the background information necessary to understand both the importance of mobile classification and the difficulties it faces. In the later chapters of this thesis we present BeWell, a smartphone system that relies on being able to interpret sensor data to monitor and promote a broad range of positive health outcomes. In this chapter we highlight additional application domains, beyond the mobile health domain considered by BeWell, which also rely on robust statistical models of sensor data and so further motivate the contributions made by this thesis.

2.2 Sensors

As mobile phones have matured as a computing platform and acquired richer functionality these advancements often have been paired with the introduction of new sensors. For example, accelerometers have become common after being initially in-

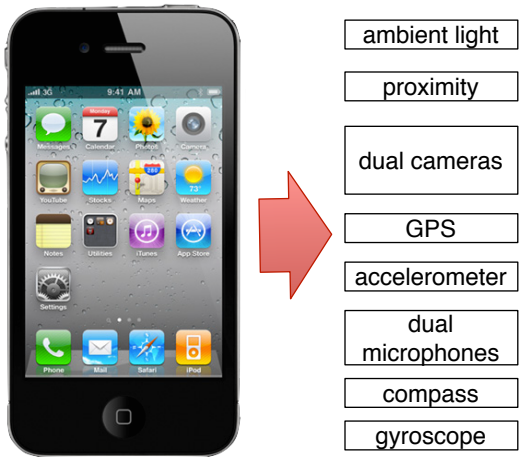


Figure 2.1: An off-the-shelf iPhone 4, representative of the growing class of sensor-enabled phones. This phone includes eight different sensors: accelerometer, GPS, ambient light, dual microphones, proximity sensor, dual cameras, compass and gyroscope.

troduced to enhance the user interface and the use of the camera. They are used to automatically determine the orientation in which the user is holding the phone and use that information to automatically re-orient the display between a landscape and portrait view or correctly orient captured photos during viewing on the phone.

Figure 2.1 shows the suite of sensors found in the Apple iPhone 4. The phone's sensors include a gyroscope, compass, accelerometer, proximity sensor, ambient light sensor as well as other more conventional devices that can be used to sense including front and back facing cameras, a microphone, GPS and WiFi and bluetooth radios. Many of the newer sensors are added to support the user interface (e.g., the accelerometer) or to augment location base services (e.g., the digital compass).

The proximity and light sensors allow the phone to perform simple forms of context recognition associated with the user interface. The proximity sensor detects, for example, when the user holds the phone to their face to speak. In this case, the touchscreen and keys are disabled preventing them from accidentally being pressed as well as saving power because the screen is turned off. Light sensors are used to adjust the brightness of the screen. The GPS, which allows the phone to localize itself, enables new location-based applications, for example local search, mobile social networks and navigation. The compass and gyroscope represent an extension of location providing the phone with increased awareness of its position in relation to the physical world (e.g., its direction and orientation) enhancing location-based applications.

Not only are these sensors useful in driving the user interface and providing location base services but they represent a significant opportunity to gather data about

people and their environment. For example, accelerometer data is capable of characterizing the physical movements of the user carrying the phone [73]. Distinct patterns within the accelerometer data can be exploited to automatically recognize different activities (e.g., running, walking, standing). The camera and microphone are powerful sensors. These are probably the most ubiquitous sensors on the planet. By collecting continuously audio from the phone's microphone for example, it is possible to classify a diverse set of distinctive sounds associated with a particular context or activity in a person's life, for instance, using an ATM machine, being in a particular coffee shop, having a conversation, listening to music, making coffee, driving [67]. The camera on the phone can be used for many things including traditional tasks like photo blogging to more specialized sensing activities, for instance, tracking the user's eye movement across the phone's display as a means to activate applications using the camera mounted on the front of the phone [74] The combination of accelerometer data and a stream of location estimates from the GPS can recognize the mode of transportation of a user, namely using a bus, bike, car or taking the subway [75].

More and more sensors are being incorporated into phones. An interesting question is what new sensors are we likely to see over the next few years? Non phone based mobile sensing devices including the Intel Mobile Sensing Platform (MSP) [25] have shown value from using other sensors not found in phones today (e.g., barometer, temperature, humidity sensors) for activity recognition; for example, the accelerometer and barometer makes it easy to identify not only when someone is walking but when they are climbing stairs and in which direction. Other researchers have studied air quality and pollution [46] using specialized sensors embedded in prototype mobile phones. Others still have embedded sensors in standard mobile phone ear-phones to read a person's blood pressure [87] or used neural signals from cheap off-the-shelf wireless electroencephalography (EEG) headsets to control mobile phones for hands-free human-mobile phone interaction [23]. At this stage it is too early to say what new sensors will be added to the next generation of smartphones but as the cost and form factor come down and leading applications emerge we are likely to see more sensors added.

2.3 Applications and App Stores

New classes of applications, which can take advantage of both the low-level sensor data and high-level events, context and activities inferred from mobile phone sensor data, are being explored not only in academic and industrial research labs [5, 10, 16,

22, 24, 54, 67, 77, 112] but also within start-up companies and large corporations (e.g., Microsoft, Google, Fitbit, Loopt, Nike/Apple). One such example is SenseNetworks, a recent U.S. based start-up company, which uses millions of GPS estimates sourced from mobile phones within a city to predict, for instance, which bars and night-clubs will be most popular with specific clusters of people called tribes within the wider population. Remarkably, it has only taken a few years for the this type of analysis of large-scale location information and mobility patterns to migrate from the research lab into commercial usage.

The combination of high-resolution sensor streams and the potential for scale can enable applications that are capable of making transformative changes within multiple application domains. In what follows, we discuss a number of emerging lead application domains and the importance of new application delivery channels (i.e., app stores) offered by all the major vendors.

Transportation. Traffic remains a serious global problem; for example, congestion alone can severely impact both the environment and human productivity (e.g., wasted hours due to congestion). Mobile phone sensing systems including the MIT VTrack project [102] or the Mobile Millennium project [6] (a joint initiative between Nokia, NAVTEQ and UC Berkeley) are being used to provide fine-grain traffic information at large scale using mobile phones that facilitate services like accurate travel time estimation for improve commute planning.

Social Networking. Millions of people participate regularly within online social networks. The Dartmouth CenceMe project [73] is investigating the use of sensors in the phone to automatically classify events in people's lives called sensing presence and selectively share this "presence" using online social networks, for instance Twitter, Facebook and MySpace, replacing manual actions people now perform daily.

Environmental Monitoring. Conventional ways of measuring and reporting environmental pollution statistics are limited. Reported information, like air quality, is an aggregate measurement which applies coarsely to a community, for instance an entire city. The UCLA PEIR project [75] uses sensors in phones to build a system that enables personalized environmental impact reports that track how the actions of individuals effect both their exposure and their contribution to problems including carbon emissions.

Health and Well-being. The information used for personal health care today largely comes from self-report surveys and infrequent doctor consultations. Sensor-enabled mobile phones have the potential to collect in-situ continuous sensor data that can dramatically change the way health and wellness are assessed as well as how care

and treatment are delivered. The Ubitfit Garden [28], a joint project between Intel and the University of Washington, captures levels of physical activity and relates this information to personal health goals when presenting feed back to the user. These types of systems have proven to be effective in empowering people to curb poor behavior patterns and improve health including encouraging more exercise or decrease the risk of contracting HIV [108].

App Stores. Getting a critical mass of users is a common problem faced by people who build systems: developers and researchers alike. Fortunately, modern phones have an effective application distribution channel first made available by Apple's *App Store* for the iPhone that is revolutionizing this new field. Each major smartphone vendor has an "app store" (e.g., Apple AppStore, Android Market, Microsoft Mobile Marketplace, Nokia Ovi). The success of the app stores with the public has made it possible for not only start-ups but small research labs and even individual developers to quickly attract a very large number of users. For example, an early use of app store distribution by researchers in academia is the CenceMe application for iPhone [73] which was made available on the App Store when it opened in 2008. It is now feasible to distribute and run experiments with a large number of participants from all around the world rather than in laboratory controlled conditions using a small user study. For instance, researchers interested in statistical models that interpret human behavior from sensor data have long dreamt of ways to collect such large-scale real-world data. These app stores represent a game changer for research. Many challenges remain with this new approach to experimentation via app stores. For example, what is the best way to collect ground-truth data to assess the accuracy of algorithms that interpret sensor data? How do we validate experiments? How do we select a good study group? How do we deal with the potential massive amount of data made available? How do we protect the privacy of users? What is the impact on getting approval for human subjects studies from university Institutional Review Boards (IRBs)? How do researchers scale to run such large scale studies? For example, researchers use to supporting small numbers of users (e.g., 50 users with mobile phones) now have to construct cloud services to potentially deal with 10,000 needy users. This fine if you are a start up but are academic research labs geared to deal with this?

2.4 Sensing Scale and Paradigms

Future mobile phone sensing systems will operate at multiple scales, enabling everything from personal sensing to global sensing as illustrated in Figure 2.2 where we

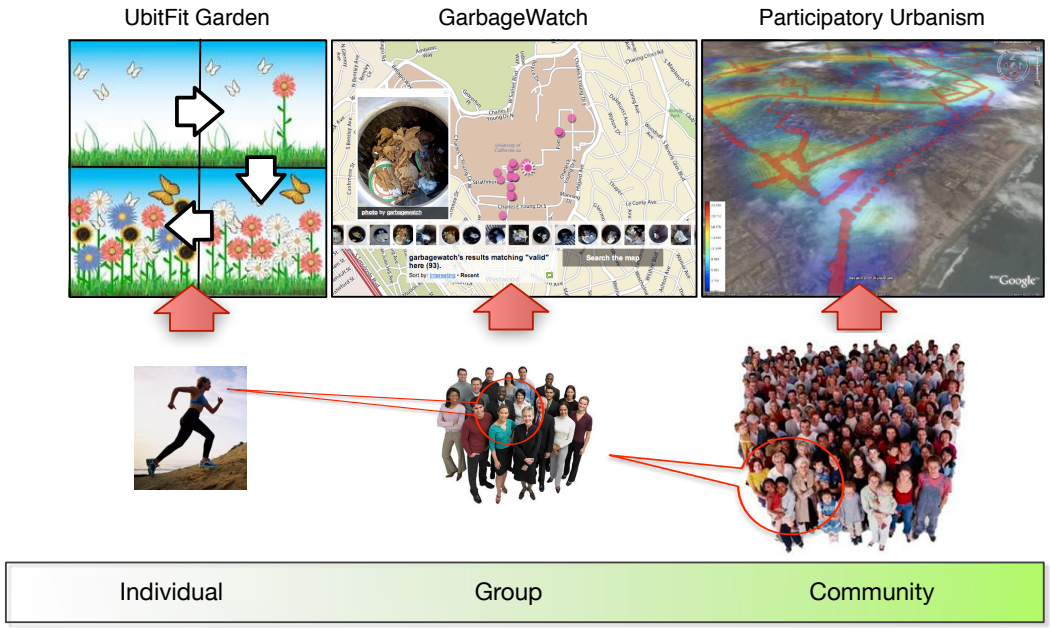


Figure 2.2: Mobile phone sensing is effective across multiple scales, including: a single individual (e.g., UbitFit Garden [28]), groups for instance social networks or special interest groups (e.g., GarbageWatch [3]), and entire communities/ population of a city (e.g., Participatory Urbanism [5]).

see personal, group and community sensing - three distinct scales at which mobile phone sensing is currently being studied by the research community. At the same time researcher's are discussing how much the user (i.e., the person carrying the phone) should be actively involved during the sensing activity (e.g., taking the phone out of the pocket to collect a sound sample or take a picture); that is, should the user actively participate, known as *participatory sensing* [22], or, alternatively, passively participate known as *opportunistic sensing* [24]. Each of these sensing paradigms present important tradeoffs. In what follows, we discuss different sensing scales and paradigms.

2.4.1 Sensing Scale

Personal sensing applications are designed for a single individual and are often focused on data collection and analysis. Typical scenarios include tracking the user's exercise routines or automating diary collection. Typically, personal sensing applications generate data for the sole consumption of the user and are not shared with others. An exception is healthcare applications where limited sharing with medical professionals is common (e.g., primary care giver or specialist). Figure 2.2 shows the

UbitFit Garden [28] as an example of a personal wellness application. This personal sensing application adopts persuasive technology ideas (discussed further in Section 2.8.3) to encourage the user to reach their personal fitness goals using the metaphor of a garden blooming as the user progresses toward their goals.

Individuals who participate in sensing applications that share a common goal, concern or interest collectively represent a group. These *group sensing* applications are likely to be popular and reflect the growing interest in social networks or connected groups (e.g., at work, in the neighborhood, friends) who may want to share sensing information freely or with privacy protection. There is an element of trust in group sensing applications that simplify otherwise difficult problems, for instance, attesting that the collected sensor data is correct or reducing the degree to which aggregated data must protect the individual. Common use-cases include assessing neighborhood safety, sensor-driven mobile social networks or forms of citizen science. Figure 2.2 shows GarbageWatch [3] as an example of a group sensing application where people participate in a collective effort to improve recycling by capturing relevant information needed to improve the recycling program. For example, students use the phone's camera to log the content of recycling bins used across a campus.

Most examples of *community sensing* only become useful once they have a large number of people participating; for example, tracking the spread of disease across a city, the migration patterns of birds, congestion patterns across city roads [6], or a noise map of a city [90]. These applications represent large-scale data collection, analysis and sharing for the good of the community. To achieve scale implicitly requires the cooperation of strangers who will not trust each other. This increases the need for community sensing systems with strong privacy protection and low commitment levels from the users. Figure 2.2 shows Carbon Monoxide readings captured in Ghana using mobile sensors attached to taxi cabs as part of the Participatory Urbanism project [5] as an example of a community sensing application. This project in conjunction with the the N-SMARTs project [46] at UC Berkeley are developing prototypes that allow similar sensor data to be collected with phone embedded sensors.

The impact of scaling a sensing application from personal to population scale is unknown. Many issues of sharing, privacy, data mining, learning, and closing the loop in terms of useful feed back to an individual, group, community and population remain open. Today, we only have limited experience in building scalable sensing systems.

2.4.2 Sensing Paradigms

One issue common to the different types of sensing scale is to what extent the user is actively involved in the sensing system [56]. We discuss two points in the design space, namely, participatory sensing, where the user actively engages in the data collection activity (i.e., the user manually determines how, when, what and where to sample), and alternatively, opportunistic sensing, where the data collection stage is fully automated with no user involvement.

The benefit of opportunistic sensing is that it lowers the burden placed on the user, allowing the overall participation by a population of users to remain high even if the application is not that personally appealing. This is particularly useful for community sensing where per user benefit may be hard to quantify and only accrue over a long time. However, often these systems are technically difficult to build (e.g., [31]) and a major resource, people, are under utilized. One of the main challenges of using opportunistic sensing is the phone context problem; for example, the application wants to only take a sound sample for a city wide noise map when the phone is out of the pocket or bag. These types of context issues can be solved by using the phone sensors; for example, the accelerometer or light sensors can determine if the phone is out of the pocket.

Participatory sensing which is gaining interest in the mobile phone sensing community places a higher burden or cost on the user; for example, manually selecting data to collect (e.g., lowest petrol prices [21]) and then sampling it (e.g., taking a picture). An advantage is that complex operations can be supported by leveraging the the intelligence of the person in the loop who can solve the context problem in an efficient manner; that is, a person who wants to participate in collecting a noise or air quality map of their neighborhood simply takes the phone out of their bag to solve the context problem. One drawback of participatory sensing is that the quality of data is dependent on participant enthusiasm to reliability collect sensing data and the compatibility of a persons mobility patterns to the intended goals of the application (e.g., collect pollution samples around schools). Many of these challenges are actively being studied. For example, the PICK project [91] is studying models for systematically recruiting participants.

Clearly, opportunistic and participatory represent two extreme points in the design space. Each approach has pros and cons. To date there is little experience in building large scale participatory or opportunistic sensing applications to fully understand the tradeoffs. There is a need to develop models to best understand the usability and performance issues of these schemes. In addition, it is likely that many applications

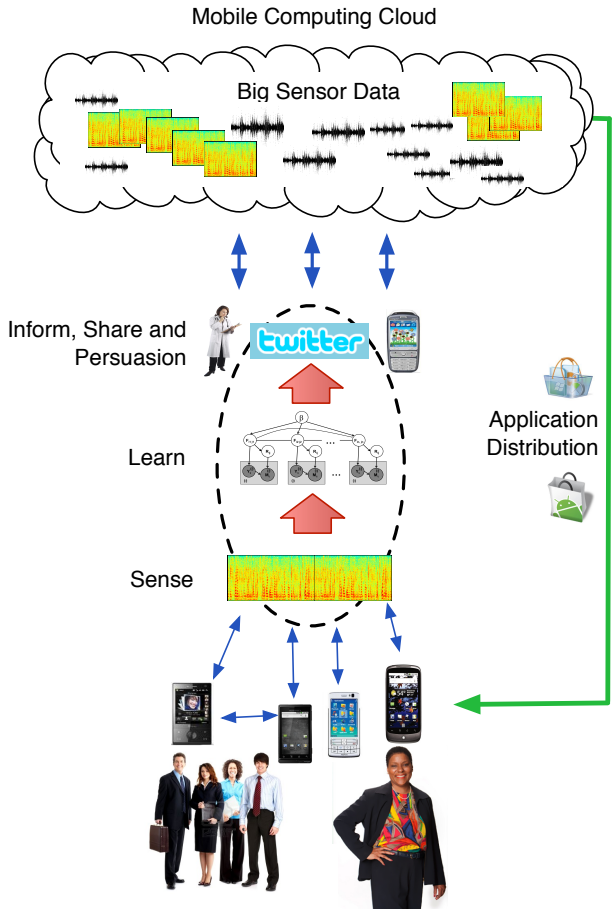


Figure 2.3: Mobile Phone Sensing Architecture

will emerge that represent a hybrid of both of these sensing paradigms.

2.5 Mobile Phone Sensing Architecture

Mobile phone sensing is in its infancy. There is little or no consensus on the sensing architecture for the phone and the cloud. For example, new tools and phone software will be needed to facilitate quick development and deployment of robust classifiers for the leading phones on the market. Common methods for collecting and sharing of data need to be developed. Mobile phones cannot be overloaded with continuous sensing commitments that undermine the performance of the phone, for example, in terms of depleting battery power. It is not clear what architectural components should run on the phone and what should run in the cloud. For example, some researchers propose that raw sensor data should not be pushed to the cloud because of privacy issues. In the following sections, we propose a simple architectural viewpoint for the mobile

phone and the computing cloud as a means to discuss the major architectural issues that need to be addressed. We do not argue that this is the best system architecture. Rather, it presents a starting point for discussions that we hope will eventually lead to a converging view and move the field forward.

Figure 2.3 shows a mobile phone sensing architecture that comprises the following building blocks.

Sense. Individual mobile phones collect raw sensor data from sensors embedded in the phone.

Learn. Information is extracted from the sensor data by applying machine learning and data mining techniques. These operations either occur directly on the phone, in the mobile cloud or with some partitioning between the phone and cloud. Where these components run could be governed by various architectural considerations, for example, privacy, providing user real-time feedback, reducing communication cost between the phone and cloud, available computing resources, and sensor fusion requirements. We therefore consider where these components run to be an open issue that requires research.

Inform, Share, and Persuade. We bundle a number of important architectural components together because of commonality or coupling of the components. For example, a personal sensing application will only inform the user, whereas a group or community sensing application may share an aggregate version of information with the broader population and obfuscate the identity of the users. Other considerations are how to best visualize sensor data for consumption of individuals, groups and communities. Privacy is a very important consideration as well.

While phones will naturally leverage the distributed resources of the mobile cloud (e.g., computation and services offered the cloud) the computing, communications, and sensing resources on the phones are ever increasing. We believe that as resources of the phone rapidly expand one of the main benefits of using the mobile computing cloud will be the ability to compute and mine big data from very large numbers of users. The availability of large-scale data benefits mobile phone sensing in a variety of ways; for example (i) more accurate interpretation algorithms that are updated based on sensor data sourced from an entire user community (see Section 2.7.2 for more details); (ii) this data enables personalizing of sensing systems based on the behavior of both the individual user and cliques of people with similar behavior (see Section 2.8.2).

In the remainder of the article we present a detailed discussion of the three main architectural components introduced in this section; that is: (i) sense, (ii) learn and

(iii) inform, share and persuade.

2.6 Sense: The Mobile Phone as a Sensor

As we discuss in Section 2.2, the integration of an ever expanding suite of embedded sensors is one of the key drivers of mobile phone applications. However, the programmability of the phones and the limitation of the operating systems that run on them, the dynamic environment presented by user mobility, and the need to support continuous sensing on mobile phones presents a diverse set of challenges that the research community needs to address.

2.6.1 Programability

Until very recently only a handful of mobile phones could be programmed. Popular platforms like the Symbian based phones presented researchers with sizable obstacles to building mobile sensing applications [73]. These platforms lacked well defined reliable interfaces to access low level sensors and were not well suited to writing common data processing components, including signal processing routines, or performing computationally costly inference due to the resource constraints of the phone. The early sensor-enabled phones (i.e., prior to the iPhone in 2007) for instance the Symbian-based Nokia N80 included an accelerometer but there were no open APIs to access the sensor signals. This has changed significantly over the last few years. Note that phone vendors initially included accelerometers to help improve the user interface experience.

Most of the smartphones on the market are open and programmable by third party developers and offer SDKs, APIs and software tools. It is easy to cross-compile code and leverage existing software like established machine learning libraries (e.g., Weka [109]).

However, a number of challenges remain in the development of sensor-based applications. Most vendors did not anticipate that third parties would use continuous sensing to develop new applications. As a result, there is mixed API and OS support to access low level sensors, fine-grain sensor control, and watch-dog timers that are required to develop real-time applications. For example, on Nokia Symbian and Maemo phones the accelerometer returns samples to an application unpredictably between 25 to 38 Hz, depending on the CPU load. While this might not be an issue when using the accelerometer to drive the display, using statistical models to interpret activity

or context typically requires high and at least consistent sampling rates.

Lack of sensor control limits the management of energy consumption on the phone. For instance, the GPS uses a varying amount of power depending on the factors like the number of satellites available and atmospheric conditions. Currently, phones only offer a black-box interface to the GPS to request location estimates. Finer grain control is likely to help in preserving battery power and maintaining accuracy, for example, location estimation could be aborted when accuracy is likely to be low or if the estimate takes too long and is no longer useful.

As third parties demand better support for sensing applications, the API and OS support will improve. However, programmability of the phone remains a challenge moving forward. As more individual, group, and community scale applications are developed there will be an increasing demand placed on phones, both individually and collectively. It is likely that abstractions that can cope with persistent spatial queries and secure the use of the resources from neighboring phones will be needed. Phones may want to interact with other co-located phones to build new sensing paradigms based on collaborative sensing [92] . Different vendors offer different APIs making porting the same sensing application to multi-vendor platforms challenging. It is useful for the research community to think about and propose sensing abstractions and APIs that could be standardized and adopted by different mobile phone vendors.

2.6.2 Continuous Sensing

Continuous sensing will enable new applications across a number of sectors but particularly in personal healthcare. One important OS requirement for continuous sensing is that the phone supports multitasking and background processing. Today, only Android and Nokia Maemo phones support this capability. The iPhone OS 4 while supporting the notion of multitasking is inadequate for continuous sensing. Applications must conform to predefined profiles with strict constraints on access to resources. None of these profiles provide the ability to have continuous access to all the sensors.

While smartphones continue to provide more computation, memory, storage, sensing and communications bandwidth the phone is still a resource limited device if complex signal processing and inference is required. Signal processing and machine learning algorithms can stress the resources of the phones in different ways: some require the CPU to process large volumes of sensor data (e.g., interpreting audio data [67], some need frequent sampling of energy expensive sensors (e.g., GPS [75]), while others require real-time inference (e.g., Darwin [71]) . Different applications

place different requirements on the execution of these algorithms. For example, for applications that are user initiated the latency of the operation is important. Applications (e.g., healthcare) that require continuous sensing will often require real-time processing and classification of the incoming stream of sensor data. We believe continuous sensing can enable a new class of real-time applications in the future, but these applications may be more resource demanding. Phones in the future should offer support for continuous sensing support without jeopardizing the phone experience; that is, not disrupt existing applications (e.g., to make calls, text and surf the web) or drain the batteries. Experiences from actual deployments of mobile phone sensing systems show that phones that run these applications can have standby times reduced from 20 hours or more to six hours [73] . For continuous sensing to be viable there needs to be breakthroughs in low energy algorithms that duty cycle the device while maintaining the necessary application fidelity.

Early deployments of phone sensing systems tended to trade-off accuracy for lower resource usage by implementing algorithms that require less computation or a reduced amount of sensor data. Another strategy to reduce resource usage is to leverage cloud infrastructure where different sensor data processing stages are offloaded to backend servers [30, 71, 73] when possible. Typically, raw data produced by the phone is not sent over the air due to the energy cost of transmission but rather compressed summaries (i.e., extracted features from the raw sensor data) are sent. The drawback to these approaches is that they are seldom sufficiently energy efficient enough to be applied to continuous sensing scenarios. Other techniques rely on adopting a variety of duty-cycling techniques that manage the sleep cycle of sensing components on the phone in order to tradeoff the amount of battery consumed against sensing fidelity and latency (e.g., [107]).

Continuous sensing raises considerable challenges in comparison to sensing applications that require a short time-window of data or a single snapshot, e.g., a single image or short sound clip. There is an energy tax associated with continuously sensing and potentially uploading in real-time to the cloud for further processing. Solutions that limit the cost of continuous sensing and reduce the communication overheard are necessary. If the interpretation of the data can withstand delays of an entire day, it might be acceptable if the phone can collect and store the sensor data until the end of the day and upload when the phone is being charged. However this delay-tolerant model of sensor sampling and processing severely limits the ability of the phone to react and be aware of its context. Sensing applications that will be successful in the real-world will have to be 'smart' enough to adapt to situations. There is a need to

study the trade off of continuous sensing with the goal of minimizing the energy cost but offering sufficient accuracy and real-time responsiveness to make the application useful.

As continuous sensing becomes more common, it is likely that additional processing support will emerge. For example, the Little Rock project [89] underway at Microsoft Research is developing hardware support for continuous sensing where the primary CPU frequently sleeps and DSPs support the duty-cycle management, sensor sampling and signal processing.

2.6.3 Phone Context

Mobile phones are often used on-the-go and in ways that are difficult to anticipate in advance. This complicates the use of statistical models that may fail to generalize under unexpected environments. The background environment or actions of the user (e.g., the phone could be in the pocket) will also affect the quality of the sensor data that is captured. Phones may be exposed to events for too short a period of time, if the user is traveling quickly (e.g., in a car), if the event is localized (e.g., a sound) or if the sensor requires more time than is possible to gather a sample (e.g., air quality sensor). Other forms of interfering context include a person using their phone for a call which interferes with the ability of the accelerometer to infer the physical actions of the person. We collectively describe these issues as the *context problem*. Many issues remain open in this area.

Some researchers propose to leverage co-located mobile phones to deal with some of these issues; for example, sharing sensors temporarily if they are better able to capture the data [34]. To counter context challenges researchers proposed supersampling [46] where data from nearby phones are collectively used to lower the aggregate noise in the reading. Alternatively, an effective approach for some systems have been sensor sampling routines with admission control stages that do not process data that is low-quality, saving resources and reducing errors (e.g., SoundSense [67]).

While machine learning techniques are being used to interpret mobile phone data, the reliability of these algorithms suffer under the dynamic and unexpected conditions presented by everyday phone use. For example, a speaker identification algorithm maybe effective in a quiet office environment but not a noisy cafe. Such problems can be overcome by collecting sufficient examples of the different usage scenarios (i.e., training data). However, acquiring examples is costly and anticipating the different scenarios the phone might encounter is almost impossible. Some solutions to this

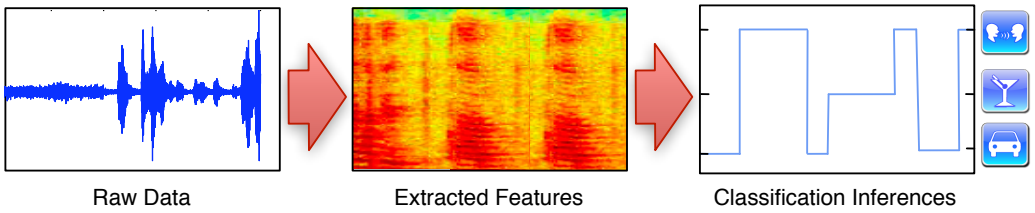


Figure 2.4: Raw audio data captured from a mobile phones is transformed into features allowing learning algorithms to identify classes of behavior (e.g., driving, in conservation, making coffee) occurring in a stream of sensor data, for example, by SoundSense [67].

problem straddle the boundary of mobile systems and machine learning and include borrowing model inputs (i.e., features) from nearby phones, [57], performing collaborative multi-phone inference with models that evolve based on different scenarios encountered, [71], or discovering new events that are not encountered during application design [67].

2.7 Learn: Interpreting Sensor Data

The raw sensor data able to be acquired by phones, irrespective of the scale or modality (e.g., accelerometer, camera) are worthless without interpretation (e.g., human behavior recognition). A variety of data mining and statistical tools can be used to distill information from the data collected by mobile phones and calculate summary statistics to present to the users, for instance, the average emissions level of different locations or the total distance run by a user and their ranking within a group of friends (e.g., Nike+ [7]).

Recently, crowd-sourcing techniques have been applied to the analysis of sensor data which is typically problematic; for example, image processing when used in-the-wild is notoriously difficult to maintain high accuracy. In the CrowdSearch [112] project, crowd-sourcing and micro-payments are adopted to incentivize people to improve automated image search. In [112] human-in-the-loop stages are added to the process of image search with tasks distributed to the user population.

In what follows, we discuss the key challenges in interpreting sensor data, focusing on a primary area of interest: human behavior and context modeling.

2.7.1 Human Behavior and Context Modeling

Many emerging applications are people-centric and modeling the behavior and surrounding context of the people carrying the phone is of particular interest. A natural question is how well can mobile phones interpret human behavior (e.g., sitting in conservation) from low-level multi-modal sensor data? Or similarly how accurately can they infer the surrounding context (e.g., pollution, weather, noise environment)?

Currently, supervised learning techniques are the algorithms of choice in building mobile inference systems. In supervised-learning, as illustrated in Figure 2.4, examples of high-level behavioral classes (e.g., cooking, driving) are hand annotated (i.e., labeled). These examples, referred to as training data, are then provided to a learning algorithm which fits a model to the classes (i.e., behaviors) based on the sensor data. Sensor data is usually presented to the learning algorithm in the form of extracted features, which are calculations on the raw data that emphasize characteristics that more clearly differentiate classes (e.g., the variance of the accelerometer magnitude over a small time window could be useful for separating standing and walking classes). Supervised learning is feasible for small scale sensing application but unlikely to scale to handle the wide range of behaviors and contexts exhibited by a large community of users. Other forms of learning algorithms, including semi-supervised (where only some of the data is labeled) and unsupervised (where no labels are provided by the user) learning algorithms reduce the need for labeled examples, but can lead to classes that do not correspond to the activities that are useful to the application or require that the unlabeled data only come from the already labeled class categories (e.g., an activity that was never encountered before can throw off a semi-supervised learning algorithm).

Researchers show that a variety of everyday human activities can be inferred, most successfully, from multi-modal sensor streams For example, [59] describes a system which is capable of recognizing 8 different everyday activities (e.g., brushing teeth, riding in an elevator) using the Mobile Sensing Platform (MSP) [25] - an important mobile sensing device that is a predecessor of sensing on the mobile phone. Similar results are demonstrated using mobile phones that infer everyday activities (e.g., [73, 75, 84]), albeit less accurately and with a smaller set of activities than the MSP.

The microphone, accelerometer and GPS found on all smartphones on the market have proven to be effective at inferring more complex human behavior. Early work on mobility pattern modeling succeed with surprisingly simple approaches to identify significant places in the life people (e.g., work, home, coffee shop). More recently

researchers [63] have used statistical techniques to not only infer significant places but also connect these to activities (e.g., gym, waiting for the bus) using just GPS traces. The microphone is one of the most ubiquitous sensors and is capable of inferring what a person is doing (e.g., in conversation), where they are (e.g., audio signature of a particular coffee shop) – in essence, it can capture a great deal both about a person and their surrounding ambient environment. In SoundSense [67] a general purpose sound classification system for mobile phones is developed using a combination of supervised and unsupervised learning. The recognition of a static set of common sounds (e.g., music) uses supervised learning but augmented with an unsupervised approach that learns the novel frequently reoccurring classes of sound encountered by different users. Finally, the user is brought into the loop to confirm and provide a textual description (i.e., label) of the discovered sounds. As a result, SoundSense extends the ability of the phone to recognize new activities.

2.7.2 Scaling Models

Existing statistical models are unable to cope with everyday occurrences for instance a person using a new type of exercise machine and struggle when two activities overlap each other or when different individuals carry out the same activity differently (e.g., the sensor data for walking will look very different for a ten year old vs. a ninety year old person). A key to scalability is to design techniques for generalization that will be effective for entire communities containing millions of people.

To address these concerns (see also Section 2.6.3) current research directions point towards models that are adaptive and incorporate people in the process. Automatically increasing the classes recognized by a model using active learning (where the learning algorithm selectively queries the user for labels) is investigated in the context of heath care [66]. Approaches have been developed in which training data sourced directly from users is grouped based on their social-network [57]. This work demonstrates exploiting the social network of the users improves the classification of location for example significant places. Community-guided learning [84] combines data similarity and crowd-sourced labels to improve the classification accuracy of the learning system. In [84] hand annotated labels are no longer treated as absolute ground truth during the training process but are treated as soft-hints as to class boundaries in combination with the observed data similarity. This approach learns classes (i.e., activities) based on the actual behavior of the community and adjusts transparently to the changes in how the community perform these activities – making it more suitable

for large-scale sensing applications. However, if the models need to be adapted on the fly, this may force the learning of models to happen on the phone potentially causing significant increase to computational needs [71].

Many questions remain regarding how learning will progress as the field grows. There is a lack of shared technology that could help accelerate the work. For example, each research group develops their own classifiers that are hand coded and tuned. This is time consuming and mostly based on small-scale experimentation and studies. There is a need for a common machine learning toolkit for mobile phone sensing that allows researchers to build and share models. Similarly, there is a need for large-scale public data sets to study more advance learning techniques and rigorously evaluate the performance of different algorithms. There is also a need for a repository for sharing datasets, code and tools to support the researchers.

2.8 Inform, Share and Persuasion: Closing the Sensing Loop

How you use inferred sensor data to inform the user is application specific. But, a natural question is once you infer a class or collect together a set of large-scale inferences how do you close the loop with people? Clearly, personal sensing applications would just inform the individual while social networking sensing application may share activities or inferences with a friends. We discuss these forms of interaction with users as well as the important area of privacy. Another topic we touch on is using large-scale sensor data as a persuasive technology – in essence using big data to help users attain goals using targeted feedback.

2.8.1 Sharing

To harness the potential of mobile phone sensing requires effective methods of allowing people to connect with and benefit from the data. The standard approach to sharing is visualization using a web portal where sensor data and inferences are easily displayed. This offers a familiar and intuitive interface. For the same reasons, a number of phone sensing systems connect with existing web applications either to enrich existing applications or make the data more widely accessible (e.g., [73, 75]). Researchers recognize the strength of leveraging social media outlets including Facebook, Twitter, Flickr as ways to not only disseminate information but build community awareness (e.g., citizen science [5]). A popular application domain is fitness,

for instance Nike+. Such systems combine individual statistics and visualizations of sensed data and promote competition between users. The result is the formation of communities around a sensing application. Even though, as in the case of Nike+, the the sensor information is rather simple (i.e., simply the time and distance of a run) people still become very engaged. Other applications have emerged that are considerably more sophisticated in the type of inference made but have had limited up take. It is still too early to predict which sensing applications will become the most compelling for user communities. But social networking provides many attractive ways to share information.

2.8.2 Personalized Sensing

Mobile phones are not limited to simply collecting sensor data. For example, both the Google and Microsoft search clients that run on the iPhone allow users to search using voice recognition. Eye tracking and gesture recognition are also emerging as a natural interfaces to the phone.

Sensors are used to monitor the daily activities of a person and profile their preferences and behavior making personalized recommendations for services, products or points of interest possible [65]. The behavior of an individual along with an understanding of how behavior and preferences relate to other segments of the population with similar behavioral profiles can radically change not only online experiences but real-world ones too. Imagine walking into a chemist store and your phone suggesting vitamins and supplements with the effectiveness of a doctor. At a clothing store your phone could identify which items are manufactured without sweatshop labor. The behavior of the person, as captured by sensors embedded in their phone, become an interface that can be fed to many services (e.g., targeted advertising). Sensor technology personalized to the user's profile empower them to make more informed decisions across a spectrum of services.

2.8.3 Persuasion

Sensor data gathered from communities (e.g., fitness, healthcare) can be used not only to inform users but to persuade them to make positive behavioral changes. For example, nudge users to exercise more or to smoke less. Systems that provide tailored feedback with the goal of changing the users behavior is referred to as persuasive technology [36]. Mobile sensing applications open the door to building novel persuasive systems which are still largely unexplored.

For many application domains, for instance healthcare or environmental awareness, users commonly have desired objectives; for instance, to loose weight or lower carbon emissions. Simply providing a user with their own information is often not enough to motivate a change of behavior or habit. Mobile phones are an ideal platform capable for using low-level individual-scale sensor data and aggregated community-scale information to drive long term change (e.g., contrasting the carbon footprint of a user with her friends can persuade the user to reduce her own footprint) . The Ubfit Garden [28] project is an early example of integrating persuasion and sensing on the phone. Ubfit uses an ambient background display on the phone to offer the user continuous updates on their behavior in response to desired goals. The display uses the metaphor of a garden with different flowers blooming in response to physical exercise of the user during their day. It does not use comparison data but simply targets the individual user. A natural extension of Ubfit is to present community data. Ongoing research is exploring methods of identifying and using people in a community of users as "influencers" for different individuals in the user population. A variety of techniques are used in existing persuasive system research, for example, the use of games, competitions among groups of people, sharing information within a social network, or goal setting accompanied with feedback. Understanding which types of metaphors and feedback are the most effective for various persuasion goals is still an open research problem. Building mobile phone sensing systems that integrate persuasion requires interdisciplinary research that combines behavioral and social psychology theories with computer science.

The use of large volumes of sensor data provided by mobile phones provide an exciting opportunity and is likely to enable new applications that have promise in enacting positive social changes in health and the environment over the next several years. The combination of large-scale sensor data combined with accurate models of persuasion could revolutionize how we deal with persistent problems in our lives ranging from chronic disease management, depression, obesity or even voter participation.

2.8.4 Privacy

Respecting the privacy of the user is perhaps the most fundamental responsibly of a phone sensing system. People are understandably sensitive about how sensor data is captured and used, especially if the data reveals users location, speech, or potentially sensitive images. Although there are existing approaches that can help with these

problems (e.g., cryptography, privacy-preserving data mining) they are often insufficient (e.g., [51]). For instance, how can the user temporarily pause the collection of sensor data without causing a suspicious 'gap' in the data stream that would be noticeable to anyone (e.g., family or friends) with whom they regularly share data?

In personal sensing applications, processing data locally may provide privacy advantages compared to using remote more powerful servers. SoundSense [67] adopts this strategy, all the audio data is processed on the phone and raw audio is never stored. Similarly, the UbiFit Garden [28] application processes all data locally on the device.

Privacy for community-based sensing applications is based on group membership. For instance, although social networking applications like Loopt and CenceMe [73] share sensitive information (e.g., location and activity) they do so within groups in which users have an existing trust relationship based on friendship or a shared common interest for instance reducing their carbon footprint.

Community sensing applications that can collect and combine data from millions of people run the risk of unintended leakage of personal information. The risks from location-based attacks are fairly well understood given years of previous research. However, our understanding of the dangers of other modalities (e.g., activity inferences, social network data) are less developed. There are growing examples of reconstruction type attacks where data that may look safe and innocuous to an individual user may allow invasive information to be reverse-engineered. For example, the UIUC Poolview project shows that even careful sharing of personal weight data within a community can expose information on whether a user's weight is trending upwards or downwards [40]. [75] evaluates different countermeasures like adding noise to the data or replacing chunks of the data with synthetic but realistic samples that do not impact the quality of the aggregate analysis.

Privacy and anonymity will remain a significant problem in mobile phone based sensing for the foreseeable future. In particular, the "'second-hand smoke" problem of mobile sensing creates new privacy challenges, including: how can the privacy of 3rd parties be effectively protected when other people wearing sensors are nearby? How can mismatched privacy policies be managed when two different people are close enough to each other for their sensors to collect information from the other party? Furthermore, this type of sensing presents even larger societal questions, for example, who is responsible when collected sensor data from these mobile devices cause financial harm? Stronger techniques for protecting the rights of people as sensing becomes more common place will be necessary.

2.9 Summary

This chapter presents the current state of the art and open challenges in the emerging field of mobile phone sensing. The primary obstacle to this new field is not a lack of infrastructure, millions of people already carry phones with rich sensing capabilities. Rather, the technical barriers are related to performing privacy-sensitive and resource-sensitive reasoning with noisy data and noisy labels and providing useful and effective feedback to users. We believe once these technical barriers are overcome this nascent field will advance quickly, acting as a disruptive technology across many domains including social networking, health and energy.

A central theme of this thesis is that many of these barriers can be overcome by leveraging communities of users within the sensing system. In the proceeding three chapters we explore this topic as it relates to the building of robust and scalable classification models of human behavior, context and events. We begin by describing in the next chapter a set of experiments which exploit collaboration between two types of communities; specifically, ad-hoc in-situ collaboration between communities of nearby mobile devices and post-facto collaboration within a social network of people. Insights from these experiments were applied in the subsequent chapters, the first of which addresses the deficiencies of crowd-sourced noisy labels with the second overcoming the population diversity problem.

Chapter 3

Leveraging Social and Opportunistic Networks

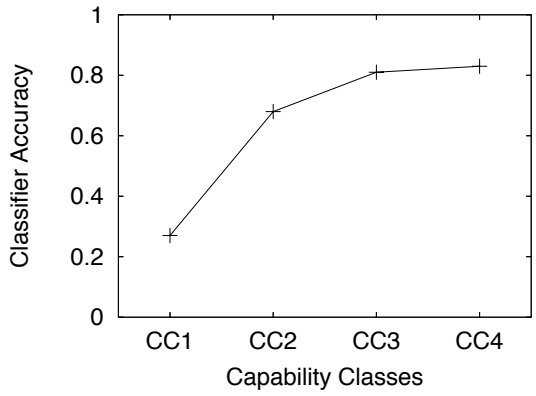
3.1 Introduction

We begin our examination of community-guided approaches to boosting classification model performance with a series of exploratory experiments presented in this chapter. The results of these experiments clearly demonstrate the potential of community grounded approaches and guide our later investigations. In this chapter we frame the challenges facing the construction of accurate inference models as being: i) the lack of appropriate data inputs (i.e., features) and ii) the time and effort that must be spent in training a model of sufficient accuracy. Two forms of communities are considered. In the first, physical communities of co-located mobile devices cooperate to make inferences using the similar views available to each device of the same nearby event. In the second, virtual communities comprised by the users of mobile devices connected by social networks are employed. This community collaborates to make inferences by leveraging similar behavioral patterns (allowing, for instance, a model from one user to be potentially useful for another) or the overlap in labels semantics due to close social ties (allowing, for example, training data to be shared). In what remains of this section we describe the two techniques we proposed in this chapter for exploiting each form of community and discuss in more depth motivating scenarios.

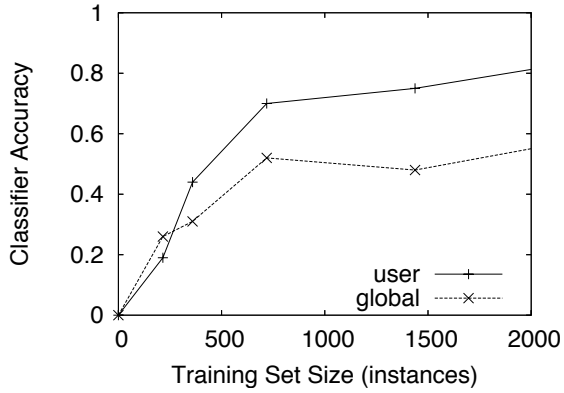
The consumer-device-based sensing substrate upon which people-centric applications are built is characterized by heterogeneity in terms of sensing and other resources (e.g., memory, battery capacity). Therefore, the data inputs most useful in generating high accuracy models are not likely to be available on all devices. As an example

using a snapshot of current technology, classifiers distinguishing indoor vs. outdoor locations are built using data features from GPS and WiFi sensors [72]. However, GPS and Wifi are integrated into only a relatively small percentage of cell phones on the market today. This heterogeneity often requires users of less capable devices to settle for less accurate models based on other available data features. Figure 3.1(a) illustrates the result of this situation, showing the experimental performance of a significant places classifier (see Section 3.4 for implementation and performance details) for four device capability classes (CC): CC1 is Bluetooth only, CC2 is Bluetooth and WiFi, CC3 is Bluetooth and GPS, and CC4 is Bluetooth, WiFi and GPS. Perhaps unsurprisingly, the accuracy of location recognition increases as the sensor inputs from more capable cell phones are used to generate better models. These observations motivate and inspire our proposed *opportunistic feature vector merging* approach with which we seek to push the model performance possible with lower tier devices (e.g., CC1) towards that possible with higher tier devices (e.g., CC4). With feature vector merging, data features from more capable devices are borrowed and merged with data features natively available from a less capable device in the model building stage, allowing the less capable device to generate a higher accuracy model. This borrowing is facilitated by opportunistic interaction (though not necessarily communication), both direct and indirect, between a less capable device and a more capable device in situ. As an example of direct interaction, as two cell phone users follow their daily routines, the cell phone without GPS can borrow GPS data features from the cell phone with GPS as an input to its indoor/outdoor model. An indoor/outdoor model based on GPS feature instances borrowed over a period of time may also be built. In the indirect interaction case, both devices collect data samples according to their respective capabilities. Subsequently, centralized matching between commonly collected features (i.e., not GPS) may provide for a binding between the feature vector collected by the phone without GPS and the GPS features collected by the GPS-equipped phone. The GPS features can then essentially be borrowed via this binding.

Even when devices provide an appropriate set of data features to build accurate models, users may be required to gather a large set of training data (perhaps manually labeling it) before applications using the model outputs work at their peak level. The inconvenience in both the labeling of training data and the time required for model training to complete may act as disincentives to the broad-scale adoption of new people-centric applications [72]. One approach to reduce model training time and effort is to support the sharing of labeled training data among users. Sharing



(a)



(b)

Figure 3.1: Classifier performance relative to varying device capabilities and the size of the training data used. In (a), accuracy is plotted for various capability classes (CC): CC1 is Bluetooth only, CC2 is Bluetooth and WiFi, CC3 is Bluetooth and GPS, and CC4 is Bluetooth, WiFi and GPS. In (b), accuracy is plotted against the training set size.

training data has the effect of reducing the per-user training time and labeling effort when building the necessary collection of training data, but is also likely to reduce the accuracy of the resulting models. This is especially true in people-centric sensing systems based on common mobile devices like cell phones. In this context, sensor data features are often limited by the non-ideal set of sensors embedded or interfaced to the cell phones, and also the quality of the training process is difficult to control. Therefore, models in this domain are often more tightly bound to the individual in order to achieve higher accuracy. Consider Figure 3.1(b), which shows the classification accuracy versus training set size for our significant places classifier. The dashed line curve reflects the accuracy of a model built by merging experimental training data from all participants (see Section 3.4 for details). The solid line curve in Figure 3.1(b) shows the average accuracy of a collection of models, built on a per-user basis

using only data sourced from each respective participant. For a given value A on the x-axis, for the per-user models, each of the N users provides A instances, while for the global model each user contributes roughly A/N instances. The quantity of per-user training data required in building the global model is low since model training cost is amortized over all the users in the system. However, the accuracy is also consistently low due to the aforementioned problems with global training data sharing in people-centric sensing systems. With our proposed *social-network-driven sharing*, we provide a hybrid approach that builds models based on training data shared within social circles, within which we conjecture group vocabularies and other commonalities lead to more consistently labeled training data and a higher model accuracy, while still reducing the quantity of per user training data required.

The contributions of the chapter are: (i) we are the first to propose opportunistic merging of feature vectors between devices to improve model accuracy on lower capability devices; (ii) we propose the sharing of training data and models between devices by leveraging the social relationships between their users; and (iii) we implement and test these two complementary techniques in the context of "significant places" [115] [15] [45], a people-centric service targeting sensor-equipped mobile devices.

3.2 Related Work

The problem of acquiring suitably labeled training data to build classification models is well recognized, and is addressed in the literature in a number of ways. To the best of our knowledge, there is no existing research targeting feature sharing through opportunistic interaction. This may be due to the fact that feature sharing may add uncertainty to the system and is thus a counter-intuitive approach to improving model accuracy. Opportunistic sharing of data features and models can be viewed as a special case of opportunistic data exchange more generally. As such, sensor fusion in ephemeral proximity-based networks is related, though neither communication within socially connected groups nor the constraints and advantages of sharing to enhance classification accuracy are treated in the general case. Sharing training sets from one user's model to improve the performance of another can be thought of as co-training [117].

The Tapestry system [42] uses a collaborative approach to perform document filtering (e.g., email) based on the reactions/responses of others. The authors of [27] propose what they term collaborative machine learning, which unifies collaborative

filtering and content-based filtering. The approach considers both the user's data content, as well as attributes and descriptors, to gain a better idea of the similarities among users, providing better accuracy for document retrieval and recommendation applications. A similar sharing concept is explored in [95], where the authors propose a method for recommendation sharing based on statistical correlations in users' data sets (e.g., music artist playlist). While these approaches enable sharing of what can be considered model training data or classification models, the sharing ignores social group connections. We conjecture our social-network-driven sharing proposal can be integrated into these systems to improve performance (e.g., in Tapestry, only considering annotations created by members of the same social group). Using social connections to guide sharing can be thought of as semi-supervised learning [117].

There are a number of research papers contributing to various aspects of "significant places" applications, including significance learning [62], location clustering [115], and location prediction [15]. We use significant places, representative of emerging applications using sensor-enhanced inferences and targeting mobile devices, to demonstrate the usefulness of our techniques of opportunistic feature vector merging and social-network-driven training data sharing.

3.3 Proposed Techniques

At a high level, a standard approach to building models involves first sensing available data, extracting and labeling sensed data features that accurately describe states, and then finding a classification technique that provides high accuracy and high confidence classification. In terms of model usage, first the available data is sensed, the necessary features are extracted and fed into the model without labeling, and then the model outputs the inferred label. In Figure 3.3, we represent these two processes pictorially, and include the stages in each process where our proposed techniques hook in (encircled with dashed/dotted lines). As indicated in the diagram, feature merging and social-based training data and model sharing are complementary techniques that can be composed in both the model learning and usage processes to improve performance. In the following, we discuss in more detail a number of design and implementation challenges, providing a roadmap for future work needed to realize the full potential of our approaches. We begin to address these challenges in this work.

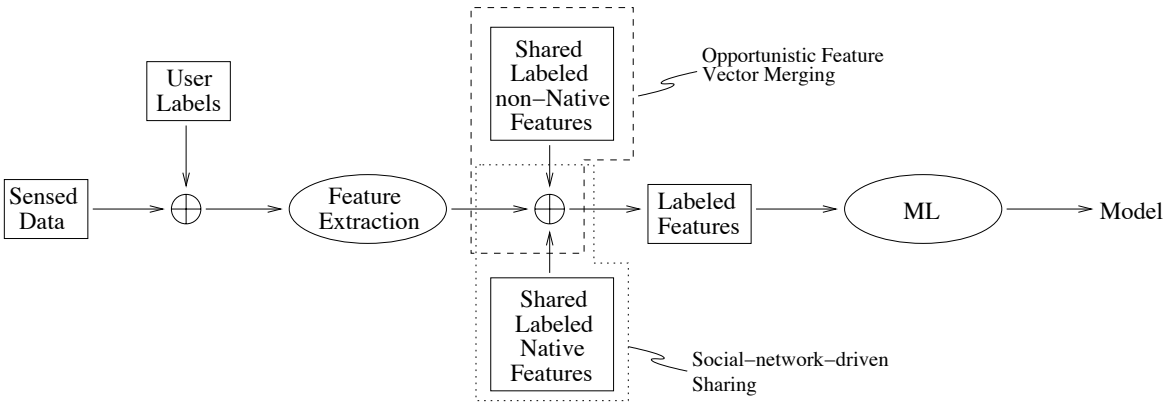


Figure 3.2: Typical model learning and usage processes, and how opportunistic feature vector merging and social-network-driving sharing hook into these. In the diagrams, the circumscribed "plus" symbols represent a merging of information (e.g., labeled features). Actions are enclosed ellipses, while objects are enclosed in rectangles.

3.3.1 Opportunistic Feature Vector Merging

With opportunistic feature vector merging, we aim to leverage opportunistic interactions (both direct and indirect) between devices with different capabilities to improve the model accuracy achievable on less capable devices. Less capable devices borrow from more capable devices features that allow for the generation and subsequent use of more accurate models than those possible to generate from only natively available data features. Here, the capability of the device can be thought of in terms of sensor configuration, available memory, and CPU/DSP characteristics. Thus, as in the example given in Section 3.1, opportunistic feature merging can provide desirable vector elements (e.g., those derived from GPS and WiFi data) that are not available natively due to the sensor configuration. Secondly, merging can provide additional data features of native types that may be needed, for example, when the device is not capable of storing a time series of the required size. Finally, opportunistic feature merging can be used to share features extracted from external data that are also available natively, but can not be calculated on the device due to device limitations (e.g., a computationally intensive FFT of microphone data cannot run on a CPU-limited device even though the device has the microphone). A number of questions arise when considering a system design that uses opportunistic feature vector merging, which we explore in the following subsections.

Determining what features are sharable.

Given the mobile devices available on the market today, the following hardware sensors are available in at least a subset of devices: camera, microphone, accelerometer, 802.11 radio, Bluetooth radio, GPS receiver, cellular radio. The raw sensor data from each of these sources can be processed in many ways, alone and in combination, to extract features useful for model building. However, not all of the features are equally sharable for opportunistic feature merging. For example, two co-located devices, one with a GPS receiver and with an 802.11 radio can likely exchange features from these sensors for mutual benefit. On the other hand, data mined from a user's calendar on one device may not be of much use on another user's device, and may even result in a worse model for the borrowing device. Similarly, raw samples from light sensors separated by even a very small distance by have very different values due shadow patterns, and may not be amenable to sharing. However, it may be useful to share temperature samples even at longer distances since temperature gradients tend to be shallower. While determining which particular features are beneficial to share or not likely depends on the classifier (e.g., how susceptible is the output to inaccuracy in the input), a reasonable guideline is to only share features that are not highly person, device, or location specific. Even if these contribute to a better classifier on their native device, they are unlikely to do so on another user's device.

What are the feature sharing mechanisms.

In Section 3.1, we introduce two types of opportunistic feature merging, depending on whether the interaction is direct or indirect. The feature sharing mechanism for each variant is slightly different. For direct interaction, devices periodically broadcast their available data sources (e.g., hardware sensors) via an available short range radio interface. Advertising only the data sources is preferable to advertising the entire feature set in terms of efficiency, since there are likely many possible features per data source. Additionally, only those data sources that are likely to be sharable (as discussed previously) should be advertised to reduce unproductive feature sharing. Devices that are interested in borrowing reply with a request for all the features available for a given (set of) data source(s). Requesting only the features, rather than all the raw data, saves on communication energy spent by both the lender and the borrower. With direct interaction, models can be used in a distributed way on each mobile device. Over time, it is also possible for a device to collect enough shared feature instances to build models based on shared features, potentially allowing for

infrastructureless bootstrapping of the system.

In contrast, feature merging via indirect interaction uses a centralized approach, requiring no direct device peer interaction. All devices collect samples, extract features, and generate models to the best of their respective abilities in situ. Subsequently, when each device transfers its training data/features to a dedicated server, the merging process looks for evidence in the features provided by all users that two or more devices were sensing the same location or event. If so, then these devices are able to share features to generate better models. Indirect sharing is helpful if two devices are co-located but can not communicate locally due, for example, to radio incompatibility. Indirect sharing also allows devices that sense the same event/phenomenon but are never co-located to share data, if the sensed event/phenomenon is relatively constant in the time between the respective devices' visits. Finally, indirect sharing potentially saves on communications costs over direct sharing since no local data exchange is necessary. For example, consider two devices that each have a GPS receiver, but only one has a CO_2 sensor. In this case, the merging process can identify through matching GPS readings that the devices were in roughly the same place at the same time. Then the device without the CO_2 sensor can borrow the CO_2 readings and incorporate them into its training data to generate improved models.

What to do when shared features are not available.

One drawback to building models requiring borrowed features is that there is no guarantee a device will be on hand to share the required features when the model is to be used. We address this with two approaches. First, each device generates a collection of models, each relying on different sets of available features. The device uses the model that has the best expected performance (i.e., w.r.t its confusion matrix) given the features available at the time of classification. In the worst case, this will be the model learned only from device-native sources. Second, we build models using algorithms that are more resilient to missing or noisy elements of the feature vector. For example, the KNN imputation method performs better relative to the comparison technique of the LNN classifier [11].

Privacy concerns in sharing.

Opportunistic feature sharing potentially leaks personally sensitive information (e.g., location trace). One option is to provide the user with the ability to configure the type of data that is sharable, and with whom. Another option is to share features without

including any identifiers in the packet payload. However, for direct sharing the MAC address of the short range radio used to share can be logged. Use of disposable MAC addresses is possible [43], but this may limit functionality for certain PHY/MAC technologies. Providing truly anonymous data exchange for ad hoc mobile devices is a focus of ongoing research in the community [29], but is outside the scope of this thesis.

3.3.2 Social-network-driven model and training data sharing

With social-network-driven training data sharing and model sharing, we aim to leverage social connections between device users to reduce the amount of time and effort an average user must expend to train her models while maintaining reasonable model accuracy. These social connections may be short-lived or persistent, and include connections based on proximity, professional groupings, family, friends, people sharing common interests (e.g., tango class), and many others. A number of techniques for mining social graphs from various information sources exist, but a review of this literature is out of scope. In the following, we discuss training data sharing, deferring treatment of model sharing to a later section. As discussed in Section 3.1, sharing training data generally has the effect of reducing the time and effort of training, but has the undesirable side effect of reducing the accuracy of models generated with this mixed data. Features that have good discriminative power within particular population subgroups, lose effectiveness within larger groups. We propose to allow sharing only within social circles to moderate this reduction in accuracy, while still reducing training time and effort. In the following, we describe a number of challenges to social-network-driven sharing, and discuss the motivation of model sharing between members of the same social group.

Exploiting social connections.

Previous work [26] [33] suggests ways to mine sensor-based and other data to infer social graphs where the vertices are people or groups and the edges are relationships. Assuming known social graphs, we construct models with training data sourced on the basis of the strength of social connections (edges in the social graph) between the intended target of the model (e.g., the device user) and others. A lower bound on the strength of connection between two users may apply such that sharing does not occur below this threshold. We expect people who are members of the same social groups (including combinations of cultural, workplace, social, or family groups) will have

similar background or other context that translates into similarity in label definitions (i.e., what classes are important and what are the appropriate labels). By exploiting awareness of the social connections between people we build a training set sourced by a variety of people that still produces a model for a particular individual (or group) that approximates the performance of a model built solely from training data sourced from this individual (or group) in terms of both classification accuracy and the understandability of labels.

A number of interwoven social graphs are likely to apply to a given set of individuals. The nature of the inference problem (i.e., the application, or learning technique) may determine which social graph to use when considering which training data to import from other users. In the context of our running example of significant place classification, if a user may provide free-form labels (e.g., colloquial labels for locations), it may be appropriate to incorporate labeled instances from other nodes in her social network with whom she is frequently physically located, under the supposition that a location-specific vocabulary is likely in use (e.g., workplace vernacular, regional dialect). On the other hand, individuals to whom one is extremely close socially (e.g., a girlfriend), may be of less use in sharing location-specific vocabulary if they are frequently physically distant. Similarly, labeling of certain activities or social settings may be more culturally and demographically driven.

Quality and consistency issues.

A number of challenges arise related to the quality and consistency of shared data instances.

First, the quality of the training instances may vary from user to user due to the care taken when the training data was gathered, the training methods used, and the training environment (e.g., data collected under non-typical circumstances can lead to a model that does not perform well in general). Challenges in repairing ill-labeled data aside, it is difficult even to determine which instances are lower quality. This is especially difficult when the pool of available labels is small and statistical techniques for instance anomaly detection are not applicable. Because of this, importing lower quality training data can pollute one's natively collected data, leading to poorer model performance.

When free-form labeling is used, opinions may vary among users on the proper size of label set, the feature support of each label, and the label itself. A related complication is that lexicographically identical labels may mean different things to different people and different labels may mean the same thing to different people.

One way to address these issues is to apply structure to the labeling stage such that a fixed set of valid labels, each with a provided definition, is imposed on all users. However, this approach restricts the classifications problems that can be solved, and may result in a model that, though accurate, gives labels that are not well understood by a given user.

Designing models robust to mixed source data.

Given the lack of flexibility of structured labeling, we support free-form labeling. While sharing within social circles mitigates labeling consistency issues to some extent, the process of learning models must still be robust to them. Incorporating contradictory instances, where the same class of features is given two or more different labels (by multiple users), leads to a situation where the same class of feature vectors will be mistakenly fragmented into multiple labels. (The impact of this fragmentation is somewhat problem-specific, since a classifier that seeks only to differentiate between logical classes of might perform well even with fragmented features.) We use an unsupervised clustering approach to detecting and correcting this fragmentation in our significant places implementation discussed in Section 3.4. Instances can then be appropriately grouped regardless of their label, with the introduction of some error due to imperfect grouping. After clustering, a normative label may be applied for consistency.

Social-group-based Model sharing.

In addition to sharing training data, the models themselves are also candidates for sharing. The trigger for borrowing models would be noticing that the performance (e.g., recall, precision) was better in the model of a fellow social group member than in yours for the same feature vector. In this case, either the user's device can check neighboring devices in situ to if they have an appropriate model with better performance (e.g., via an advertise-request-response protocol), or the model sharing can be done in a centralized way on a dedicated server. The rationale for model borrowing between members of a social group in particular is that even though the models may have been learned based on training data labeled by your buddy, your buddy's labels are likely to make sense to you because of your shared membership in the social group. Elements of shared models that may be particularly helpful in improving a user's locally generated model can be permanently incorporated by importing the appropriate training data and relearning the local model. This is beneficial in the

online case since performance can be maintained even when the neighbor with the better model is not nearby, and in the offline case it reduces unnecessary processing.

3.4 Evaluation

To evaluate the impact of opportunistic feature vector merging and social-network-driven data and model sharing on a real people-centric application, we implement a version of the "significant places" classifier (e.g., [50] [33]). We use this as a vehicle to demonstrate the application of our techniques. In the following, we describe the implementation and focus of our variant of significant places, and the experimental data collection methodology, followed by selected performance results.

3.4.1 Significant Places

A frequently examined classification problem in the literature is that of taking location traces of a user and distilling them into a sequence of visits to places that are significant to her (e.g., home, work, gym). This is used by applications that present historical summaries of the user's daily life [15], or even to determine when a person has taken a wrong turn heading toward home [83]. A generic significant places classifier may be thought of in terms of three main phases. In the first phase, various data features (e.g., visitation frequency and dwell time) of a user location trace are extracted from the raw data and analyzed to identify locations and infer whether they are significant to the user. In the second stage, the significant places are labeled, either by mapping the location feature vector to a set of system-provided labels or by manual prompting of the user to allow for personalized labels. In the third phase, the classifier is run to see how accurately the system can recognize that a user has entered a significant place. A number of proposals (e.g., [115] [50]) exist addressing the first phase of learning models to infer significance. As significance inferencing is orthogonal to our techniques, in our implementation we simplify the first two phases and have the user manually label instances of location feature vectors as significant or not (c.f. the collection methodology in Section 3.4.2). Based on these labeled instances, we then evaluate the impact of our merging and sharing techniques on the accuracy and label understandability of models built to recognize the labeled significant places.

3.4.2 Data Collection Methodology

As the sensing, processing and display capabilities of cell phones increase, cell phones provide a unique chance for researchers to understand the real mobile user behaviour and to provide true in situ mobile services. To gather user-labeled significant place instances we use Nokia N80 and N95 smart phones. Both models feature Bluetooth and an 802.11g WiFi interface. The N95 also comes with an integrated GPS receiver. To facilitate user labeling of significant places, we implement and install a PyS60 (Python for Symbian S60) client on each cell phone. The client provides two fundamental services: user labeling and daily trace recording, and sensor sampling. For each significant place, the user enters a new label (or selects a previously entered one). With a button click, the user indicates when she enters and leaves the selected significant place. The client records the label, and the enter and leave times for each significant location visit. From these entries, the client generates a significant location trace for each user. The user is able to review and edit the daily trace to verify its correctness. The sensing daemon runs in the background to sample from the Bluetooth, WiFi and GPS, if available. We use an inter-sampling interval of approximately three minutes, which gives an average battery life of more than 6 hours. The sampling duration is lasts between 30 and 60 seconds, depending on how long the function call to scan the Bluetooth neighborhood takes to return. The following data are captured: GPS - latitude, longitude, altitude, accuracy, time, speed, number of satellites; WiFi - beacon interval, security mode, SSID, BSSID, signal strength; Bluetooth - address, device name, service type.

3.4.3 Data Analysis Methodology

The inputs to the models we construct are based on a feature vector formed from three types of elements; location, time and social context. Clock, GPS, WiFi and Bluetooth data give rise to the following features, which are also further processed to generate averages and variances. From the clock, we extract day/night, 3-hour block, the duration of visitation, weekday/weekend and, business/after hours. From WiFi, we extract the absence or presence of access points (APs) identified by their MAC addresses, the relative RSSI order among the visible APs, the individual and aggregate RSSI, and other AP statistics that have been previously used to distinguish geographic locations [45] [55]. From social context, we seek to capture the social characteristics of the location. We extract the number of Bluetooth-toting people in the area (assuming one device per person). This is used in concert with a list

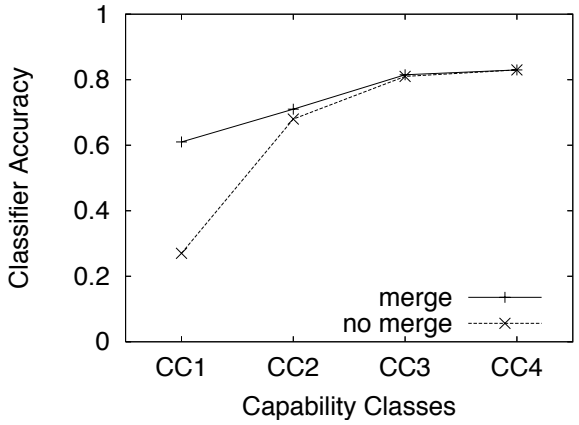
of the people with whom the individual has social connections (e.g., from Facebook or other social networking sites). Use of Bluetooth and WiFi features allows us to distinguish between adjacent locations that may have very similar GPS features. We use the Weka machine learning workbench [109] for our analysis, specifically the default configuration of the bagging algorithm applied to the decision tree module, REPTree. All models are trained on a randomly selected 50% of the data set available for it. In the following, we describe initial performance results achieved with models based on a proof-of-concept implementation of our sharing and merging ideas. We leave a deeper investigation of the design space for later work.

3.4.4 Performance Results

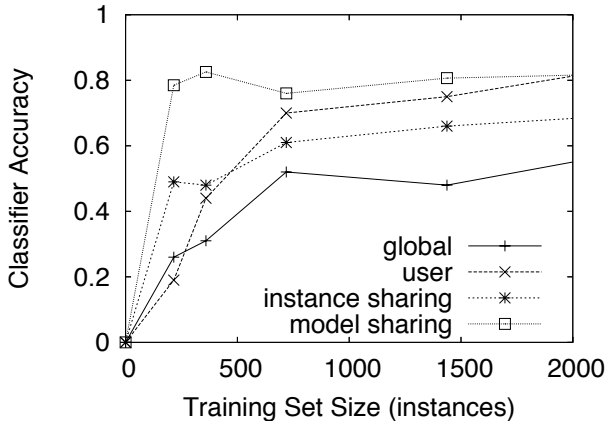
In an experiment run over 12 days, data we collect from 13 phone users (four Nokia N95 and nine N80 cell phones) using the collection methodology outlined above comprises 14375 labeled instances of 62 uniquely labeled locations. We run post-collection validation via manual checking and participant interviews to verify the integrity of the data set. All phone users are members of, or are socially connected to the Computer Science Department at Dartmouth College. Participant ages range from 24 to 49; one user of the 13 is female. We also gather results from a survey (described subsequently) that includes the 13 phone users and an additional 8 survey-only participants. These latter fall in the same aforementioned age range and have the same departmental connection; one of the additional 8 is female. We present results demonstrating the potential impact of the opportunistic feature vector merging and social-network-driven sharing.

Feature Vector Merging Performance.

We generate models, "merge" and "no merge", from experimental data, and examine the impact of performing direct sharing of features based on different device capabilities. Sharing of features is done on the basis of Bluetooth connectivity. Whenever two devices in the experiment detect each other in their Bluetooth neighborhood then feature sharing is enabled. An exchange of feature vector elements occurs when possible, giving participant nodes a richer feature vector they would otherwise have based on native sensors. Although all Nokia N80 and N95 phones have WiFi and all N95 phones have GPS, to emulate four distinct capability classes of devices for some devices in the experiment the WiFi on the N80 phones and the GPS on the N95 phones is ignored as needed to support allowing four different classes to be emulated.



(a) Feature merging.



(b) Social sharing.

Figure 3.3: Performance Plots

We build the models as follows: a single model is generated for each user during the evaluation. This model is trained using all the feature vectors available, even those that are intermittently available via sharing. This results in numerous feature vector instances with missing elements, since sharing is not continuously available. We do not explicitly handle missing data within execution of our model (e.g., using a model swapping technique), but instead use a machine learning technique (bagging) innately robust to the missing data. Models are built on a per-user basis using training data specific to the user and user's device, and based on his or her own opportunities for merging.

Figure 3.3(a) shows a comparison of these two models. It reports the average classification accuracy for each of the per-user models generated. Accuracy is plotted against the phone capability class. In each of these classes the performance is reported for all phones being limited to this operating level or lower. The plot shows that

with feature sharing, we can always gain an advantage in model accuracy, except for capability class CC4 devices since those already have all the sensors natively. In our campus environment and predominantly indoor significant places, WiFi is the most powerful feature to share to improve accuracy, as indicated in the large increase between CC1 (Bluetooth only) and CC2 (Bluetooth and WiFi). In environments where WiFi features are less available, we expect shared GPS-based feature elements to be the most helpful.

Social-network-driven Sharing Performance.

We generate four models from experimental data, including two that incorporate sharing to support model generation, "instance sharing" and "model sharing", and two that do not, "global" and "user". With these models, we investigate the impact of sharing on classification accuracy with respect to the amount of training data provided by each user. In all cases, models are trained using a randomly selected 50% of the data, with the balance used for performance testing. The device population comprises the following mix of capability classes: 5 Bluetooth only; 4 Bluetooth and WiFi; and 4 Bluetooth, WiFi and GPS.

As discussed in the Introduction, the "global" model is generated by pooling training data from all participants, with each user contributing roughly the same amount to the pool. The "user" model is generated on a per-user basis using only training data sourced from the user herself.

The "model sharing" approach generates per-user models as in the "user" approach, but then multiple models are tested before settling on a label output for a particular instance. The decision to apply another model or settle on the current result is based on estimated accuracy for the generated label. The choice of whose per-user models to choose for a given classification task is driven by social connections between users, prioritizing social connections that are logically related to the classification task. In so doing, models are applied according to a hierarchy of social groupings. Users' models within a group are ranked arbitrarily in our implementation, but the strength of personal social ties within a group can also be considered when deciding the order of model application. The application of models terminates either when a confidence threshold is reached (to improve classification accuracy), or if a certain maximum number of models is applied (to limit overhead). Lastly, with "model sharing", we always test the "global" model (global sharing) as well, and the result with the highest confidence among all the tested models becomes the final output label.

With "instance sharing", per-user models are built, but for a given user the training data is sourced from the user and from people within the user's social networks. As with "model sharing", a social group hierarchically is constructed considering the purpose of the classifier and the groups' potential impact in this regard, and intra-group ranking is also handled in the same way. In "instance sharing", training data instances are accumulated iteratively, considering one user per step, until the overall required number of training instances L is assembled. At each step i, the user is tapped to provide up to L/i instances. The goal, if K steps are taken to accumulate L instances, is to have each user provide L/K instances. At any point, if a user can not contribute the desired L/i instances, randomly chosen instances from the global pool are chosen, but are removed from the overall required L if they are no longer needed as filler.

The social groups present in our experimental user population and used for the sharing-based models are: "students", enrolled students at any college; "Dartmouth", enrolled students at Dartmouth College; "batch", grouped according to the year arriving at Dartmouth; "founders", founding members of SensorLab that have worked together since the inception of our research group; "SensorLab", all members of the SensorLab research group; "CMC", all members of the CMC Lab research group; "Chinese", have a strong social tie including a daily lunch group; "Facebook", social connections as defined within Facebook; "basketball", participants in a local summer basketball club; "town", those with a common town of residence; "European", those with a European origin; "non-U.S.", those with any non-U.S. origin. We order these groups according to the degree to which we expect them to improve our significant places classifier. Participants in the study are members of multiple different social groups.

The rationales for a few of the ranking decisions are as follows. We rank "Facebook" above "Dartmouth" since we expect Facebook friends to use common names for specific locations more so than does the general pool of Dartmouth students. The same logic leads us to rank the "members" ahead of "students". Group rankings may fluctuate seasonally. For example, the "basketball" group meets regularly, but only during the summer - familiarity (e.g., common experience, shared stories, shared vocabulary) decays over the rest of the year. Conversely, the "Chinese" group meets every day at noon for lunch, so the social ties remain strong throughout the year.

Figure 3.3(b) shows the classification accuracy versus training set size for each type of model. For a given value A on the x-axis, for the per-user models, each of the N users provided A instances, while for the global model each user contributes

Model	User (avg)	Global	Feature Merging	Model Sharing	Instance Sharing
TPR	0.598	0.383	0.617	0.721	0.389
FPR	0.563	0.349	0.565	0.652	0.304
Precision	0.691	0.496	0.712	0.807	0.544
Recall	0.598	0.383	0.617	0.721	0.389

Table 3.1: Classifier statistics.

roughly A/N instances. For the sharing-based models, we assign equal weights to all social groups; the group hierarchy is flat. Each of the M users involved (i.e., through common social group membership) contributes A/M instances. The figure demonstrates the advantage of sharing only within social groups rather than globally as the model accuracy curves for both instance sharing and model sharing are always above that of the global model. Additionally, we see the advantage in terms of model learning time (i.e., required training data set size) that both instance and model sharing provide. We find that social-based model sharing achieves a higher maximum accuracy than training instance sharing for our data set. As expected, the per-user model outperforms instance sharing as the amount of available training data becomes large enough.

Classifier Performance Details

Figure 3.3 shows the sensitivity of the model accuracy to both the richness of the feature vector and the availability of training data, accuracy alone does not provide the full picture of the model performance. In Table 3.1, we present additional performance details (true positives rate (TPR), false positives rate (FPR), precision and recall) for each of the classifiers we use (i.e., the average performance across all classes). For the results shown here, the training set size is fixed at 719 instances.

3.4.5 Survey Results

To understand the impact of social-network-driven model and instance sharing on the understandability and appropriateness of the model output labels, we survey 20 participants concerning the outputs of the sharing-based models used in the experiments described in Section 3.4.4. In this survey, we focus on determining the participants' depth of understanding of shared labels, and their feeling of the appropriateness of these labels when shared socially.

In Table 3.2, we report statistics on the level of comprehension people from different social groups have of labels produced by members of their own versus other

Group	Strong		Weak		None	
	"SensorLab" Labels	"CMC" Labels	"SensorLab" Labels	"CMC" Labels	"SensorLab" Labels	"CMC" Labels
"SensorLab"	0.75	0.32	0.09	0.15	0.16	0.53
"CMC"	0.40	0.55	0.05	0.03	0.55	0.43
All	0.48	0.34	0.14	0.31	0.38	0.49

Table 3.2: The level of comprehension people from different social groups have of labels produced by members of their own or other social groups. Members of the same social group share a better comprehension of each other's labels on average.

social groups. Survey participants are asked questions to determine their level of understanding regarding 8 different labels. For each label, users are asked to identify to where they think a label refers when given the label provider's name and the label itself. The understanding is categorized as "strong", "weak", or "none" depending on how accurately the label is positioned on a map; "strong" if the exact location is indicated, "weak" if a location in the vicinity is indicated, and "none" otherwise. Label providers are not asked about their own labels.

Comprehension levels are shown in Table 3.2 for the dominant social groups ("SensorLab" and "CMC") that produced the most labels in our experiments. Members of the same social group share a better comprehension of each other's labels on average, compared both with members of the other group and the average population. For example, on average members of "SensorLab" stated they had a "strong" comprehension of 75% of labels generated by members of their own group, but no comprehension of 53% of the labels generated by "CMC" members. These results indicate that a model based on global sharing is likely to perform poorly in terms of understandability, in addition to accuracy (Figure 3.3(b)), underscoring the importance of social-based sharing.

To determine the statistical significance of the results in Table 3.2, we run a χ^2 test with a threshold of 0.05, and calculate $\chi^2_\alpha = 5.9915$. First, we test the null hypothesis that comprehension of the labels provided by the "SensorLab" group is independent of group membership. The null hypothesis rejected with $Q = 14.401$. In the analogous test for the comprehension of labels provided by "CMC" members, we calculate $Q = 6.3068$, again rejecting the null hypothesis, concluding that the "CMC" members' better understanding (relative to that of "SensorLab" members) of labels provided by fellow members is statistically significant. These results give statistical credence to the notion of social-group-based sharing.

Table 3.3 presents results from the same survey on the appropriateness of labels provided by selected individuals for particular places. Given a place, survey partic-

Groups	Place:Label SensorLab lab:'Lab' ("SensorLab")	CMC lab:'Lab' ("CMC")	Orient Restaurant:'Ori' ("Chinese")
"SensorLab"	2.10	1.00	0.83
"CMC"	1.25	1.75	1.25
"Chinese"	0.80	1.20	1.20

Table 3.3: The level of appropriateness of selected labels as viewed by different social groups. Social connections can strongly impact the perceived appropriateness of a label, an important motivation for social-based instance/model sharing.

ipants rate four possible labels (each taken from labels generated by the 13 phone experiment participants) to describe the place on a scale from 0 to 4 (0 means "not appropriate"). Table 3.3 shows selected results for three (place,label) combinations. Generally, the table shows that the perceived appropriateness of a given label can be strongly impacted by social connections, as reflected in the higher values along the diagonal. For example, at least one member of each of the two laboratory groups ("SensorLab" and "CMC") included in the user set use the label 'Lab' to refer to their respective lab. Members of each lab think this label applies most appropriately to their own lab (i.e., average rating of 2.10 and 1.75 for their own versus 1.25 and 1.00 for the other lab). "Chinese" comprises those that often go together for lunch at the Orient restaurant. The table shows that "Chinese" members are more likely than "SensorLab" members (though not more so than "CMC" members) to find the diminutive 'Ori' acceptable. The lack of distinction between between "SensorLab" and "Chinese" for this label may be due to the existing overlap in group membership. These results support the use of socially shared labels in the significant places test application.

3.5 Summary

As the sensing and computation capabilities of consumer mobile devices (e.g., smartphones) increase, the development of people-centric applications augmented with sensor inputs will also accelerate. To facilitate the wide-scale adoption of these applications, we have proposed two techniques aimed at both increasing the accuracy of classification models used by these applications, reducing the burden on the user in terms of providing labeled training data. We have demonstrated the efficacy of both opportunistic feature vector merging and social-network-driven sharing in the context of "significant places", a useful classification process for people-centric sensor-enabled applications. To the best of our knowledge we are the first to demonstrate the ability

of classification accuracy to be increased by grouping similar users together, as identified by their membership in social networks. Our results underscore the opportunity and importance of leveraging the inevitable device heterogeneity that results from the evolution of technology, and the importance of taking social relationships into consideration when sharing in support of model building.

Although these results confirm the potential for approaches that leverage communities at the same time they have a number of glaring limitations. Features are shared between mobile devices without any mechanism that validates the suitability of data to be shared. This weakness most likely did not become apparent during evaluation as features based on location estimates are fairly robust to being shared between devices over short distances. In contrast, features based on other sensor modalities (e.g., sound, camera) are far more sensitive to distance and orientation of the devices. Similarly, this chapter assumes that untrained users will provide labels that are used to build classification models but ignores the mistakes they will no doubt make in the segmentation of classes and labeling of sensor data. This assumption will not withstand use within a real-world large-scale user population. Class "pollution" from noisy labels will quickly accumulate to the point of counteracting the benefits of larger pools of labels. Finally, it is unclear as to how generalizable our techniques actually are beyond the specific application of "significant places" under which they are demonstrated.

In the next two chapters we focus solely on communities formed by users in favor of in-situ co-located devices. We develop techniques that make realistic assumptions about user behavior, for instance the error-prone nature of the labels they provide, and that can clearly generalize to support broad classes of mobile sensing applications.

Chapter 4

Exploiting Crowd-sourced Labels

4.1 Introduction

In the experiments described in Chapter 3 everyday users provide labels from which individually or collaboratively trained classifiers are built. The premise of incorporating the mainstream public so directly into the learning process initially may seem unrealistic. However, millions of people now voluntarily carry more sensors and computational power than was available on specialized sensing devices [17, 25] just two years ago. Constrained learning applications for these devices already abound: dozens of iPhone applications have been released that incorporate basic human sensing and learning. Citysense [76] for example traces user movement patterns to find nightlife in San Francisco. It is not hard to envision even larger, more general systems for learning more complex trends from user data evolving from these early encouraging applications.

The availability of this data presents an opportunity to radically change the way we build computational models of human behavior. We believe large-scale learning systems will benefit greatly from the effective use of unconstrained human data. Current approaches, however, are ill-equipped to deal with such data: they typically rely on a static well-defined set of labels, or deal with semantic discrepancies. Furthermore, to improve generality and reduce individual user requirements, labels are often pooled from multiple users – a technique that only works if the labels are consistent across users. In reality it might be beneficial for classification to split a class with a specific textual label into several sub-classes or merge classes with different labels. Supervised and semi-supervised learning techniques assume a rigid closed set of labels, and unsupervised algorithms do not incorporate labels at all.

In this chapter, we propose *Community-Guided Learning (CGL)*, a novel frame-

work for learning in a dynamic human environment. Community-based models are already widely used in other domains (e.g., Wikipedia, SETI@home, Freebase), where they successfully amortize individual users' shortcomings by incorporating contributions from others. Analogously, our approach uses notions of similarity to incorporate data from multiple users in a flexible manner that neither places excessive weight on labels nor discards them entirely. More specifically, CGL uses existing unsupervised and supervised classifiers to find groupings of the input data that maximize robustness and classifier performance.

The novel contributions of this chapter are as follows:

- We propose and develop the CGL framework for learning models of human behavior from crowd-sourced sensor data.
- We demonstrate the effectiveness of similarity measures to regroup classes specified by imperfect labels from users.
- We present experimental results showing CGL's advantages over conventional learning techniques.

4.2 Related Work

There is much prior work that carefully collects and labels high-quality training data [17, 25, 48, 59] for learning models of human behavior. Unsupervised activity discovery is another active area of research where no labeled data is used during training. Both approaches need labels to be specified at some stage to perform classification. To achieve reasonable performance, the domain of discovery is often restricted to simple behaviors [101] or relies on output from low-level supervised models [49] during unsupervised learning. Semi-supervised [69, 100] and multi-instance learning approaches [98] have been proposed to deal with limited and inconsistent labels. In multi-instance learning, labels are associated with a set of data points that includes at least one positive data point but may also include points that are not from the labeled class. In all of these cases, class labels are considered to be fixed.

We are not the first to identify sensor-enabled consumer devices as an opportunity to radically alter an existing paradigm. In recent years researchers have been considering ways that mass deployments of mobile sensors (e.g., embedded in cell phones) can change people-centric data collection [10, 22, 24, 54]. This work has focused primarily on privacy, mobile device resource limitations, and data fidelity. In the previous chapter (also published in [57]) we proposed a community-based technique that groups

people based on their social clique and partitions their data accordingly to improve learning. However, the grouping is only network-based and the label domain is still fixed. Hence, the challenges we identified above remain unsolved.

4.3 Community-guided Learning

Inevitably, any large learning system will have to deal with mislabeled data for reasons ranging from simple misunderstandings to malicious behavior. Naïvely accepting user-provided labels can undermine learning, as these may be overly broad or narrow and either way, will rarely be textually equal. For example, distinguishable (from the point of view of the sensor data) classes may be given a single label like "work". In such cases, modeling the distinct subclasses separately may lead to better performance. On the other hand, similar or indistinguishable classes may be given labels like "driving" and "commuting" and a joint model will perform better.

We present community-guided learning as a framework to build classifiers using inconsistently labeled sensor traces. To achieve this, CGL trains data in groups not only defined by users' labels but also by properties of the data. In essence, we account for "soft ground truth" by first performing a clustering step to refine the classes suggested by labels and then training a supervised classifier on those clusters. We list specific steps below and provide figure 4.1 for a schematic overview:

- Take as input inconsistently labeled mobile sensor data from multiple users.
- Measure *intra-class similarity* between segments that have the same label to decide whether to keep the original grouping of the data or to split dissimilar segments into sub-groups and model them as different classes.
- Train classifiers using groupings from previous step.
- Measure the *inter-class similarity* between segments that have different labels and decide whether to keep the original groupings based on the labels or to merge similar segments that have different labels and model them as one class.
- Retrain classifiers using groupings from previous steps.

In a deployed system, the above process can be applied iteratively using new contributions from the community to update classifiers.

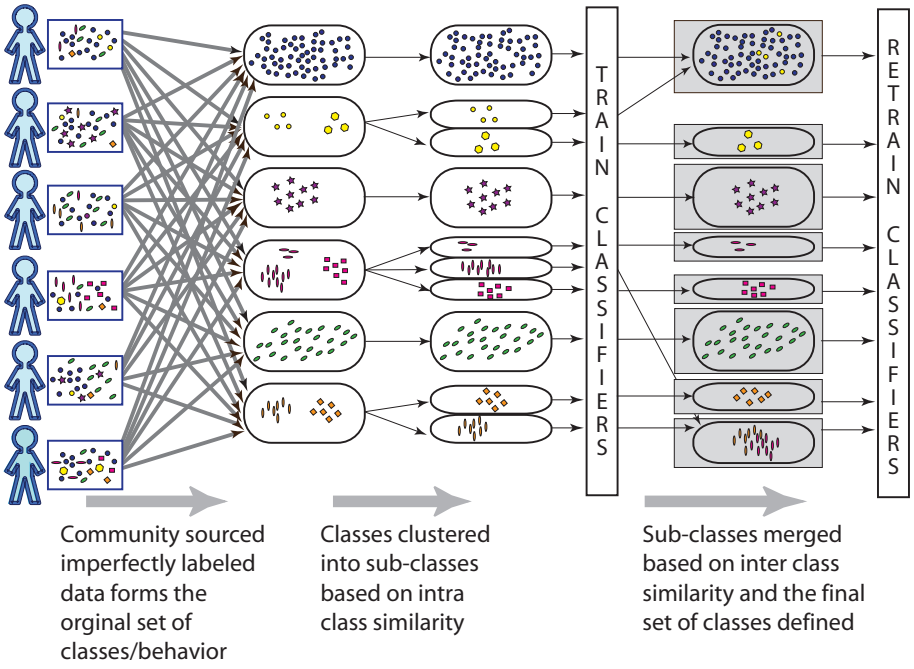


Figure 4.1: The main steps of CGL. Segments are first grouped according to user-provided labels. The class groupings are redefined based on inter- and intra-class similarity measures. Classifiers are built based on the resulting groupings.

4.3.1 Splitting and Merging User-Defined Classes

A primary goal in CGL is to preserve the user-provided labelings as these are what they expect. But users cannot be expected to understand *how* to label their data such that the classifiers perform reliably. Instead, the reasoning system, guided by the labels and the input data, should make the final decision on how to re-partition that data to achieve good performance.

To that end, CGL first uses clustering to determine whether distinguishable classes have been assigned the same label. If so, the data is split into logical subclasses. Next, if indistinguishable classes have been assigned different labels, CGL unifies them. For example, data labeled by a user as 'driving' may include both urban and stationary segments – these different types of driving data will likely have different sensor signatures. This example should be clustered and split into two classes. However, in the merge phase, CGL may find the stationary subclass to be indistinguishable from other similar data, for instance waiting at a drive-thru, and will thus merge and model it as a single class. In the following section, we describe the two types of similarity metrics we use to split and merge the contributed segments and how we utilize these similarity measurements in the learning process.

Defining Similarity.

CGL uses intra-class and inter-class similarity measures, as described below:

Intra-class similarity. This is used during the splitting stage. The goal is to find the underlying groupings in data that share a label. This is purely data-driven and is effectively a clustering operation. Many clustering algorithms exist, but we choose Euclidean k-means [18] for its simplicity, but make no assumptions about the clustering used. Unfortunately, there is no agreed-upon solution for choosing the number of clusters k. We estimate k by evaluating the objective function for varying values of k – typically the objective function decreases rapidly when k is smaller than the correct number of clusters and flattens when k is larger. Thus, the location of an *elbow* in the objective function is a good choice for k [70].

If a class is split into two or more sub-classes, CGL treats them as independent classes for training, but associates them with the same user-visible label. This is purely for the user's benefit, and a different front-end could instead ask the user to refine the subcluster labels.

Inter-class similarity. Unlike the purely data driven intra-class similarity, we measure inter-class similarity by comparing the output of two different classifiers on the same input. The aim is to identify classes that may be labeled differently but belong to or is better modeled as a single category. For example, 'talking' and 'conversation' classes may both represent scenarios where a person is speaking. If the data belonging to two or more labels represent the same underlying class then these classifiers are likely to "agree" and are potential candidates for merging. Let us assume two classifiers, C_A and C_B. On a give test segment S, we compute C_A's performance on B and C_B's performance of A – let us assume they are based on confusion matrices $conf_{AB}$ and $conf_{BA}$ respectively. The agreement between the classifiers is defined to be the f-score computed from the combined confusion matrix $conf_{AB+BA} = conf_{AB} + conf_{BA}$.

The intuition behind the similarity score is as follows: if two classes A and B are the same or very similar, C_A will perform well on B's data and C_B will perform well on A's data and the combined f-score will also be high. If the classes are dissimilar, C_A will perform poorly on B's data and vice verse, resulting in a low combined f-score.

If the *inter-class similarity* score is greater than a experimentally determined threshold, the classes C_A and C_B are merged and used to train a unified classifier C_{AB}. We set this threshold such that merging operates conservatively, causing merging only to be applied at levels of similarity that caused high subsequent f-scores during exploratory experiments. If classes are merged, the system associates the union of

the labels to the newly defined class. But similar to the split case, users could be prompted to confirm whether the classes should be merged and whether they want to re-label.

4.3.2 Training Classifiers

After classes are redefined based on the similarity measures, any classification algorithm can be trained and used for recognition. Since our primary goal is to test the effectiveness of CGL, we use simple boosted decision-stump classifiers [38] in our current experiments to train binary activity classifiers. We are not tied to a specific type of classifier and can use more sophisticated models if needed. However, boosted decision stumps have been successfully used in a variety of classification [103] tasks including human activity recognition [19, 60].

For each activity A_i , we iteratively learn an ensemble of weak binary classifiers $C^i = {c_1}^i, {c_2}^i, {c_3}^i, ..., {c_M}^i$ and their associated weights ${\alpha_m}^i$ using the variation of the AdaBoost algorithm proposed by [104]. The final output is a weighted combination of the weak classifiers. The prediction of classifier C_i is:

$$C^i = sign(\sum_m {\alpha_m}^i {c_m}^i)$$

Training is done after *both* the split and merge step based on the class groupings produced by those stages.

4.4 Evaluation

In this section, we describe our dataset and the experiments we conducted to evaluate the performance of CGL and test its ability to cope with two common forms of human labeling error: (i) inconsistencies in the class definition applied by people and (ii) errors in marking class boundaries during labeling.

4.4.1 Dataset

We collected a 33-hour audio dataset of high-level activities and their associated contexts, as shown in table 4.1. We recruited five researchers to run a custom audio logger on jailbroken iPhones and instructed them to loosely but truthfully provide freeform labels of anything they chose. In the resulting dataset, we found that the

Category	Labels
Working	working, office, name of the building/room
Driving	driving, car, vehicle
Transport	bus, vehicle, cab, airplane
Eating	eating, lunch, dinner, kitchen, Dirt Cowboy, Quiznos, Boloco, Ramuntos, Five Olde, Novack
Shopping	shopping, supermarket, grocery store
Gym	gym

Table 4.1: Audio dataset

Domain	Features
Time	ZCR, RMS, low energy frame rates
Frequency	Spectral entropy, flux, centroid, bandwidth, normalized phase deviation, band energy
Cepstral	13 MFCC, DC component removed

Table 4.2: Features extracted from audio dataset

labels consisted of activities performed (e.g., driving, eating, working) or the type/-name of the places the users visited (e.g., supermarket, office, library, restaurant). A full list is shown in table 4.1.

4.4.2 Data processing and feature extraction

To calculate similarity scores and perform classification, we extract a set of features from the recorded audio. We choose acoustic features that emphasize important characteristics of the data and have been used successfully in other mobile audio sensing applications. Table 4.2 summarizes the 39 computed features, but for a detailed overview the reader should refer to Soundsense [67].

As we deal with large datasets and seek to summarize the sound of high-level activities, we chose a fairly long frame length (window of samples used to calculate features) a half-second non-overlapping frame compared to that common in speech applications.

4.4.3 CGL Stages

Here we explain the experimental procedure for each stage of CGL. According to the framework, user-contributed data is initially grouped based on user-provided labels, the data in these groups is then clustered during the intra-class similarity step. Classifiers are trained based on the clusters found with similar classes then merged together before a final training step occurs to complete the process.

Intra-class similarity measurements. We start by clustering the data using Euclidean k-means, determining k as described previously. Note that this procedure

can pick $k = 1$, and often does.

Figure 4.2 shows the result of principal component analysis (PCA) on the driving data. The figure clearly demonstrates three distinct clusters and in fact k-means also splits the driving class into three clusters but operates on the full 39-dimensional data. Table 4.3 lists the classes that had more than one cluster.

Classifier training. The clusters produced by the previous step are then used to train boosted decision stump classifiers. Assuming that a class C gets broken into subclasses by the previous step, we construct a classifier for C in the following manner:

- For each subclass, consider points within its cluster as positive examples and points outside (including other subclasses of the same class) as negative examples.
- Run 10 iterations of Adaboost on the above to get a classifier for each subclass, making sure to give the positive and negative classes equal weight.
- The output of our classifier for class C is the disjunction of the outputs of each of its subclasses' classifiers.

For example, if a class is found to have three subclasses, we train one boosted decision stump classifier for each of these. The compound classifier returns true for a given sample if (and only if) any of the three classifiers returns true on that sample.

Inter-class similarity measurements. After obtaining the set of classes and classifiers from the split stage, we determine the inter-class similarity between each pair of classes. Given two classes A and B and their respective classifiers C_A and C_B, we compute the similarity as follows:

- Run classifier C_A on B's data and classifier C_B on A's data
- Compute the confusion matrix conf_{AB} for C_A's when applied to B's data, and vice versa
- Compute the f-score of the combined confusion matrix $\text{conf}_{AB+BA} = \text{conf}_{AB} + \text{conf}_{BA}$
- The combined f-score is used as the similarity measure between A and B

The similarity matrix thus generated can be seen as a complete graph with edge weights corresponding to pairwise similarity. Raising this matrix to a power simulates

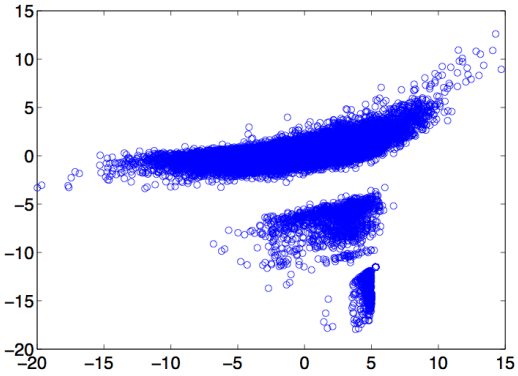


Figure 4.2: Two-dimensional PCA on driving data, showing three clear clusters.

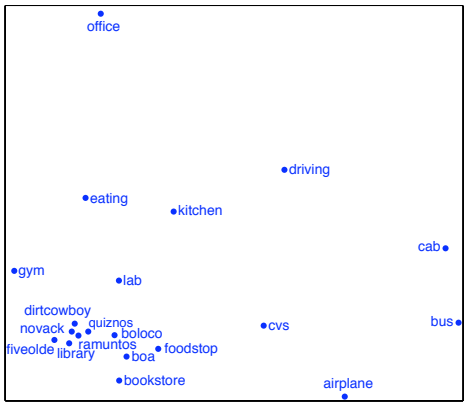


Figure 4.3: Multidimensional Scaling results for similarity between classes. MDS axes have no meaningful units.

transitivity of similarity. For example, raising it to the second power "walks" one step over all the edges, accentuating similar elements and de-emphasizing dissimilar ones.

Based on the similarity matrix, we chose a conservative threshold experimentally chosen to favour merges only for highly similar classes.

Classifier re-training. After classes are merged, classifiers for the new categories are trained using the same boosted decision stump classifier. If classes A and B get merged in the previous step, we construct a classifier C_{AB} in the following manner:

- Consider points in A *or* B as positive examples and points outside as negative examples.
- Run 10 iterations of Adaboost on the above to get the parameters for classifier C_{AB}.
- The output of the classifier on A, B, or AB is simply the classification output

Class	k	f-score		Precision		Recall	
		pre	*post*	*pre*	*post*	*pre*	*post*
Driving	3	80.4	82.3	89.4	89.2	73.0	76.5
Eating	2	64.4	60.6	70.2	68.1	59.5	54.6
Gym	2	83.7	85.6	83.4	82.0	84.0	89.4
Lab	3	81.5	81.1	78.5	71.3	84.6	93.9
Office	2	85.4	85.5	77.8	76.9	94.8	96.3

Table 4.3: Performance pre and post splitting on classes that contained multiple clusters.

Class	f-score		Precision		Recall		Accuracy	
	pre	*post*	*pre*	*post*	*pre*	*post*	*pre*	*post*
eating	87.5	90.2	84.6	88.5	90.6	92.0	87.1	90.0
communal places	88.6	91.7	85.6	89.2	91.8	94.4	88.2	91.5
transportation	93.6	83.5	95.4	93.4	91.9	75.6	93.7	85.1

Table 4.4: Performance before and after for two examples where CGL will merge two subsets of a tight cluster (visible in figure 4.3). In the third row we see the result of merging related sub-clusters which that CGL would not perform, although all subclusters are associated with transportation merging the results in worse performance.

of C_{AB}.

4.4.4 Experimental Results

Two common types of human error from low-commitment users during labeling are: (i) inconsistent class definitions between people since they are free to use their own definitions and (ii) unreliable boundaries that mark the start and end of classes that often occur due to user distraction. To study the ability of CGL to cope with these deficiencies we perform individual experiments, each of which focuses on one of these sources of error. In all experiments we evaluate CGL using standard five-fold cross-validation and tables 4.3 and 4.5 show the mean of the folds. As performance will not change for unsplit classes, we provide the average performance numbers: 88.7% accuracy and 89.3% f-score.

In our experimental data we find frequent disagreement of class definition occurs within our users (e.g., users providing different labels for the same activity). Table 4.3 and 4.4 demonstrate the ability of CGL to handle this type of human error.

Scenario	Precision		Recall		Accuracy		f-score	
	naive	*CGL*	*naive*	*CGL*	*naive*	*CGL*	*naive*	*CGL*
gym/driving	76.4	97.2	69.5	93.6	69.5	95.5	67.4	95.4
eating/lab	81.7	77.2	81.7	88.2	81.7	81.1	81.7	82.3
airplane/bus	91.4	99.0	91.9	99.8	91.6	99.4	91.7	99.4

Table 4.5: Performance under the class boundary errors experiment for a model using CGL relative to one that does not.

Table 4.3 shows classification performance for both the user-provided groupings (*pre*) and CGL's split groups (*post*). For majority of these classes, we see an improvement in the overall performance as measured by the f-score which is usually the best number to use for evaluating whether the classifier is correctly recognizing the true examples and rejecting the false ones. In particular, the recall numbers go up significantly for all but one class (eating). But as we will see later, eating may be a candidate for merging with other classes that are capturing eating related events (i.e., the classes corresponding to various eateries).

To evaluate the empirical result of inter-class similarity, we apply non-classical multidimensional scaling (MDS) to the similarity matrix and plot the result in figure 4.3. Distances between points in this figure are proportional to the differences in the similarity. The figure shows a clump of highly similar eateries in the bottom-left region as well as some fairly dissimilar transportation classes. The strange position of the 'library' class in the plot may be explained by the fact that our library includes a cafeteria area ('novack', also present in the plot). Additionally, we computed the similarity between the two eating subclasses and the various eateries. One of the eating subclasses is quite similar to the eateries (f-score = 0.70) and is a potential candidate for merging while other subclass is completely dissimilar (f-score = 0.0).

To calculate the performance of the post-split (*post*) classifiers, we consider the disjunction of each subclass classifier's output. To evaluate the benefits of merging, we evaluate individual classes' performance before merging and see if retraining merged classes improves the original performance. Table 4.4 shows that performance increases consistently for merging classes CGL determines to be similar. In this table each class refers to more than one user-provided grouping with the following associations in use: eating $\rightarrow$ { dirtcowboy, novack, quiznos, ramuntos }, communal places $\rightarrow$ { dirtcowboy, novack, quiznos, ramuntos, boloco, fiveolde, library }, transportation $\rightarrow$ { cab, airplane, driving, bus }. In contrast, if semantically similar classes (e.g., transportation) that our technique considers dissimilar are merged blindly, the performance decreases significantly.

To measure the impact of pronounced class boundaries errors we perform an experiment which uses the original data set but synthetically increases class boundary errors. We deliberately splice additional audio from a different activity into a class segment identified by the user. We base each of these synthetic examples of class pollution off naturally occurring scenarios we observe during the experiment. We present three scenarios in table 4.5 that include: data from the gym is mistakenly mixed with data sampled during the drive home or data acquired while eating is acci-

dently combined with data from in the lab. We compare the performance of CGL to boosted decision stump classifiers that treat the labels from users at face value. In the first scenario in the table, "gym/driving", CGL outperforms the naïve benchmark's accuracy by 41.5% and its other metrics by a similar margin. In contrast, CGL's accuracy is marginally outperformed in the second scenario "eating/lab". The reason for this is that the gym and driving classes are very easily split due to their large similarity distance (see figure 4.3). Due to the CGL is cleanly able to separate this class pollution and so it has little impact on the classifiers that are trained. However, in the case of "eating/lab" the classes themselves are have similar signatures so the improvement can only ever be minor.

4.5 Summary

In this chapter, we introduced CGL as a novel framework for building robust context classifiers. Our experimental results showed that CGL can overcome the limitations of existing techniques in coping with inconsistent labels, which are inevitable in real-world scenarios. By dynamically regrouping the classes that are modeled, CGL can recognize a wide range of classes more robustly than the conventional "train in a controlled environment then deploy" approach. Furthermore, user contributions can be intelligently shared to minimize unnecessary duplication of effort.

Under CGL the crowd-sourcing of labels from the general public becomes a viable option. In the following chapter we propose a complementary technique to CGL which exploits the crowd-sourcing of labels to train specialized classification models; one for each distinct community found in a large user population. By adopting this approach the challenges to robust classification posed by population diversity problem can be overcome.

Chapter 5

Scaling to Diverse Large-scale User Populations

5.1 Introduction

Perhaps the key message from Chapter 2 is that large-scale mobile phone sensing systems are not just possible but are likely in the near future. Unlike alternative sensing platforms (e.g., body-area networks [44] or embedded low-power sensors [88]) mobile phones are already deeply integrated into the lives of millions of people, as a result mainstream sensing via smartphones will not require any changes in the broader community. However, this vision ignores a critical technical challenge. Mobile sensing applications commonly require the interpretation of sensor-data but as user populations increase in size the differences between people cause the accuracy of classification to degrade quickly – we call this the *population diversity problem*. As the number of users of mobile phone sensing systems grow from the small scale of today to the very large scale of the tomorrow, it still remains unclear if the data collected can be interpreted and incorporated into applications. We demonstrate (see Section 5.2) that the population diversity problem exists and classification accuracy varies widely even as the user population is scaled to up as few as 50 people.

In this chapter, we propose *Community Similarity Networks (CSN)*, a classification system that can be incorporated into mobile sensing applications to address the challenge to robust classification caused by the population diversity problem, which will be present in large-scale deployments if left unchecked. The conventional approach to classification in mobile sensing is to use the same classification model for all users. Using CSN, we construct and continuously revise a personalized classifica-

tion model for each user over time. Typically, personalized models require all users to perform manual sensor-data collection where users provide hand annotated examples of them performing certain activities while their devices gather sensor-data (i.e., labeling data). This is both burdensome to the user and wasteful as multiple users often collect nearly identical data since the training of each model occurs in isolation of each other.

The key contribution of CSN is that it makes the personalization of classification models practical by significantly lowering the burden to the user through a combination of crowd-sourced data and leveraging networks of inter-personal similarity between users. CSN exploits crowd-sourcing to acquire a steady stream of sensor data and user input, either corrections to classification errors or the more common hand annotated examples of sensor data when performing an activity (i.e., labeling). Under CSN building classification models becomes a networked process where the effort of individual users benefits everyone. However, the use of crowd-sourced data must be done carefully, by selectively using crowd-sourced data during training so the resulting model is optimized for the person who is intend to use the model. CSN solves this problem by maintaining similarity networks that measure the similarity between people within the broader user population. We do this by three different proposed similarity metrics (i.e., physical, lifestyle/behavior and purely sensor-data driven) that measure different aspects of inter-person diversity which influence classification model performance. The CSN model training phase then utilizes a form of co-training to allow these different types of similarity to each contribute to improving the classification model used by a specific user.

The contributions of this chapter are as follows:

- CSN is the first system to propose embedding inter-person similarity within the process of training classification models for human activity. To the best of our knowledge, CSN represents the only activity recognition system designed specifically to cope with the population diversity problem, which would otherwise jeopardize large-scale deployments of mobile phone sensing systems.

- We propose similarity metrics and a classification training process that support: i) the extraction of similarity networks from crowd-sourced data and additional end-user input and ii) a learning process that adapts generic classification models through careful exploitation of crowd-sourced data guided by similarity networks.

- We have evaluated our system with two large-scale mobile sensor datasets each

comprising approximately 50 people. We measure the robustness of our classifiers and the ability of CSN to cope with population diversity.

5.2 Community-scale Classification

In this section we discuss a key difficulty in realizing large-scale mobile sensing applications. Specifically, we examine how population diversity can cause classification to become unreliable and inaccurate. We conclude by highlighting the inability of the existing state-of-the-art approaches to mobile classification to overcome this challenge.

One Size Does Not Fit All. As mobile sensing matures prototype systems are being deployed to user populations that are increasing in scale. Not surprisingly, accompanying this increasing scale is increasing user diversity. These users differ from one another in a variety of ways, a very concrete example being physical dissimilarities as measured by sex, weight, height or level of physical fitness. Beyond these visually obvious differences exist other examples based on lifestyle and background. People come from different ethnic and social-economic origins, live and work in different locations and while they may perform the same core collections of activities (e.g., socializing, exercising, working) they can do these activities in significantly different ways.

Inter-personal differences can manifest as differences in the discriminative patterns contained in sensor data that are used to classify activities, events and contexts. For example, the features from accelerometer data that allow classifiers to distinguish between the basic activities of walking and running can be completely different between a group of aged people (all older than 65 years) and a group of people who are in their 20s and 30s. Figure 5.1 visualizes this difference and plots the first two PCA components on each axis of the figure based on a range of already validated activity recognition accelerometer features [60] as these two groups are performing the same activity of walking. The very clear distinction between sensor data sourced from these two groups is surprising, particularly given the homogeneity you would expect in a simple activity like walking. To further quantify this problem we build a LogitBoost classification model [18] and reuse the same previously validated activity recognition features. This model is trained using the labeled data sampled from the group of people in their 20s to 30s. Classification accuracy while they walk and climbed stairs ranged from 80% to 90%. However, when this same classification model was used by the group of aged people the average accuracy dropped to nearly 60%. Clearly, a one size fits all approach to classification models will not scale to large user populations

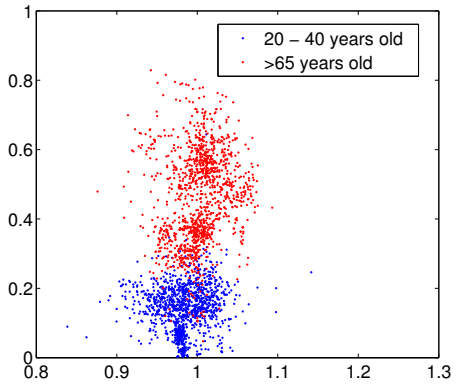


Figure 5.1: We visualize the differences in features under an identical activity, walking, for two distinct community sub-groups. One of which contains people over 65 years old with the other group ranging between 20 and 40 years of age. Here we show just the first two components of the PCA of these features.

which will contain many such groups.

This effect is not only limited to strictly physical behavior (e.g., walking, running or climbing stairs) but extends to a broader range of behavioral inferences. We investigate the breadth of this problem by performing an experiment on two distinct mobile sensing datasets. In what follows we briefly describe the experiment and datasets but a more comprehensive discussion is available in the evaluation section of this chapter. The first dataset (obtained from the authors of [113,114]) contains GPS sensor data for 51 people performing 7 different transportation modes (e.g., driving a car or riding a bike). The second is comprised of multi-modal sensor data (e.g., microphone and accelerometer data) for 41 people performing a range of everyday activities (e.g., walking up stairs, exercising, brushing teeth). We build a single classifier for each dataset and apply this classification model to the data of each person to measure the spread in classification accuracy within the user population. Figure 5.2 is a CDF which shows the range of accuracy levels for different people under both datasets. Accuracy levels for the transportation mode dataset are as low as approximately 40% for the bottom performing 60% of end-users to as high as 90% for the top 7%. Similarly, we find for the everyday activities dataset for 40% of the users the accuracy is only 12% and for even 80% of users the accuracy raises only marginally to 55%.

Limitations of Current Practice. The de-facto standard practice in incorporating classification into mobile sensing systems centers around a single unchanging classification model. This single model is deployed to the entire user population, a one-size-fits-all approach. The classification model is trained, prior to deployment,

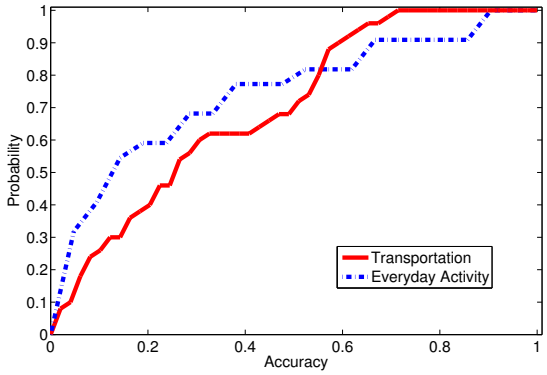


Figure 5.2: Classification accuracy varies significantly within a large-scale user population for two datasets, one containing everyday activities and the other transportation modes.

with controlled experiments that provide example sensor data. Then after system deployment the model remains the same without the opportunity for revision. Due to the reasons of population diversity this model works for some people, but not others; the accuracy of the system remains difficult to predict and increasingly unreliable as the user population grows larger.

Ideally the classification model could capture the distinctions between certain activities, when they are performed by subgroups in the user population, as different activities entirely (e.g., walking when performed by two sub-groups could be two different classes); whenever these distinctions impact the classification process. However, this would significantly increase the amount of examples required, as sufficient examples would be required for all these different variations of the same logical activity. Acquiring these examples is manually intensive (requiring the careful labeling of data segments), making this approach impractical as it simply does not scale.

A promising direction being actively explored is the personalization of classification models to improve accuracy (e.g., [52, 66, 67, 99]). These models are tuned to sensor data generated or encountered by the individual. Typically tuning occurs based on input from the user. For example, the user corrects classification errors or provides additional examples of activities by labeling sensor-data with the ground-truth activity occurring during the sampling of the data. Such user generated input allows the classification model to be retrained and to emphasize its robustness to types of sensor-data examples and specific errors the user provides to the system.

The limitation of such personalization of classification models is that accuracy *only* improves when and if people make the effort to manually provide additional sensor-data examples. It requires time and effort for users to provide suitable input.

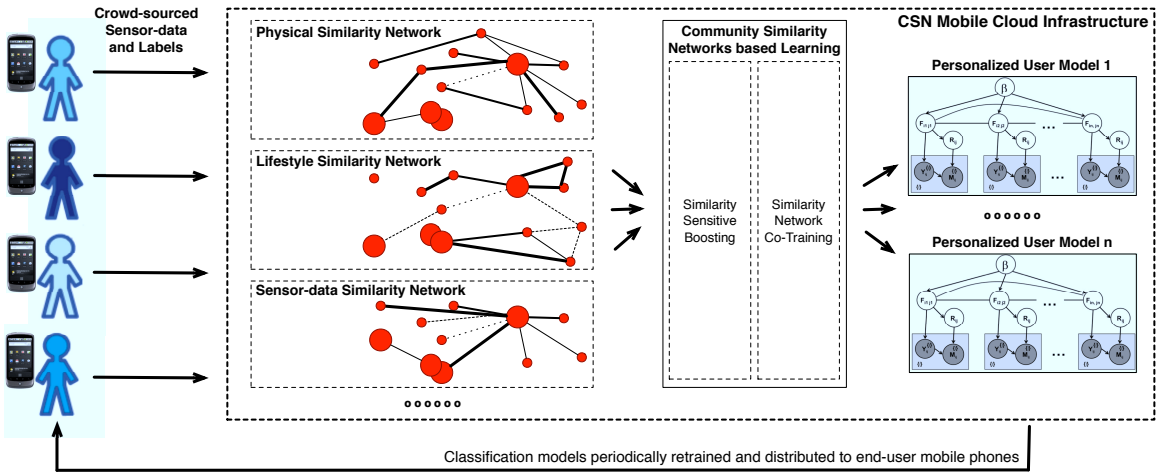


Figure 5.3: The processing phases within Community Similarity Networks

Independent of effort it will also take time for people to encounter certain situations that are good discriminative examples to incorporate into the model. The key problem with this type of gradual improvement of classification models by the user is that it leads to enormous amounts of redundant effort. Classification models are improved in *isolation* and each user potentially has to repeat steps that have already been done by other users to improve their own personal model.

5.3 Community Similarity Networks

In this section we describe system components and detail key algorithms used at each stage of CSN. The CSN system is designed to construct and periodically update personalized classification models for each user. Personalizing a model maximizes its classification accuracy when used by a specific target user. Typically, personalization is performed by an individual manually labeling a large amount of collected data, which is used to train a classifier specific to the person. A key novelty of CSN is that individuals need only provide small amounts of their own training data with the rest of the required training data being recruited from the other users with whom they share similar traits. CSN achieves this by incorporating into the model training process the selective sharing of crowd-sourced training data using measurements of inter-personal similarity.

5.3.1 Framework

Figure 5.3 illustrates stages within the CSN framework that produce personalized classification models for each user. Each of these stages occur either in one of two architectural components, the Mobile Phone Client software or the Mobile Cloud Infrastructure.

The Mobile Phone Client software samples sensor data to recognize human behavior and contexts by performing inference using classification models. While inference occurs locally the models themselves are downloaded from the Mobile Cloud Infrastructure. The client software also collects crowd-sourced training data comprised of both raw sensor-data and data segments that have been labeled with the ground-truth activity or context by users.

The Mobile Cloud Infrastructure is responsible for training classification models. Crowd-sourced data is used to construct similarity networks where network edges indicate the level of similarity between two users. CSN employes multiple dimensions of similarity (e.g., sensor-data, physical and lifestyle) to quantify the various ways users can differ. Several similarity networks are generated for each user, one for each of the similarity dimensions. Similarity-sensitive Boosting trains a classification models using each of the different similarity networks. Similarity Network Multi-training performs semi-supervised learning and improves each model by recruiting additional labels from the unlabeled pool of data. The final step of multi-training unifies the three independent classifiers, trained by similarity-sensitive boosting, into a single ensemble classifier ready to be installed on the phone of the user.

5.3.2 Mobile Phone Client

In the following subsection we describe key aspects of the Mobile Phone Client functionality, specifically: i) the classification pipeline which performs inference, ii) implementation specifics, and end with iii) collecting crowd-sourced sensor data and ground-truth labels of data segments.

Classification Pipeline. The classification process used by the client is one of: sensor data sampling, extraction of features and finally the recognition of an activity, event or context using a classification model. In this chapter our pipeline design choices are made to match the activities labeled, and the raw sensor data available in the two datasets we use for evaluation.

We use the accelerometer, microphone and GPS sensors to make a variety of proof-of-concept inferences. Our choice of features were based on observations made in prior

work [67,68,113,114]. For the accelerometer and microphone we use the same feature set described in [68] which include a variety of time domain and frequency domain features effective for general activity recognition. For the GPS we adopt features that were specifically designed in [113,114] for transportation inference based on time-series GPS readings. Classification occurs using a boosted ensemble [80] of naive bayes classifiers that each assume a gaussian distribution for each feature [18]. Inference results are temporally smoothed using a simple markov model. Although in our client the stages of the classification pipeline (i.e., features and classification model) remain fixed the parameters of the model are determined, and updated periodically, by the Mobile Cloud Infrastructure.

Implementation. Our prototype client is implemented on the Google Android Nexus One [4]. The design of the phone client is split between a portable classification pipeline library written in C++ and set of device specific supporting components (e.g., sensor sampling, end-user GUI). The library provides core classification pipeline components, including feature extraction and model inference. The device specific components are written in Java and connected with the library via a JNI bridge.

CrowdSourcing. CSN exploits the crowd-sourcing of both sensor data and user input to improve classification models. User input provides the ground-truth activity to segments of sensor data, with two specific types of user input supported by CSN. First, users can be asked to confirm or deny a class inference made by the model used by the client. For example, asking the user a question – *'Are you currently driving?'*. Such responses are used later when training classifiers as positive or negative examples of certain activities. Second, users can explicitly label data as being an example of an activity or event. For example, users indicate the ground-truth activity when a segment of sensor data was sampled by selecting it from a list presented on the phone GUI. These types of interactions with users can be incorporated into applications in various ways. As an example, simple binary yes/no questions can be presented when the user unlocks their phone. Alternatively more involved interaction, when users are selecting activities from a list, can be framed as software configuration or *calibration*. Similar forms of user interaction already occurs in real products, for example, reading training sentences into speech recognition software or running for precisely one mile to calibrate a single activity system like Nike+ [7].

5.3.3 Mobile Cloud Infrastructure

After first providing brief prototype implementation details in the remainder of this section we describe how the Mobile Cloud Infrastructure: i) computes similarity networks and ii) uses these networks to train personalized classification models which are distributed to all users.

Implementation. Our prototype implementation makes extensive use of Amazon Web Services [9] (AWS) which offer a number of generic components useful in building a distributed system, these include message queues (SQS), a simple queryable hash table (SimpleDB) and binary storage (S3). Each stage of the model training performed by the cloud is implemented either as python scripts or C++ modules depending on available library support given the required functionality. These stages run on a pool of linux machines as part of the Amazon Elastic Cloud product and interact with the individual AWS services as needed. Once a classification model is complete it is serialized into a JSON-like format and written to the binary storage (S3), ready to be downloaded by the client.

5.3.4 Similarity Networks

Every similarity network is built from the perspective of a single target CSN user. Nodes in the network represent other CSN users and edge-weights measure the degree to which the target user is similar to these other users. The CSN framework is designed to leverage multiple similarity measurements, which capture different dimensions of affinity between people. For each user CSN maintains multiple similarity networks, one for each dimension supported. Depending on the activities or contexts that are to be recognized different dimensions may be utilized. Specifically, in this chapter we propose the use of three dimensions of inter-person similarity: sensor-data similarity, physical similarity and lifestyle similarity. Each of these different dimensions require varying amounts of computation and we find them to be effective in classifying different categories of activities and contexts (see evaluation section for more details). However, CSN is agnostic as to the exact similarity dimensions used.

We now detail the current similarity dimensions included within the CSN framework.

Physical Similarity. Physical differences between people (e.g., weight, height, age, level of physical fitness or wellbeing) will vary greatly from person to person within a large user population. Such differences can alter the way people move and perform certain physical activities. For example, as we detail in the prior section on

community-scale classification these differences can effect seemingly simple everyday activities like walking upstairs or jogging. Consequently, we incorporate a measure of physical similarity within our design of CSN.

CSN computes a single physical similarity value based on five features: the age, height, weight, and the scores from two well established physical well-being surveys (Yale Physical Activity Survey [32] and SF-36 physical activity score [8]). For this given vector, we use a mahalanobis distance to compute the physical similarity between two users as follows,

$$sim(i,j)^{phy} = exp(-\gamma(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)) \tag{5.1}$$

where, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the physical feature vectors of user i and user j, Σ is the covariance matrix used to keep the scale invariant property of feature vectors, and γ is a scaling parameter.

Lifestyle Similarity. This metric attempts to capture the diversity in how people lives their lives, examples of which include: occupation, diurnal patterns (e.g., are they an early morning person or active late at night), the distribution of activities performed, mobility patterns and significant places [15] (e.g., where they work and live). Occupation and the location of work alter, for instance, the accelerometer and audio patterns occurring during social interactions (e.g., meetings and conversations). The time of day and significant places can effect the background context in which people perform activities, for example, late at night or early in the morning different locations will have different background activities that alter the sampled data (e.g., noise from people or cars) – all of these factors an change the distribution of features and shift effective discriminative boundaries for recognizing classes of activity. Later in our evaluation section we show that the use of lifestyle similarity can benefit the accuracy of particular classes of inference (e.g., driving).

As proof-of-concept we compute this metric based on: mobility and diurnal patterns in combination with the distribution of activities performed by users. Mobility patterns are considered as a time-series of blurred GPS estimates. Diurnal patterns are presented as the times at which the user is inferred to be doing any activity other than being stationary, with time represented as the particular hour during a day in the week (i.e., ranging from hour 0 at the start of the week to hour 167 on the final hour of the final day). The activity distribution is based on the frequency of inferred activities.

We compute the similarity for each of these different measurements of lifestyle in

precisely the same way. First, we tesselate the data into m distinct one dimensional (or two dimensional in the case of location) tiles, each of which can be regarded as a bin. For each user, we then construct a histogram $\{\boldsymbol{T^{(k)}}, k \in [1, m]\}$ of these bins. Histogram frequencies are normalized and the value of the histogram vector reflects the distribution of the data (e.g., location) belonging to that user. For each pair of users (i, j), we compute the lifestyle based similarity by the following equation:

$$sim(i,j)^{life} = \sum_{k=1}^{m} \boldsymbol{T_i^{(k)}} \boldsymbol{T_j^{(k)}} \tag{5.2}$$

Sensor-data Similarity. A wide variety of the differences between users (e.g., lifestyle, behavioral patterns, location and even culture) will manifest as differences in their sensor-data. Unlike lifestyle similarity it does not require the development of explicit features to extract the dimension from sensor data. Similarly, unlike physical similarity it does not require additional information potentially provided by the user. Instead, measuring inter-person similarity based on sensor data is inherently purely a data-driven approach. Later, we report it is effective across a wide range of the classification tasks that we encounter during evaluation. However, it requires much larger amounts of computation to determine similarity between users than computing lifestyle or physical similarity.

Computing similarity based on the raw sensor data will be effected by noise and capture too many insignificant variations in the data. Instead, we compute sensor-data similarity between the features extracted from the raw data. For this purpose CSN employes the same features used by the classification pipeline, described earlier in this section. Individual users will accumulate varying amounts of sensor data based on how frequently the use their device. Consequently, we compute "set" similarity whereby any duplicate feature vector for a user is ignored and only the unique vectors generated from the data of a person are used. For our similarity measurement we adopt a commonly used formulation [110] where the similarity between two users is,

$$sim(i,j)^{data} = \frac{1}{N_i N_j} \sum_{l=1}^{N_i} \sum_{m=1}^{N_j} sim(\boldsymbol{x}_{il}, \boldsymbol{x}_{jm}) \tag{5.3}$$

where, $\{\boldsymbol{x}_{il}, l = 1 : N_i\}$ is the data of user i, and $\{\boldsymbol{x}_{jm}, m = 1 : N_j\}$ is the data of user j.

However, this pairwise computation quickly becomes impractical as the average data number of users increases. To cope with this problem, we adopt Locality Sen-

sitive Hashing (LSH) [14] to construct a histogram to characterize the "set" of data from each user and then compute the similarity between a pair of users by applying this histogram representation. Our method obviates the need to compute the pairwise relations of data from two users as the traditional "set" similarity, which has a linear time complexity with the average data number of each user. The basic idea of the LSH method is that the hashing function family can capture the similarity between data. In other words, similar data have a high probability to share the same value after hash mapping.

$$Pr_{h\in\mathcal{H}}[h(\boldsymbol{x_1}) = h(\boldsymbol{x_2})] = s_{\mathcal{H}}(\boldsymbol{x_1}, \boldsymbol{x_2}) = E_{h\in\mathcal{H}}[s_h(\boldsymbol{x_1}, \boldsymbol{x_2})] \tag{5.4}$$

Therein, $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{X}$ are two data, $\mathcal{H}$ is a LSH family, h is the hash function sampled from $\mathcal{H}$, and $s_{\mathcal{H}}$ is a similarity measure of $\mathcal{X}$, which is induced by the LSH family $\mathcal{H}$ [14].

In CSN, we randomly choose B independent 0/1 valued hashing functions $\{\boldsymbol{h_i}\}$ from the *random projection for $\mathcal{L}_2$ distance LSH family* [14] and form a $B-$bit hash function $f = (\boldsymbol{h_1}, \boldsymbol{h_2}, , ..., \boldsymbol{h_B})$. The number of functions B controls the tradeoff between efficiency and accuracy [14].

We apply the $B-$bit hash function to build histograms for each user, whose size is 2^B. Now, we formalize how to construct a histogram from the features of the user. According to the description, let $\mathcal{X}$ be the data space, $\mathcal{F}$ be the B-bit hash functions family mapping from $\mathcal{X}$ to $\mathcal{D} = \{0, 1, \ldots, 2^B - 1\}$, and $\{\mathbf{e}[i] \,|\, i \in \mathcal{D}\}$ be the standard basis of the $|2^B|$-dimensional vector space. Hence, given $h \in \mathcal{H}$, the histogram T_f for any user i is defined as follows,

$$\mathbf{T}_f(i) = \sum_{\boldsymbol{x_{il}}\in i} \boldsymbol{e}[f(\boldsymbol{x_{il}})] \tag{5.5}$$

here, $\{\boldsymbol{x}_{il}, l = 1 : N_i\}$ is data of user i, and $\mathbf{T}_f(i)$ is determined by the hash function f sampled from $\mathcal{F}$.

Thus each dimensionality of the histogram vector $\mathbf{T}_f(i)$ can be regarded as a bin to record the frequency at which data from user i is mapped into it. As the value of the hash function indicates the probability that two data share the same value after mapping, two users that have many "matched" values in the corresponding bins of histograms implies a high similarity between them. The inner product of the two

histogram vectors is next applied to compute the similarity metric for the two users:

$$sim(i,j)^{data} = \mathbf{T}_f(i)^\top \mathbf{T}_f(j) \tag{5.6}$$

To estimate the expectation shown in Eq. 5.5, we construct several histograms $f \in \mathcal{F}$ for each user and compute an average value using Eq. 5.6.

The time complexity of computing the LSH based similarity metric is linear with the average quantity of data for each user. Compared with the pairwise computing method shown in Eq. 5.3 which is quadratic, the LSH based similarity metric is very efficient.

5.3.5 Community Similarity Networks based Learning

Learning the personalized classification models for each user occurs in two stages under CSN. First, Similarity-sensitive Boosting trains three separate classifiers, one for each type of similarity network that is maintained for every user. Each of the classifiers have different strengths when recognizing user activities, not only are they personalized to the characteristics of the specific individual who will use it but it also specializes in recognizing specific categories of activity depending on which similarity network is used (e.g., physical similarity performs well with physical activities like climbing stairs). Second, Similarity Network Multi-training occurs which: i) uses a semi-supervised approach to recruit additional labels from the unlabeled pool of crowd-sourced data leveraging the different strengths of each classifier, and ii) unifies the three classifiers, trained by Similarity-sensitive Boosting, into a single final ensemble classifier ready to be installed on the phone of the user. Our use of multi-training exploits the diverse perspectives of each classifier to recruit, in turn, greater amounts of training data to the benefit of the other classifiers; compensating for the fact that much of the crowd-sourced data is unlabeled.

Similarity-Sensitive Boosting. A personalized classification model emphasizes the particular characteristics found in a target user to increase accuracy. CSN accomplishes personalization using a modified online boosting algorithm [80]. Boosting is a common learning technique that builds model that is a composite of several weak classifiers trained over multiple iterations. At each iteration certain data segments are weighted higher than others. Under conventional boosting these weights are only altered based on the classification performance of the weak learner trained during the previous iteration. Those data segments that were incorrect are weighted higher than others so the weak classifier produced in the next iteration will be better

able to classify these previously incorrect segments. CSN modifies this process by imposing an additional term to the weight at the initial iteration. This weight is based on the similarity between the user i, whose personalized model is being trained, and the user which provides the data,

$$weight^{(0)}(\boldsymbol{x}_k) = sim(i,k) \tag{5.7}$$

where, k indicates the user who produces the data $\boldsymbol{x}_k$. $sim(i,k)$ is defined as the edge weight between these two individuals within the similarity network being used during the boosting process. As a consequence of this modification only data segments from user i or any users who are highly similar to user i will be weighted highly and able to have strong influence over the learned classification boundaries. In subsequent iterations the weighting of data segments is left to fluctuate based solely on classification performance. As boosting is an ensemble technique the CSN framework remains flexible as the weak learner can be replaced with any alternative supervised classifier based on the requirements of intended classification task.

Similarity Network Multi-Training. The three varieties of similarity networks used in CSN, namely, physical, lifestyle and sensor-data, capture different dimensions of similarity between users. Each network may even have completely different topologies with, for example, some users being highly similar in terms of physical characteristics but polar opposites when it comes to lifestyle. Using similarity-sensitive boosting in conjunction with any of these different networks will result in different classification models as each network will emphasize different partitions of the crowd-sourced training data. This diversity is valuable as the different similarity networks produce models that are highly effective for some classes of activity but not others (see evaluation section). A simple example of this being those activities that are closely connected to the physical characteristics of the person, e.g., running and exercising, benefit from a classification model trained using a physical similarity network.

CSN exploits the strengths of each similarity network by adopting the technique of multi-training (a variation of co-training proposed in [116]). Multi-training is a semi-supervised training algorithm designed to utilize multiple complementary views of the same labeled training data to generate additional labels which are assigned to data segments within the pool of unlabeled data. This approach is appropriate for CSN given that crowd-sourcing generates large amounts of unlabeled data. People will only infrequently take the time to provide any manual user input; but since

simply collecting data is transparent to the user then large pools of unlabeled data quickly can accumulate. Employing a multi-training approach allows CSN to use the diversity provided by the different similarity networks to make use of a plentiful and otherwise wasted resource, unlabeled data.

The multi-training process to train a classification model for one particular user begins by initially using the three classifiers trained by Similarity-sensitive Boosting. Each of these classification model maintains an independent logical copy of the labeled and unlabeled crowd-sourced data. An iterative process is applied whereby the classification models are used to in turn to "label" the unlabeled portions in the logical datasets maintained by each of the other models. At the end of each iteration the classifiers are then retrained (using Similarity-sensitive Boosting) based on the combination of the labeled data from the previous iteration along with any new additional labels. Acquiring labels in this way can be an error-prone process, as a result labels are only accepted when there is agreement with more than half of the classification models. Judging the quality of a proposed new label, based on a majority decision, is only one of many ways that quality can be assessed. Multi-training continues to iterate for several round until a stopping condition is met. CSN uses currently a stopping condition based on how many labels are accepted at each iteration. If the number of recruited labels is too low for too many iterations then multi-training stops.

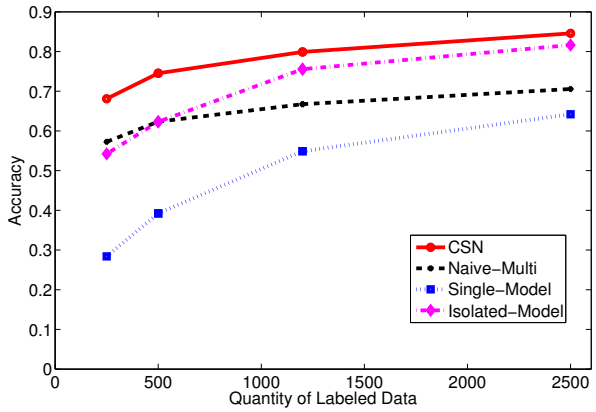
5.4 Evaluation

In this section we evaluate the effectiveness and design choices of CSN. Our experiments show that by incorporating similarity networks between users into the classification process CSN is better equipped to cope with the population diversity problem than previously known techniques.

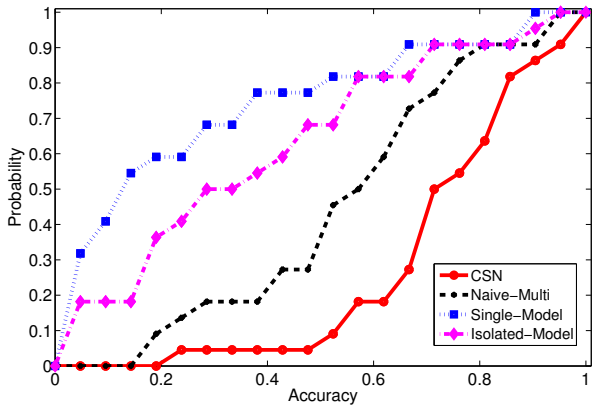
5.4.1 Experimental Methodology

To evaluate CSN we use two large real-world datasets and three representative baselines.

Datasets. Our two datasets require a variety of activity inferences which are in frequent use in mobile sensing applications today. The first dataset, *Everyday Activities*, contains a broad range of routine human activities that have been used to support application domains like mobile health [28]. The other, *Transportation*, is much more focused on a single category of activity, transportation modes. These



(a) Accuracy



(b) CDF

Figure 5.4: Classification accuracy for the Everyday Activities dataset under CSN and three baselines.

inferences are building blocks used, for instance, to promote green transportation [39] in society. We collect the data for *Everyday Activities* as part of series of internal experiments. The data comprises both simple activities: {*walk*, *run*, *stationary*} and high-level behaviors: {*meeting* , *studying*, *exercising*, *socializing*}. A total of 41 people contribute to this dataset using a Nexus One smartphone sampling sensor data from the accelerometer, microphone and GPS. People carry the device for variable lengths of time. For *Transportation* we use an external source [113, 114], with the dataset containing only different transportation modes, specifically: {*bike*, *bus*, *car*, *walk*}. This dataset comprises 51 people who carry for three months one of a variety of mobile devices that are equipped with a GPS, including, smartphones, PDAs and personal navigation devices.

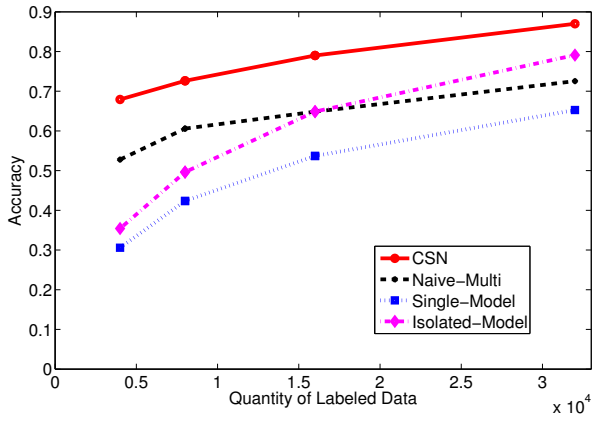
Benchmarks. We compare the performance of CSN against three benchmarks,

`single`, `isolated` and `naive-multi`. Our benchmarks use the same features and apply the same classification model as CSN but differ significantly in how they approach the training of the model. The benchmarks of `single` and `isolated` correspond with the two types of common practice we describe in the later part of the prior section on community-scale classification. In `single` the same generic model is provided to all users. The classification model is trained, prior to the deployment of a system, on all available labeled data. Unlike CSN after the release of the system the model does not change and new training data is not collected. Under `isolated` every user has their own model. Each user model is personalized by using training data sourced directly from the user. The weakness is that each classification model is considered in isolation of one another. No co-operation or sharing of training data occurs between users. Finally, `naive-multi` allows us to demonstrate the benefit of CSN solely attributable to the use of similarity networks. During training `naive-multi` performs the conventional forms of precisely the same learning techniques as CSN during model training, namely, boosting and multi-training. However, in their conventional form none of the techniques are able to exploit community similarity networks. Specifically the differences are: i) the weighting of training data during boosting only changes the performance at each iteration instead of based on similarity and ii) the assignment of people to subgroups during multi-training occurs not due to similarity but randomly.

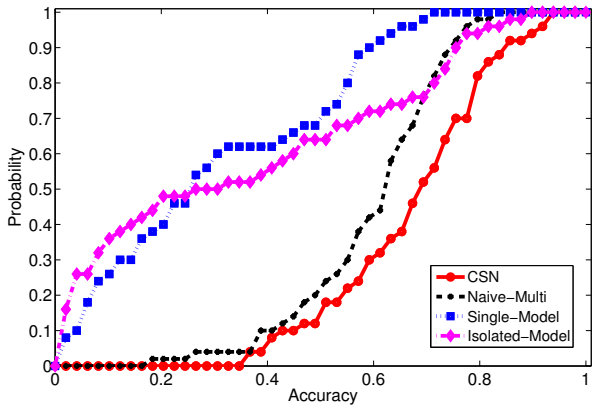
5.4.2 Robust Classification with Low User Burden

Our first set of experiments finds CSN provides more robust classification than any of the benchmarks under both datasets. Not only is CSN able to achieve higher classification accuracy but we observe classification accuracy is also more evenly distributed throughout the user population. Moreover, under CSN the burden to provide training data is lowered as CSN can offer comparable levels of accuracy relative to the benchmarks, but with smaller quantities of crowd-sourced data.

Figures 5.4(a) and 5.5(a) show the results of experiments where we assume users contribute different amounts of labeled data. For each quantity of labeled data we measure the average per person accuracy of classification for models trained under CSN and the three other benchmarks. Figure 5.4(a) uses the the *Everyday Activities*, and Figure 5.5(a) repeats the experiment using the *Transportation* dataset. In both figures the accuracy of CSN outperforms all baselines for each quantity of training data tested. For example, Figure 5.4(a) shows if 500 labeled data segments are used (approx. 15 minutes of training data from each user) then CSN outperforms



(a) Accuracy

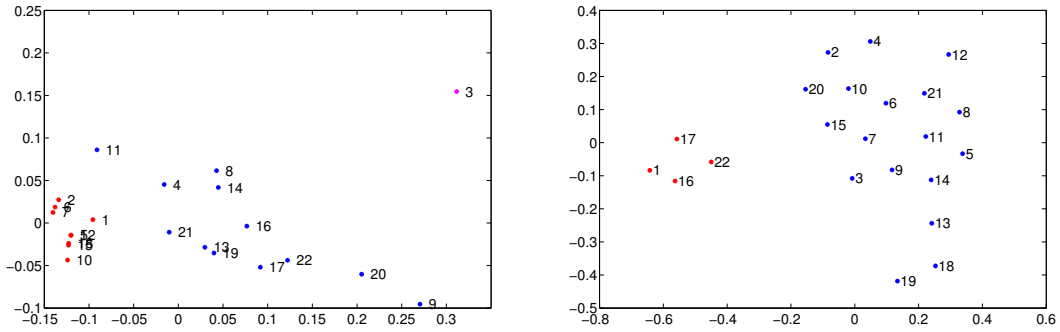


(b) CDF

Figure 5.5: Classification accuracy for the Transportation dataset under CSN and three baselines.

`naive-multi` and `isolated` by 22%. Similarly, from Figure 5.5(a) we see if 1.6 x 10^4 labels are used (approx. 137 minutes of training data from each user) the accuracy of CSN exceeds `single` by 47% and `naive-multi` by 32%.

From Figures 5.4(a) and 5.5(a) we also learn that CSN is able to lower the user burden of contributing training data. As an example, Figure 5.4(a) shows `isolated` requiring 36 minutes of training data from a user to achieve 74% accuracy. CSN can provide approximately this same accuracy for only 15 minutes of training data, a reduction of 58%. Alternatively, if we consider Figure 5.5(a) `isolated` is able to perform with 77% accuracy but requires 270 minutes of training data. Again, CSN can provide approximately this level of accuracy but with 49% less data, only needing 137 minutes of crowd-sourced data per user. Under CSN users are better rewarded for contributing data due to a higher ratio of crowd-sourced training data to classification



(a) Physical (b) Lifestyle

Figure 5.6: MDS projection of example physical and lifestyle similarity networks used by CSN

accuracy.

Figures 5.4(b) and 5.5(b) present CDFs of per person accuracy within the user population. We illustrate the fraction of the user population who receive different levels of classification accuracy under CSN and all benchmarks. Figures 5.4(b) uses the *Everyday Activities* and assumes users provide 15 minutes of training data each, while Figure 5.5(b) assumes users provide 137 minutes of labeled data and uses the *Transportation* dataset. Ideally all users should receive the same consistent level of accuracy, otherwise classification accuracy will be unpredictable when deployed. Better performance is indicated on these figures by curves that are furtherest to the right. We observe from each figure CSN has the most even distribution of accuracy compared to all benchmarks. For example, Figures 5.4(b) shows for 75% of users CSN provides 82% accuracy compared to just 65% for
`isolated`, 48% for `single` and 52% for `naive-multi`. Figure 5.5(b) reinforces this finding and indicates for again 75% of users CSN provides 77% accuracy instead of the 68%, 53% and 66% accuracy offered by `isolated`, `single` and `naive-multi` respectively.

5.4.3 Benefits of Leveraging Similarity Networks

With the following experiments we investigate the effectiveness of the similarity networks used by CSN.

To test if the similarity networks used by CSN are capturing meaningful differences between people we collect additional demographic information from 22 of the people who contribute to *Everyday Activities*. Figures 5.6(a) and 5.6(b) plot the result of applying multidimensional scaling (MDS) to two similarity matrices for these people

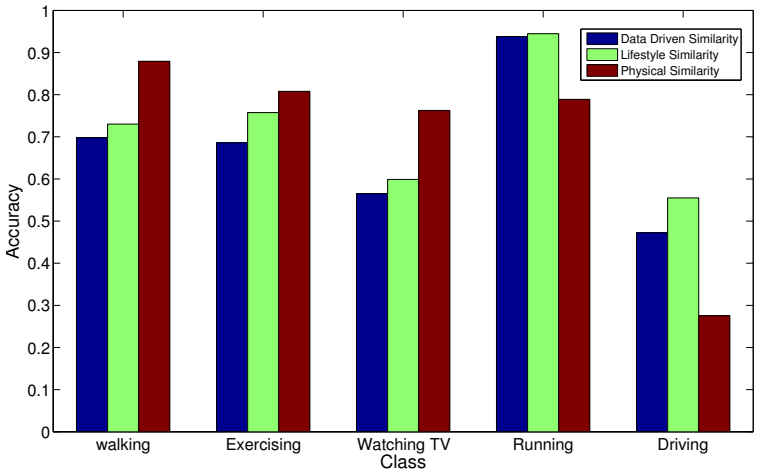


Figure 5.7: The classification accuracy of each activity class under different similarity dimensions on Everyday Activity. It shows different similarity dimensions are effective for different activities

using physical and lifestyle similarity. Distances between points in these figures are proportional to the differences in the similarity. Figure 5.6(a) shows two clear clumps. We find these correspond to people with similar physical characteristics, the tight cluster near the bottom are all over 30 years old, all male, and have similar physical levels. In contrast, the looser clump of people near the middle are in the same age range (22 - 26) but have diversity in sex and fitness. The outlier in Figure 5.6(a) is a 50 year old woman and is distinct due to her sex and exceptional fitness. The clusters in Figure 5.6(b) also correspond to our interview ground truth. The tight cluster to the left were a small group of people who lived off campus and maintained regular 9 to 5 working hours. They were in sharp contrast to the very loose cluster on the right of the figure, which is contains students who although live very close to each also have erratic sleeping and activity patterns which results in them being grouped but not that tightly.

In Figure 5.7 we can see the value of using multiple similarity dimensions. This figure illustrates the different levels of classification accuracy achieved when using each of our three similarity dimensions to classify classes found in the *Everyday Activities* dataset. We see that no one single similarity metric performs best for all the activities. We find a similar pattern exists within the *Transportation* dataset. By exploiting all of these forms of similarity CSN is able to better handle a wide range of classification tasks. This result supports the design choice to use multiple dimensions of similarity and leverage them all when training classification models.

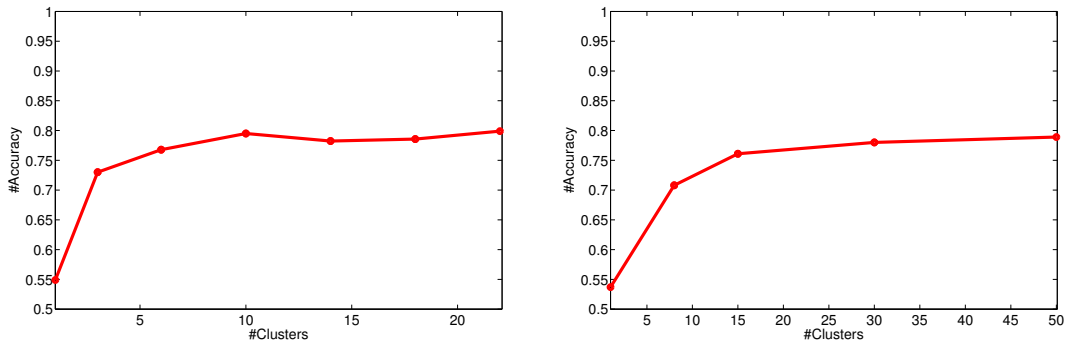
5.4.4 Cloud Scalability with Low Phone Overhead

Our remaining results report on the overhead to smartphones of adopting CSN along with the ability for CSN to scale to large user populations.

We profile the computation and energy consumption of our CSN client on the Android Nexus One. We find resource consumption comparable to prior implementations of classification pipelines on phones (e.g., [68, 75]). As this overhead is not specific to CSN but found in any mobile sensing application we do not report further details. Overhead specific to CSN includes the transmission of sensor data and the downloading of classifiers trained in the cloud. We find typical file sizes for our classification models are on the order of $1 \sim 2$ KBs, which means the cost of downloading classification models is minor. However, a significant cost can accrue to the phone due when uploading sensor data. To eliminate this cost our client implements an uploading strategy that waits until the phone is recharging before uploading data, effectively removing any burden to the battery.

The computational demands of computing the three CSN similarity dimensions range from being light-weight to very demanding. We quantify this by profiling the computational overhead for computing similarity networks for all people within the *Everyday Activities* dataset. This dataset is more than 400GB (due mainly due to audio data). Using our CSN Mobile Cloud Infrastructure, configured with only one linux machine in the node pool the computational time for each variety of similarity is, ≈200 minutes, ≈9 minutes and ≈3 minutes for sensor-data, lifestyle and physical similarity respectively. The sensor-data similarity is the most costly of these three as it requires pairwise calculations between users.

Personalized models are trained by CSN for each user, however, this can become a bottle-neck. The workload of the CSN Mobile Cloud Infrastructure increases with



(a) Everyday Activities (b) Transportation

Figure 5.8: The accuracy of CSN when we group the users into different number of clusters under both datasets

population size due to: i) the pairwise calculation of similarity between users and ii) each new user requires a new model to be both trained. For this reason we designed our Mobile Cloud Infrastructure to effectively leverage variable pool of cloud nodes, so additional nodes can be added when required, but there is an alternative. A simple extension to CSN allows it to scale any population size without incurring additional computation. Instead of training a model for each user, users are first grouped together by clustering. Similarity networks are then built not between people but between these groups, with a model trained for each group. We investigate this trade-off and cluster people with k-means using the least computationally costly similarity dimensions, lifestyle and physical. By lowering the number of groups we can reduce the Mobile Cloud Infrastructure workload. This will also sacrifice accuracy as lowering the group count has the side-effect of clustering together less similar people This trade-off is seen in Figure 5.8. These figures illustrates how accuracy falls as the cluster size (the k in k-means clustering procedure) is reduced. Reducing the number of models dilutes the similarity between people in the cluster. Consequently, the model used by the entire group is less appropriate for everyone. Still, as the cluster number decreases the overhead to the mobile cloud is reduced, since fewer models need to be maintained.

5.5 Related Work

Interest into the potential of sensing with mobile phones has been steadily building (e.g., [10, 22, 24, 28]). This area is presenting an emerging new sensor-based mobile application domain to explore (e.g., [28, 75]). The importance of classification is becoming clearer as applications mature.

The limits of activity recognition and more general types of classification are frequently encountered by those investigating sensor-enabled mobile phones. It is becoming obvious that conventional approaches that rely on supervised learning and carefully controlled training experiments are not suitable. In recognition researchers are considering alternatives. Current research directions point towards models that are adaptive and incorporate people in the process. Automatically broadening the classes recognized by a model is discussed in [66] where active learning (where the learning algorithm selectively queries the user for labels) is investigated in the context of heath care. In SoundSense [67] a general supervised classification pipeline for sound classification is combined with a unsupervised learning. This body of work focuses primarily on the individual to assist with classification. CSN leverages the user but also

exploits communities of people (rather than just isolated individuals). Researchers are beginning to explore this direction. The clearest example being Community-guided Learning (CGL) [84], which was proposed in Chapter 4. CGL uses both data similarity and crowd-sourced labels to improve the classification accuracy of the learning system. It focuses on the problem of noisy labels being introduced to the training process by using similarity in the data associated with the labels. CGL is complementary to CSN. CSN relies on crowd-sourced data and CGL proposes a technique to clean crowd-sourced labels before model training.

The potential for crowd-sourcing has been long recognized with interest in the area being established by Luis Von Ahn [106]. Now, commercially available systems including Amazon's Mechanical Turk [1] have made it simple to exploit the power of using thousands of people. The use in CSN of crowd-sourcing builds directly on these existing directions. We see CSN as part of an exciting area of hybrid systems that intelligently combine the effort of the masses towards a task that neither computers nor humans can perform on their own.

5.6 Summary

In this chapter, we have proposed Community Similarity Networks (CSN), a classification system designed to address the population diversity problem. We demonstrated that the population diversity problem appears, when using conventional techniques, with as few as 50 users. CSN relies on the crowd-sourcing of labels and sensor-data combined with multiple similarity networks used to identify communities of people who require personalized classification models. To learn a personalized model CSN leverages the different perspective of each similarity networks by adopting form of co-training. We implement a complete CSN prototype system including an efficient mobile client and a mobile cloud backend. To examine the generality of our approach we evaluated CSN using two different mobile classification datasets. One dataset spanning a range of commonplace activities that people perform on a daily basis, the other focusing on a single category of activity, namely transportation mode. The ability of CSN to remain effective under both of these distinct classification problems highlights the flexibility of the approach.

Chapter 6

Case Study: Monitoring, Modeling, and Promoting Overall Wellbeing

6.1 Introduction

In this final chapter we shift focus from the difficulties of understanding sensor-data collected from smartphones. Instead, we consider how the techniques for constructing robust and scalable classifiers that we have developed over the preceding three chapters can have real-world impact. In what follows we describe a system which relies on classification to allow mobile phones to continuously monitor a broad set of health outcomes. Such fine-grain monitoring of overall wellbeing in the mainstream population has the potential to revolutionize the way society diagnoses and treats medical conditions.

Our lifestyle choices have a deep impact on our personal health. For example, our sleep, socialization and exercise patterns are connected to the presence of a wide range of health related problems such as, high-blood pressure, stress [79], anxiety, diabetes and depression [37, 41]. Positive health effects can be observed when these wellbeing indicators (e.g., sleep, physical activity) are kept in healthy ranges. However, people are typically not exposed to these health indicators as they go about their daily lives. As a result, unbalanced unhealthy lifestyles are present in the general population. People demonstrate concern for some aspects of their wellbeing, such as fitness or diet, yet neglect the wellbeing implications of other behaviors, such as, poor sleep, hygiene or prolonged social isolation. We believe this situation is caused by an absence

of adequate tools for effective self-management of overall wellbeing and health.

We envision a new class of *personal wellbeing applications* for smartphones capable of monitoring multiple dimensions of human behavior, encompassing physical, mental and social dimensions of wellbeing. An important enabler of this vision are the recent advances in smartphones, which are equipped with powerful embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone, and camera. Smartphones present a programable platform for monitoring wellbeing as people go about their lives [58]. It is now possible to infer a range of behaviors on the phone in real-time, allowing users to receive feedback in response to everyday lifestyle choices that enables them to better manage their health. In addition, the popularity of smartphone application stores (e.g., the Apple App Store, Android Market) has opened an effective software delivery channel whereby a wellbeing application can be installed in seconds, further lowering the barrier to user adoption. We believe production-quality wellbeing applications will gain rapid adoption globally, driven by: i) near zero user effort, due to automated sensor based activity inference and ii) universal access, only requiring a single download from a mobile phone application store and installation on an off-the-shelf smartphone.

A number of technical barriers need to be tackled to make this vision a reality. The majority of existing mobile health systems have focused on a specific health dimension (e.g., stress [35], diet [82], physical activity [28]) and consider only one or two types of behavior. Instead, personal management of wellbeing requires applications that monitor a diverse range of daily behaviors, which have broad health related consequences. Unfortunately the small, but growing, number of mobile health applications that consider a wider perspective of user health commonly rely on manual data entry [82]. This type of manual effort is burdensome to users and is unlikely to scale to mass adoption. Rather, the automated and continuous inference of the users behavior and environment based on embedded sensors promotes sustained usage over the long-term. Ultimately, to be effective these applications must understand the impact on the health and wellbeing of the user due to the observed behavioral patterns. Simply reporting behavioral patterns to the user is not necessarily intuitive. It can be difficult for a user to identify which behaviors have a larger impact on wellbeing. Therefore, providing intuitive and interpretable feedback is a key challenge for *wellbeing monitoring apps*.

In this chapter, we present BeWell, a personal health application for smartphones that is designed specifically to help people manage their overall wellbeing. BeWell continuously monitors multiple dimensions of behavior and incorporates user feedback

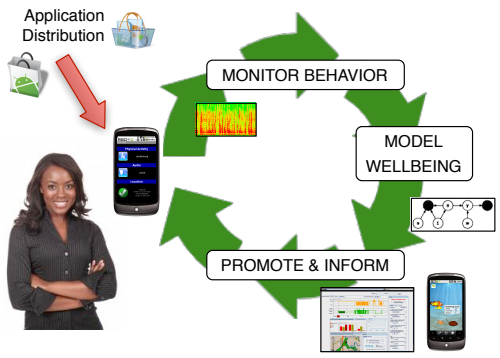


Figure 6.1: BeWell approaches end-user self-management of wellbeing with three distinct phases. Initially, everyday behaviors are automatically monitored. Next, the impact of these lifestyle choices on overall personal health is quantified using a model of wellbeing. Finally, the computed wellbeing assessment drives feedback designed to promote and inform improved health levels.

mechanisms that are able to increase awareness of how different aspects of lifestyle impact the personal wellbeing of the user. BeWell uses commercial, off-the-shelf smartphones (Android Nexus One [4]) to automatically: (i) monitor a person's physical activity, social interaction and sleep patterns; (ii) summarize the effect of the monitored behavior on wellbeing; and (iii) provide feedback that enables users to effectively manage these three key aspects of their health. We present the design, implementation and evaluation of BeWell, an *automated wellbeing monitoring app* for the Android smartphones. Our detailed evaluation incorporates both system benchmarks and controlled experiments, along with a 19 day, 27 person field trial. Results indicate that not only are popular consumer smartphones a viable platform for *wellbeing monitoring apps* but users are capable of digesting and responding to multidimensional wellbeing feedback.

6.2 BeWell Architectural Design

In this section, we describe the BeWell architecture which includes cloud computing servers as well as smartphones. The operational phases of the BeWell system are shown in Figure 6.1 and discussed below.

6.2.1 Monitor Behavior

The BeWell application automatically infers behavioral patterns using sensors embedded in smartphones. Rather than tracking a single wellbeing dimension, such as,

physical activity, the BeWell system concurrently monitors multiple dimensions (e.g., sleep patterns, social interaction, and physical activity), representing a more complete picture of the user's overall wellbeing. The current implementation of BeWell is limited to three wellbeing dimensions and does not yet incorporate a number of other important health components such as diet and stress. Users are able to manually add new behaviors by using a diary maintained by the BeWell web portal, which visually show the inferences made by the application and allows users to correct classification errors.

6.2.2 Model Wellbeing

Simply collecting user patterns of behavior is insufficient. Users need to easily understand how their behavior affect different dimensions of their personal wellbeing. Using existing guidelines provided by healthcare professionals, we estimate multi-dimensional wellbeing scores that capture the relationship between behavioral patterns and health outcomes.

6.2.3 Promote and Inform End Users

Armed with the ability to track changes in overall wellbeing, BeWell is able to present users with richer information and make them self-aware of their current wellbeing. BeWell presents information in two ways: i) directly, when the user interacts with the BeWell application installed on their smartphone or from a desktop using the BeWell web portal; and, ii) passively, using an ambient display rendered on the smartphone wallpaper, which is visible as the user performs typical phone operations (e.g., making a call, texting, etc.).

6.3 Monitoring and Modeling Wellbeing

A variety of health outcomes are tightly linked to everyday decisions involving sleep [13], diet [93], exercise [85] and socialization [41] patterns. The BeWell monitoring stage involves sensor-based inferences of user activities (e.g., sleeping, exercising, socialization). The wellbeing model interprets these behavioral patterns and estimates a multi-dimensional wellbeing score to make it easier for the user to understand the impact on overall wellbeing. Behavioral patterns and estimates of wellbeing are used to generate user feedback that are designed to inform users of their current wellbeing, highlighting the behavior changes needed to improve this state.

Although there are still many unanswered questions about the links between behavior and wellbeing, in BeWell we take a pragmatic approach and build an initial wellbeing model, which can be extended and refined.

Under our proof-of-concept model wellbeing is assessed for the three behaviors independently, with three simple wellbeing scores produced that ranges between 0 and 100. Every score seeks to summarize the impact of recent user behavior on overall wellbeing for just one of the three behaviors monitored. A score of 100 indicates the person is matching the accepted guidelines of performance for that behavior (e.g., averaging more than 8 hours sleep per day). Scores of 0 indicate the individual is not even attaining minimum recommended patterns of behavior. The score for each dimension is computed using an exponentially weighted average of daily scores along each dimension. In what follows, we describe the importance of these three behaviors to overall wellbeing and explain how the scores, for individual days, are computed.

6.3.1 Sleep

A body of literature exists that links sleep hygiene to a range of mental and physiological health conditions, including, affective disorders, hypertension, heart disease and the development of diabetes [13]. However, poor sleep behavior (e.g., chronic lack of sleep, erratic sleep cycles) are wide spread across the general population. People commonly exchange sleep for additional waking hours as a coping mechanism for busy lifestyles.

Research exploring sleep health effects focus both on the quantity and quality of sleep [86]. Although both of these facets are important we focus solely on monitoring sleep duration. We take a simple approach that approximates the sleep duration of users by measuring mobile phone usage patterns, as discussed in Section 6.5.2. Studies show oversleeping ("long sleep") carries similar negative health consequences to insufficient sleep [13], thus, we penalize both behaviors. BeWell computes a wellbeing score for sleep behavior within a single day using a gaussian function,

$$sleep_{day}(\texttt{HR}_{act}) = Ae^{-\frac{(\texttt{HR}_{act}-\texttt{HR}_{ideal})^2}{2(\texttt{HR}_{hi}-\texttt{HR}_{lo})^2}}$$

in which $\texttt{HR}_{act}$ is the total quantity of sleep over a 24 hour period, $\texttt{HR}_{ideal}$ is the ideal hours asleep with $\texttt{HR}_{hi}$ and $\texttt{HR}_{lo}$ being the upper and lower limits of acceptable sleep duration. Our sleep function is parameterized using a $\texttt{HR}_{ideal}$ of 7 hours with a $\texttt{HR}_{hi}$ of 9 hours and $\texttt{HR}_{lo}$ of 5 hours, these values are consistent with existing sleep studies [13].

6.3.2 Physical Activity

Benefits of physical activity, such as, lower mortality rates and cardiovascular disease are well-known but health benefits also extend to cancer and sexual dysfunction [85]. Further, a number of studies have linked exercise to improved depression, self-esteem, mood, sleep, and stress [37, 81].

Our automated wellbeing assessment of physical activity begins with common categories of end-user activity (e.g., walking, stationary, running) being recognized by smartphone sensors. The duration the user spends performing these activities is computed, allowing the Physical Activities Compendium metabolic equivalent of task (MET) value [12] to be estimated each day. Being definitive as to the ideal MET levels for an individual is difficult as mental and physical health benefits occur at different levels of activity. These MET levels are also sensitive to user characteristics, such as, existing physical fitness or particular genetic determinants. We currently rely on generic guidelines established by the Centers for Disease Control and Prevention [2] (CDC). Our daily scores of physical activity are simply a linear regression,

$$physical_{day}(\texttt{MET}_{act}) = (\texttt{MET}_{hi} - \texttt{MET}_{lo})\texttt{MET}_{act} + \texttt{MET}_{lo}$$

where $\texttt{MET}_{act}$ is the actual MET value for a user during that day, with $\texttt{MET}_{hi}$ and $\texttt{MET}_{lo}$ being calibrated by the high-end and minimum guidelines for adult aerobic activity set by the CDC. These values range between 300 and 150 minutes of moderate-intensity per week. Such aerobic activity should be accompanied by muscle-strengthening programs, ideal behavioral patterns for these programs are also available from the CDC and are included within the existing physical activity guidelines. However, we neglect this aspect of physical activity due to the inaccuracy in monitoring muscle-strengthening programs without specialized sensors (e.g., on-body sensors). Currently, BeWell users enter this behavior manually (see Section 6.4.4). In the future, we will study alternatives such as using coarse inferences based on location or sound.

6.3.3 Social Interaction

The daily social interactions of people have been shown to have impact on many dimensions of wellbeing. The connection between the availability of social support and psychological wellbeing is well established, with low levels being linked to symptoms of depression [41]. Individuals who maintain dense social connections are more likely to have resilient mental health. They tend to be able to cope with stress and often

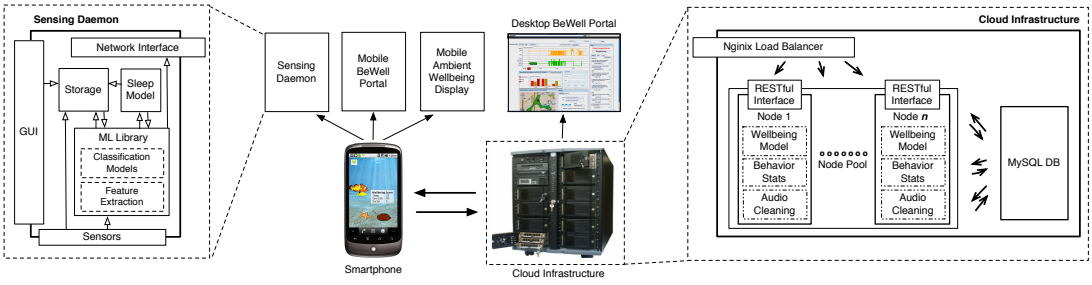


Figure 6.2: BeWell implementation, including smartphone components supported by a scalable cloud system

are better able to manage chronic illness.

Medical studies use a variety of measures to capture the social environment of a person. The development of these measures are still an active area of research. BeWell focuses on one of these metrics, social isolation, as it is more easily captured with sensors available in smartphones today. Studies of particular high-risk communities show social isolation is correlated with basic forms of human contact. For example, health deterioration exhibited in the elderly is linked with, amongst others, a decline in the frequency of human interaction (e.g., phone calls and visits with friends and relatives) [20]. In the general population, those with profound acquired hearing loss have been seen to suffer a deterioration of psychological wellbeing due to the associated communication difficulties [53]. We measure social isolation based on the total duration of ambient conversations, which are detected by inferences made using the mobile phone microphone. Insufficient medical evidence exists to parameterize this relationship. At this time we again use a wellbeing score for social interaction with a linear regression,

$$social_{day}(\texttt{DUR}_{act}) = (\texttt{DUR}_{hi} - \texttt{DUR}_{lo})\texttt{DUR}_{act} + \texttt{DUR}_{lo}$$

where $\texttt{DUR}_{act}$ is the duration of conversation detected relative to the total time the microphone is active during a single day. We determine empirically a value for $\texttt{DUR}_{hi}$, 0.35, using the mean conversation ratio of a small 10 person experiment; we also utilize this group to train our classifiers (see Section 6.5). As we lack a population in which poor wellbeing has caused atypical conversation patterns our $\texttt{DUR}_{lo}$ ratio is simply set to zero.

6.4 Implementation

In this section, we present the BeWell implementation based on the architecture discussed in Section 6.2. As illustrated in Figure 6.2 the BeWell application and system support consists of a software suite running on Android smartphones and cloud infrastructure. The software components installed on the phones include: i) Sensing Daemon, which is responsible for sensing, classification, data processing (e.g., privacy preserving audio processing) and uploading of sensor data; ii) Mobile BeWell Portal, which displays the user's behavioral patterns and the wellbeing score associated with these behaviors; and iii) Mobile Ambient Wellbeing Display, which provides an always-on visualization of the user's wellbeing scores. The BeWell Cloud Infrastructure provides storage, computation, and web access to user data via the Desktop BeWell Portal.

6.4.1 Sensing Daemon

The BeWell Sensing Daemon combines an internally developed platform-independent machine learning C library with device specific components, written in Java, that are responsible for communication, storage and the user interface. These core components are connected with a JNI bridge. The control flow of the daemon is comprised of two independently operating processes. The first process, the classification pipeline is responsible for the inference of end-user behavior. The pipeline continuously samples the phone sensors and extracts features used by classification models, which also run on the phone. The second process, asynchronously transfers inferences made by the classification models and the raw sensor data to the cloud infrastructure used by BeWell.

The classification pipeline samples three sensors, the GPS, accelerometer and microphone. Using audio sensor data the pipeline recognizes social interaction based on classifying $\{voicing, non\text{-}voicing\}$ audio segments. The accelerometer data is used to classify everyday behaviors necessary to monitor the user's physical activity including $\{driving, stationary, running, walking\}$. Inferences are made by applying a combination of feature and classification models developed in previous work, SoundSense [67] and Jigsaw [67]. Specifically, we use time (e.g., mean and variance) and frequency domain features (e.g., spectral roll-off for audio) that are classified using a Naive Bayes classifier based on a multi-variate Gaussian Model for each class. A simple Markov Model is used to apply temporal smoothing to the resulting stream of inferences.

The BeWell sleep model is based on a simple but effective approach to modeling the sleep duration of the user (see Section 6.5.2). The classification process is based on features we extract from the phone, specifically, the frequency and duration of: phone recharging events (since often people recharge their phones overnight); and the periods when the phone is either stationary or in a near silent sound environment. These features are not continuously computed rather they are extracted and computed every 24 hours. We found an effective estimate of user sleep duration could be achieved using a simple logistic regression model which incorporated a prior based on typical adult sleeping patterns.

All data is stored within independent SQLite files. These files are transferred to the cloud infrastructure with an uploading policy that emphasizes energy efficiency to minimize the impact of using the phone's batteries. Uploading only occurs if the phone has both line-power and WiFi networking available. Sensor data is uploaded to the cloud to make additional context available to end-users so they can verify, and if necessary add or edit, inferences made by the Sensing Daemon. However, people are sensitive to sharing sensor data. To address these concerns users are given complete control of the sensors being used on the phone, and inferences made, through user configuration options. Privacy controls are further augmented with automated voicing filtering. As a result, raw audio is never stored if voicing is detected within ± 5 seconds, reducing the chance that conversations are captured.

6.4.2 Mobile BeWell Portal

The Mobile BeWell Portal provides a simplified mobile phone version of the BeWell portal allowing users to track their current and historical wellbeing scores and view an automated activity diary (see Section 6.4.4). Users can view trends in the their behavioral patterns as well as their wellbeing scores.

6.4.3 Mobile Ambient Wellbeing Display

In addition to the more interactive mobile portal discussed above, BeWell also supports an ambient wellbeing display to provide constant feedback of a user's current wellbeing state. The BeWell ambient display is an animation rendered on the phone's lock-screen and wallpaper making it visible to the user whenever they glance or interact with their smartphone. Prior examples of effective mobile health applications have found that mobile phone wallpaper is an effective ambient display able to promote changes in user behavior (e.g., UbiFit Garden [28]). The BeWell ambient display

(a) (b)

Figure 6.3: Multiple wellbeing dimensions are displayed on the smartphone wallpaper. An animated aquatic ecosystem is shown with three different animals, the behavior of each is effected by changes in user wellbeing.

represents overall wellbeing as three independent scores each corresponding to sleep, activity and social interactions, as discussed in Section 6.3. Each wellbeing dimension is captured by different characters in an aquatic ecosystem, as shown in Figure 6.3. The animated activities of the orange clown fish represents the recent activity level of the user; the turtle's movement reflects a user's sleep patterns; and a school of fish indicates the sociability of the user. By quickly glancing at the screen at different times during the day the user gets a quick summary of their overall wellbeing. If the user clicks on the star fish character on the ambient display a pop-up dialog box is displayed with the numerical values that drive the animation. The relationship between the ambient display and the wellbeing scores is described below:

Turtle

Sleep patterns are captured by the turtle. The turtle sleeps on behalf of the user; that is, when the user lacks sleep the turtle sleeps for the user. When the user is getting enough sleep, the dimension score is high and the turtle comes out of its shell and walks around with varying degrees of energy.

Clown Fish

The clown fish represents the physical activity of the user. The score modifies the speed and movements of the clown fish. At low levels of physical activity the fish

Figure 6.4: The BeWell web portal provides access to an automated diary of activities and wellbeing scores.

moves slowly from across the screen. As physical activity increases the clown fish moves more vigorously and even performs summersaults and back flips when the user has high levels of activity.

School of Fish

Like the clown fish a school of yellow fish also swims across the screen. The size of the school of fish grows in proportion to the amount of social interaction by the user. In addition, the school and the clown fish swim closer together as social interaction increases.

6.4.4 Desktop BeWell Portal

BeWell users are able to access a web portal, as shown in Figure 6.4. The portal provides two primary services: i) to provide an automated diary-like visualization of the behavioral patterns and wellbeing scores of the user, which users are also able to manually edit; and, ii) to collect self-report survey data using standard validated medical surveys that monitor depression, sleep and overall wellbeing.

The diary-like visualization allows users to both browse and edit the behavioral inferences made by the BeWell Sensing Daemon. Users are able to access any sensor data collected by their phone during the day to assist with their recall of events. Users can access their location, listen to ambient (non-conversational) audio and view the raw time-series graphs of their accelerometer and microphone data. This additional

context assists users to recall, for example, when they actually woke up, or identify inference errors, such as a period of walking misclassified as jogging. Users are also able to manually add activities that are unable to be inferred by BeWell, but that may still impact wellbeing. For example, types of exercise such as swimming, which the Sensing Daemon can not recognize, or social interactions, that can be missed if the user forgets to carry their phone or turns it off. For those activities that are forms of exercise users are able to select any physical activity in the Physical Activities Compendium [12]. Wellbeing scores are updated based on both automatic sensor data from the phone and through the manual input by the user via the portal.

6.4.5 Cloud Infrastructure

The cloud infrastructure supporting the BeWell application is a pool of standard Linux-based servers. Servers offer a variety of support services to BeWell components via RESTful interfaces. These RESTful services perform the following functions: i) to store the SQLite files uploaded from mobile phones along with all user input (e.g., survey responses, activity diary edits) collected by the mobile or desktop BeWell portals; and, ii) to respond to queries for raw data, wellbeing scores or inference data which are made by all three components (viz. Sensing Daemon, Mobile and Desktop BeWell Portal).

In addition, there are continuous background daemons running on each server that perform two key tasks: i) to update the wellbeing scores of users based on incoming inferences; and, ii) to clean all the audio provided by smartphones to further protect user privacy. The cleaning process makes it difficult to inadvertently overhear conversations, which may be accidentally recorded despite the safe-guards implemented on the phone, as discussed in Section 6.4.1. The cleaning procedure segments each one second of audio into 12 chunks, where every 3rd chunk is zero-ed out. We find this is a simple and effective way to further clean the audio data beyond the voicing based protection implemented on the phone.

6.5 Evaluation

In this section, we perform experiments that validate the design and implementation of the BeWell application. First, we benchmark the resource consumption of BeWell operating on an Android Nexus One phone. We demonstrate that an *automated wellbeing monitoring app*, such as BeWell, can be deployed on an off-the-shelf smart-

BeWell Sensing Daemon		
	CPU Usage	Memory Usage
GUI only	0%	13511K
Audio sensor only	2%	14373K
Accel sensor only	2%	13917K
Audio classification	25%	14778K
Accel classification on	11%	14736K
Both Accel and Audio classification	31%	15357K
Benchmark Applications		
	CPU Usage	Memory Usage
MP3 Player	16%	27056K
Web Browser	5%	62376K

Table 6.1: Android Nexus One CPU and Memory Usage for BeWell and benchmark applications

phone. Next, we examine the accuracy of BeWell's behavior monitoring through a five person experiment. Finally, we provide results from a 19 day, 27 person real-world deployment that investigates the ability of users to understand and benefit from the multidimensional wellbeing feedback that BeWell can provide.

6.5.1 Smartphone Benchmarks

We profile the performance of the BeWell prototype application on an Android Nexus One smartphone, for CPU, battery, memory and storage. Phones in this experiment are equipped with an extended life battery (3200 mAh) and 4 GB microSD card.

Table 6.1 reports memory and CPU usage during different operational phases (e.g., sensing, inference) of the BeWell Sensing Daemon. As the table shows, CPU and memory usage vary significantly depending on the operational phase occurring within the daemon. Not surprisingly, the more burdensome phases involve inference, which encompasses sensor sampling, feature extraction and classification. Under all phases the CPU usage and memory never exceed 31% and 16 MB, respectively. Table 6.1 includes a comparison of BeWell with two widely used Android applications (viz. playing MP3 music and browsing the web). A web browser is an example of an application users will use during the day from time to time. Table 6.1 shows that BeWell and the web browser can co-exist within resource limitations. Both the MP3 player and the BeWell Sensing Daemon are background processes, and so are designed to be run for long periods of time. Table 6.1 shows the MP3 player's CPU usage when averaged over 5 minutes uses more resources than the BeWell Sensing Daemon. This suggests our daemon is competitive with existing applications in terms of resource efficiency.

We perform a five person experiment which captures battery life performance and data generation rates for BeWell. Subjects are asked to go about their normal daily

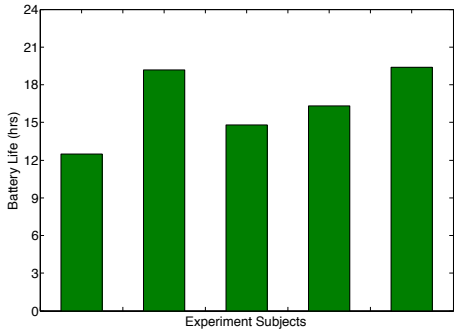


Figure 6.5: Smartphone battery life for subjects during experiment

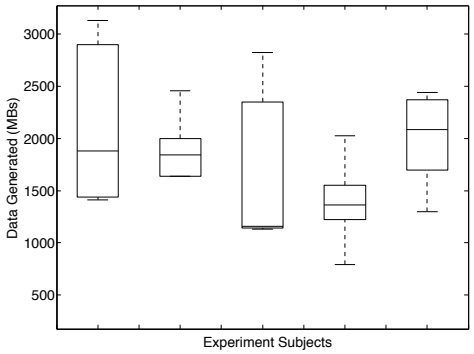


Figure 6.6: Daily data generation by subjects during one week experiment

routines. Figures 6.6 and 6.5 present per-subjects values for daily data generation and battery life, respectively. Battery life varies from user to user by 36%. This value increases to 42% for data generation. However, the results for data generation indicate that 4 GB external microSD storage is sufficient. Similarly, the experiment shows that the battery life is consistently above 15 hours, which is sufficient to run BeWell for an entire day if recharged once, very briefly during the day, as well as each night.

6.5.2 Behavioral Inference Accuracy

The accuracy of the classification process is critically important to the overall performance of BeWell. We perform a series of experiments to test the accuracy of the BeWell classification models (i.e., activity and social interaction classifiers) and sleep model. Models are trained prior to these experiments using training data from 10 people. To maintain consistency across all users for each experiment all subjects position the phone in the same body location, and attach the phone to their hip using

a holster we provide.

	Voicing	Walking	Stationary	Running
Accuracy	85.3%	90.3%	94.3%	98.1%

Table 6.2: Behavior Classification Accuracy

	RMSE	MAE
Linear Regression	2.18 hrs	1.54 hrs
Logistic regression	2.254 hrs	1.56 hrs

Table 6.3: Sleep Duration Estimate Error

Our results find that inference accuracy is inline with previous mobile phone sensing experiments [67,68] conducted with a larger number of subjects. This is expected as our classification models leverage prior work. Table 6.2 shows the accuracy for a 5 person experiment where subjects record ground-truth activity diaries for a week.

To validate our sleep model the same 5 users complete a daily survey of sleep duration, using the BeWell web portal. Table 6.3 provides the root mean square error (RMSE) and mean absolute error (MAE) when the sleep model uses either logistic or linear regression. The results show that a simple model that correlates phone recharging, movement and ambient sound context is sufficient to predict the amount of hours slept within ± 1.5 hours. Medical studies suggest that there is little difference in health outcomes for people who have sleep durations that differ by only ± 1 hour. Therefore, our coarse sleep model which is solely based on a user's phone, and not on specialized sensors worn while asleep, is adequate for a wellbeing monitoring application.

Raw inference accuracy is only one contributing factor towards how effectively wellbeing is assessed for the physical activity and social interaction of users. In the case of social interaction, for example, our voicing based approach is open to being confused by ambient sound from activities that are not actual conversations (e.g., when the user is watching TV). To test this we perform an experiment involving three people. We randomly select days during the week and require the subject to keep a detailed log of their social interactions for the entire day and evening. From this experiment we find that on average BeWell overestimates true social interaction by 14%. As part of our future work we plan to study more robust techniques for conversation recognition (e.g., using temporal characteristics of conversations such as turn taking).

Similarly, there will be inaccuracies when monitoring user activities that contribute to physical wellbeing. For example, the process of computing METs based

on just the "average" energy expended during different categories of activity will introduce noise (e.g., inter-person variation due to weight or sex). However, this is not a new problem. We use the Physical Activities Compendium [12] to compute METs as is standard practice. BeWell potentially introduces an additional error associated with correctly computing the duration of an activity due, for example, to errors in classification. To quantify this error, we compute the difference in ground-truth and estimated duration for all BeWell physical activities. We find that duration error averages 22% across all of the users in the experiment. As part of our future work, we plan to study more accurate techniques to compute energy expended during activities; for example, [61] demonstrates that mobile phones can estimate day-long calorie expenditure with 80% accuracy.

6.5.3 User Field Trial

To improve the understanding of wellbeing apps under real-world usage we deploy the prototype BeWell application to 27 people for a 19 day period. We investigate the ability of users to understand and benefit from the multi-dimensional feedback provided by the ambient display.

The study group recruited from the Dartmouth College community along with residents of the Hanover, NH region. This group is comprised by 16 men and 11 women aged between 21 and 37. Of these subjects, 9% are faculty or graduate students in the computer science department, 34% are doctors or medical researchers and the remaining 57% are students in the arts and life sciences graduate program. We request each volunteer carries a phone with the BeWell application installed continuously throughout the day. The subjects either move their mobile phone SIM card into a Nexus One or use call forwarding so they can use the study phone as their primary phone. We provide each user with a holster to clip the phone on to their belt or clothing.

Participants are randomly and uniformly divided into two groups: *multi-dimensional group* and *baseline group*. All subjects have the core BeWell software installed that tracks sleep, physical activity and social interaction. However, the baseline group do not have the ambient display and can only view the collected information via the BeWell web portal (see Figure 6.4) that summarizes the time spent in each activity as a fraction of the day. The multi-dimensional group has the ambient display.

The results presented in the remainder of this section are based on detailed analysis of the data collected using the BeWell app as well as the analysis of the exit interviews

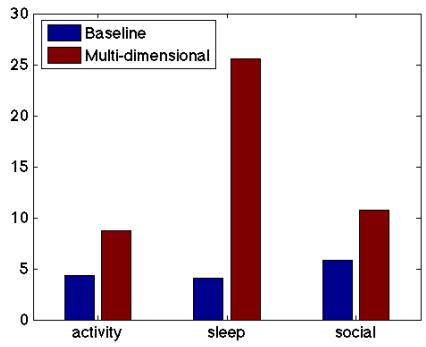


Figure 6.7: Comparison of the change in wellbeing scores during user field trial for multi-dimensional and baseline user groups.

that were conducted at the end of the deployment.

Benefit of Multi-dimensional Feedback. We measure the quantitative benefit of providing feedback along multiple dimensions by comparing changes in the wellbeing scores during the trial between the two experiment groups. To compensate for individual variation that could bias results (i.e., subject participants that have abnormally high or low wellbeing scores) we compare any changes during the study relative to a baseline average score for each person along each dimension. The baseline score is calculated from data collected during the calibration phase just before the start of the study – none of the subjects had feedback or the ambient display during the calibration phase. Figure 6.7 shows average difference in the daily score for each person during the study period, relative to their personal baseline. This figure shows a significantly greater increase in score for the multi-dimensional group compared to the baseline group. Specifically, these increases are 105% for physical activity, 88% for social interaction and 507% for sleep. Two-sample t- tests at 95% significance level indicate that multi-dimensional group is performing significantly better than baseline group ($p = 0.049$, $p < 0.01$ and $p = 0.04$ for the three dimensions respectively).

Connecting Choices with Wellbeing Consequences. Understanding why subjects within the multi-dimensional group experience sizable increases in their wellbeing scores is just as important as identifying the presence of this increase itself. We investigate if this increase was partially due to an improved ability within the subgroup to connect everyday actions to wellbeing outcomes. To test this hypothesis we perform a simple recall test during the exit interview. We show a timeline of participant wellbeing scores along different dimensions and ask the participant to annotate and explain the variations seen in the timeline. Our findings show that the subjects that had access to the multi-dimensional feedback on the phone are better

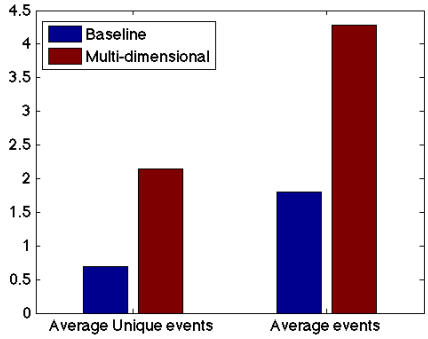


Figure 6.8: Results of user wellbeing recall test for each group in the user field trial.

	Multi-dimension group
1. User would prefer different wallpaper	-1.00
2. Multidimensional Display easy to interpret	1.50
3. Multidimensional Metrics helped keep balance	1.56
*-2: Strongly disagree, -1: Disagree, 0: Neutral, 1: Agree, 2: Strongly agree	
4. I showed others my wallpaper	83.5%
5. Animation was annoying	0.00%
*Percentage of person choose	

Table 6.4: Summary of user responses to the ambient display during the exit interview

able to connect life events to fluctuations in wellbeing. Figure 6.8 shows that the multi-dimensional group on average recalls 4.28 events per week compared to just 1.8 events for the baseline group. Similarly, the multi-dimensional group is able to recall a larger number of unique events as well. Common annotated events included: friends visiting for the weekend, change of (hospital) rotation, or pressure from a work or school deadline.

User Reactions to Multi-dimensional Ambient Display. Table 6.4 summarizes exit interview questions that survey user sentiment towards the ambient display. Participant responses indicate they have a positive reaction to the phone wallpaper as a means to visualize multidimensional wellbeing scores. A natural concern is that the use of multiple dimensions will overwhelm the user and they will not be able to easily digest the information. However, for instance, question 2 in Table 6.4 shows that people overall had little difficulty in interpreting the ambient display.

During exit interviews we discover friends and co-workers often casually ask *how is your fish today?*. Many of the participants mention that they compare scores with other participants; 83.5% of the multi-dimensional group reported that they show the display to their friends and colleagues. Overall, we are surprised to discover the amount of social activity the ambient display engendered in only a few weeks. From

Table 6.4 we find very few subjects prefer an alternative wallpaper – we believe this number may rise when deployed in a broader population. We realize that the novelty factor may have lead to this enthusiasm and this result needs to be tested with a longer term followup study. However, none of the subjects describe the visualization or the frequent animation as annoying (see Question 4 in Table 6.4).

6.6 Summary

In this chapter, we discussed the design, implementation and evaluation of BeWell, a personal health application for smartphones capable of automatically monitoring a user's overall wellbeing. BeWell is a real-time, continuous sensing application that provides easily digested feedback that promotes healthier lifestyle decisions. Our prototype implementation of BeWell demonstrates the viability of *personal wellbeing applications* using off-the-shelf smartphones. To study the real-world usage of BeWell we deploy our system to 27 people over 19 days. We find BeWell users are capable of interpreting multidimensional wellbeing scores and are responsive to feedback from an ambient display rendered on smartphone wallpaper. BeWell helps users better understand the impact to their personal wellbeing of their day to day social interaction, physical activity and sleep patterns. By providing a more complete picture of health, BeWell can empower individuals to improve their overall wellbeing and identify changes in lifestyle that can result in improvements to their quality of life.

Chapter 7

Conclusion

7.1 Summary

The field of mobile phone sensing has commonalties with many pre-existing areas of study, for example, mobile systems research, embedded sensor networking, activity recognition, HCI and context-aware pervasive computing. However, if we are to realize the potential of this new domain we will be forced to reconsider much of what these related research topics have taught us. In this thesis we have proposed *community-guided mobile phone sensing systems*, as a new direction in sensing research. This approach seeks to couple communities and their sensing systems in symbiotic arrangements that overpower the challenges presented by smartphone sensing. Our work has primarily considered how this approach relates to a single challenge, namely mobile classification. We adopt this focus as we believe the inability to perform robust and scalable mobile classification is currently the most critical impediment to the smartphone sensing revolution being realized.

The primary contribution of this dissertation is the development of techniques for intelligently integrating communities into the different stages of training and inference used in mobile classification. Smartphones have evolved to become personal companions with which we are in constant contact. Although this makes people an easily reached resource they are not necessarily simple to incorporate into a sensing system. For example, crowd-sourcing labeled sensor data from millions of everyday people can provide vast amounts of training data. However, these labels are of low-quality and prone to error. A significant contribution of this thesis is the proposal of Community-guided Learning (CGL) in Chapter 4. Under CGL labels are no longer assumed to provide accurate ground-truth. Instead, the underlying structure in the sensor-data along with unconstrained labels is exploited to intelligently group data into classes

of human behavior and context. This framework relies on similarity within data to determine when and how to split and merge contributions from different labeled categories provided by untrained contributors. CGL allows any supervised model to be trained on such data making, for the first time, crowd-sourcing labels practical for a wide range of applications.

Importantly, we demonstrate the power of not only using people directly (i.e., human-in-the-loop) but of leveraging the naturally occurring networks and behavioral patterns that exist within communities. Among its contributions, this dissertation studies a variety of the commonly occurring communities within mobile phone sensing systems. Not only do we identify a number of communities that are able to assist in mobile classification and but also develop novel algorithms and architectures to extract such potential. In Chapter 3 we study two very different communities (opportunistic and social networks). We propose Cooperative Communities, a framework that includes two complementary techniques which leverage the strengths of these communities to counter both heterogeneity in mobile device capabilities and the cost and effort in acquiring training data. Chapter 5 investigates less visible, yet still ever-present, networks comprised of clusters of people who share characteristics in one or more dimensions (e.g., behavior or physical characteristics). This chapter proposes Community Similarity Networks (CSN), a system where these logical networks of similar people are used to adapt generic classification models to the specific needs of distinct communities of users. By leveraging these networks CSN enables mobile classification to remain robust even when deployed to diverse large-scale user populations of millions of people, a challenge which we refer to as the population diversity problem. CSN is the first classification system which is able to cope with the population diversity problem.

Finally, in the previous chapter we provide a case study of a mobile sensing application which relies upon the technical breakthroughs in mobile classification made by this thesis. We present BeWell, a system able to monitor and promote the overall health and wellbeing of everyday people using existing mobile phone technology. This application represents a first-of-its-kind mobile health application that is able to consider a wide range of health outcomes. Our approach to wellbeing management is to sense along multiple dimensions of particular everyday end-user behavior which collectively exert powerful influence over the overall health of the user. This requires not only the accurate monitoring of behavioral patterns from sensor-data, a process underpinned by mobile classification, but additionally requires advances in how users are informed of the insights BeWell makes about their wellbeing and daily routines.

Using a combination of ambient displays and interactive applications we provide feedback as to the wellbeing implications for multiple behavioral patterns simultaneously. By successfully, monitoring behavior, modeling wellbeing and promoting awareness BeWell enables everyday people to assume much greater control over their own health.

7.2 End Note

The contributions made by this thesis push the boundaries of how researchers should view hybrid systems that blend the strengths of humans and machines. Within existing research into hybrid systems significant attention is, quite rightly, focused on the individual. People are directly incorporated into systems using crowd-sourcing or other human-in-the-loop approaches. However, this dissertation contends an equal amount of attention needs to be placed on how the strength of communities as a whole can be effectively leveraged. We believe both the design and operation of these systems need a broader perspective. They must still maintain awareness of the user-level interaction that takes place within the system; but at the same time recognize the system collectively interacts with another entity, a community, which has patterns of behavior and characteristics in much the same way as the people who comprise it. Sensing systems of the future will need to treat community properties, for instance, group behavior and human influence networks, as carefully as individual properties are treated today.

In this dissertation we have taken the first steps towards building mobile sensing systems that not only have such deep a understanding of the communities that use them, but are equipped to leverage these communities to more effectively serve the end user. Our detailed investigation into community-guided approaches to mobile classification has instilled in us a belief that a community-guided approach will prove valuable in addressing a much wider range of the open problems that smartphone sensing brings to the fore. We believe the examples of mobile classification and persuasion seen in this thesis represent only the beginning of how communities can impact the operation of mobile sensing systems.

By exploring sensing systems that have community awareness this thesis has exposed a number of open research questions. Uncovering additional aspects of mobile sensing which will benefit from adopting community-guided techniques is only one of these. From our investigation we find community-guided methods tend to be tied both to the form of community leveraged and the technical challenge being overcome. Perhaps the key unresolved question is if there exists a generalized model for exploit-

ing groups that will span different mobile sensing domains. The complexities and shear variety of community interactions within large-scale populations are extensive, what are the salient issues that a generalized model must encompass? At this early stage we believe these to be: i) recognizing communities within the wider population, ii) determining the collective characteristics of these communities, and iii) identifying the various connections that exist between them. To make progress we will need new community-guided systems which operate at two at both the micro-scale (personal) and macro-scale (community). Systems which are cognizant of the individual end-user, while at the same time operate with an awareness of the community; understanding how these two scales interact with each other and impact the system and algorithmic design of sensing systems will be important for future community-guided sensing research to consider.

Collectively, the individual chapters of this thesis provide a single clear powerful example of the potential for *community-guided mobile phone sensing systems*. Our work shows how community involvement and awareness can enable mobile classification to be more robust and scalable. Through the technical contributions of Cooperative Communities, CGL, CSN and BeWell we have laid the foundation necessary for further exploration. We hope this dissertation has provided both the impetus for additional study and acts as a useful guide to researchers seeking to identify the alternative avenues where communities can further impact the evolution of mobile phone sensing systems.

Bibliography

[1] Amazon mechanical turk. http://www.mturk.com.

[2] Center for Disease Control and Prevention. http://www.cdc.gov.

[3] Garbage watch. http://garbagewatch.com/.

[4] Google nexus one. http://www.google.com/phone/detail/nexus-one.

[5] Intel / UC Berkeley. Urban Atmospheres. http://www.urban-atmospheres.net/.

[6] UC Berkeley / Nokia / NAVTEQ. Mobile Millennium. http://traffic.berkeley.edu/.

[7] Nike+. http://www.apple.com/ipod/nike/run.html.

[8] Sf-36.org. a community for measuringhealth outcoming using sf tools. http://www.sf-36.org/tools/SF36.shtml.

[9] Amazon Elastic Cloud Computing. http://aws.amazon.com/ec2.

[10] Tarek Abdelzaher, Yaw Anokwa, Peter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Samuel Madden, and Jim Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.

[11] E. Acuna and C. Rodriguez. The treatment of missing values and its effect in the classifier accuracy. In *Classification, Clustering and Data Mining Applications*, pages 639–648, 2004.

[12] B. E. Ainsworth, W. L. Haskell, M. C. Whitt, M. L. Irwin, A. M. Swartz, S. J. Strath, W. L. O'Brien, D. R. Bassett, K. H. Schmitz, P. O. Emplaincourt, D. R. Jacobs, and A. S. Leon. Compendium of Physical Activities: An Update of Activity Codes and MET Intensitiess. *Medicine and science in sports and exercise*, 32(9 Suppl), September 2000.

[13] GG Alvarez and NT Ayas. The impact of daily sleep duration on health: a review of the literature. *Progress in cardiovascular nursing*, 19(2):56, 2004.

[14] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[15] Daniel Ashbrook and Thad Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[16] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[17] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2004.

[18] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[19] Ulf Blanke and Bernt Schiele. Daily routine recognition through activity spotting. In *LoCA '09: Proceedings of the 4th International Symposium on Location and Context Awareness*, pages 192–206, Berlin, Heidelberg, 2009. Springer-Verlag.

[20] D.G. Blazer. Social support and mortality in an elderly community population. *American journal of epidemiology*, 115(5):684, 1982.

[21] N. Bulusu, C.T. Chou, S. Kanhere, Y. Dong, S. Sehgal, D. Sullivan, and L. Blazeski. Participatory sensing in commerce: Using mobile camera phones to track market price dispersion. In *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08)*, 2008.

[22] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *In: Workshop on World-Sensor-Web (WSW06): Mobile Device Centric Sensor Networks and Applications*, pages 117–134, 2006.

[23] Andrew Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu, Matthew K. Mukerjee, Mashfiqui Rabbi, and Rajeev D.S. Raizada. Neurophone: brain-mobile phone interface using a wireless eeg headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 3–8, New York, NY, USA, 2010. ACM.

[24] Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, and Ronald A. Peterson. People-centric urban sensing. In *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*, page 18, New York, NY, USA, 2006. ACM.

[25] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harri son, Bruce Hemingway, Jeffrey Hightower, Pedja Klasnja, Karl Koscher, Anthony L aMarca, James A. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Re a, and Danny Wyatt. The mobile sensing platform: An embedded system for activity recognition. *Appears in IEEE Pervasive Magazine - Special Issue on Activity-Based Computing*, 7(2):32–41, 2008.

[26] Tanzeem Choudhury and Alex Pentland. Sensing and modeling human networks using the sociometer. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, page 216, Washington, DC, USA, 2003. IEEE Computer Society.

[27] Collaborative machine learning. In *From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments*, pages 173–182, 2005.

[28] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in the wild: a field trial of ubifit garden. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1797–1806, New York, NY, USA, 2008. ACM.

[29] Landon P. Cox, Angela Dalton, and Varun Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 233–245, New York, NY, USA, 2007. ACM.

[30] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, C. Ranveer, and P. Bahl. MAUI: Making Smartphones Last Longer with Code Offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10)*. ACM, 2010.

[31] Tathagata Das, Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, and Asankhaya Sharma. Prism: platform for remote sensing using smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10)*. ACM, 2010.

[32] L. Dipietro, C.J. Caspersen, A.M. Ostfeld, and E.R. Nadel. A survey for assessing physical activity among older adults. *Medicine and Science in Sports and Exercise*, 25:628–628, 1993.

[33] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

[34] Shane B. Eisenman, Nicholas D. Lane, and Andrew T. Campbell. Techniques for improving opportunistic sensor networking performance. In Sotiris E. Nikoletseas, Bogdan S. Chlebus, David B. Johnson, and Bhaskar Krishnamachari, editors, *DCOSS*, volume 5067 of *Lecture Notes in Computer Science*, pages 157–175. Springer, 2008.

[35] Pedro Ferreira, Pedro Sanches, Kristina Höök, and Tove Jaensson. License to chill!: how to empower users to cope with stress. In *NordiCHI '08: Proceedings of the 5th Nordic conference on Human-computer interaction*, pages 123–132, New York, NY, USA, 2008. ACM.

[36] B. J. Fogg. Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002(December):2, 2002.

[37] K.R. Fox. The influence of physical activity on mental well-being. *Public Health Nutrition*, 2(3a):411–418, 1999.

[38] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.

[39] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A. Landay. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1043–1052, New York, NY, USA, 2009. ACM.

[40] Raghu K. Ganti, Nam Pham, Yu-En Tsai, and Tarek F. Abdelzaher. Poolview: stream privacy for grassroots participatory sensing. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 281–294, New York, NY, USA, 2008. ACM.

[41] L.K. George, D.G. Blazer, D.C. Hughes, and N. Fowler. Social Support and the Outcome of Major Depression. *The British Journal of Psychiatry*, 154(4):478, 1989.

[42] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[43] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. In *WMASH '03: Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 46–55, New York, NY, USA, 2003. ACM.

[44] Mark A. Hanson, Harry C. Powell Jr., Adam T. Barth, Kyle Ringgenberg, Benton H. Calhoun, James H. Aylor, and John Lach. Body area sensor networks: Challenges and opportunities. *Computer*, 42:58–65, 2009.

[45] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith, and Jeff Hughes. Learning and recognizing the places we go. In *Proceedings of the Seventh International Conference on Ubiquitous Computing (Ubicomp 2005)*, Lecture Notes in Computer Science, pages 159–176. Springer-Verlag, September 2005.

[46] Richard Honicky, Eric A. Brewer, Eric Paulos, and Richard White. N-smarts: networked suite of mobile atmospheric real-time sensors. In *NSDR '08: Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*, pages 25–30, New York, NY, USA, 2008. ACM.

[47] R. Hurling, M. Catt, M. De Boni, B.W. Fairley, T. Hurst, Murray P., A. Richardson, and J.S. Sodhi. Using Internet and Mobile Phone Technology to Deliver an Automated Physical Activity Program: Randomized Controlled Trial. *Journal of Medical Internet Research*, 9(2):e7, 2007.

[48] T. Huynh, U. Blanke, and B. Schiele. Scalable recognition of daily activities with wearable sensors. In *3rd International Symposium on Location- and Context-Awareness (LoCA)*, 2007.

[49] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *UbiComp*, pages 10–19, 2008.

[50] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations. In *WMASH '04: Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 110–118, New York, NY, USA, 2004. ACM Press.

[51] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: Security challenges for the new paradigm. In *Proc. of 1st International Conference on Communication Systems and Networks, COMNETS, Bangalore*. Citeseer, 2009.

[52] Ashish Kapoor and Eric Horvitz. Experience sampling for building predictive user models: a comparative study. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 657–666, New York, NY, USA, 2008. ACM.

[53] J.F. Knutson and C.R. Lansing. The relationship between communication problems and psychological difficulties in persons with profound acquired hearing loss. *Journal of Speech and Hearing Disorders*, 55(4):656, 1990.

[54] Andreas Krause, Eric Horvitz, Aman Kansal, and Feng Zhao. Toward community sensing. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 481–492, Washington, DC, USA, 2008. IEEE Computer Society.

[55] John Krumm and Ken Hinckley. The nearme wireless proximity server. In Nigel Davies, Elizabeth D. Mynatt, and Itiro Siio, editors, *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 283–300. Springer, 2004.

[56] Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T. Campbell. Urban sensing systems: opportunistic or participatory? In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 11–16, New York, NY, USA, 2008. ACM.

[57] Nicholas D. Lane, Hong Lu, Shane B. Eisenman, and Andrew T. Campbell. Cooperative techniques supporting sensor-based people-centric inferencing. In Jadwiga Indulska, Donald J. Patterson, Tom Rodden, and Max Ott, editors, *Pervasive*, volume 5013 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2008.

[58] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *Comm. Mag.*, 48:140–150, September 2010.

[59] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In Kenneth P. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron J. Quigley, editors, *Pervasive*, volume 3968 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2006.

[60] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI*, pages 766–772, 2005.

[61] Jonathan Lester, Carl Hartung, Laura Pina, Ryan Libby, Gaetano Borriello, and Glen Duncan. Validated caloric expenditure estimation using a single body-worn sensor. In *Proceedings of the 11th international conference on Ubiquitous computing*, Ubicomp '09, pages 225–234, New York, NY, USA, 2009. ACM.

[62] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 787–794. MIT Press, Cambridge, MA, 2006.

[63] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. Rob. Res.*, 26(1):119–134, 2007.

[64] James J. Lin, Lena Mamykina, Silvia Lindtner, Gregory Delajoux, and Henry B. Strub. Fish'n'steps: Encouraging physical activity with an interactive computer game. In Paul Dourish and Adrian Friday, editors, *Ubicomp*, volume 4206 of *Lecture Notes in Computer Science*, pages 261–278. Springer, 2006.

[65] Jie Liu. Subjective sensing: Intentional awareness for personalized services. *NSF Workshop on Future Directions in Networked Sensing Systems*, November 2009.

[66] B. Longstaff, S. Reddy, and D. Estrin. Improving Activity Classification for Health Applications on Mobile Devices using Active and Semi-Supervised Learning. In *Proceedings of ICST Conference on Pervasive Computing Technologies for Healthcare*, 2010.

[67] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. Soundsense: Scalable Sound Sensing for People-centric Applications on Mobile Phones. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178, New York, NY, USA, 2009. ACM.

[68] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *SenSys '10: Proceedings of the 8th international conference on Embedded networked sensor systems*, 2010.

[69] M. Mahdaviani and T. Choudhury. Fast and scalable training of semi-supervised crfs with application to activity recogni tion. In *In Proc. of the Advances of Neural Information Processing Systems 20 (NIPS 2007)*, 2007.

[70] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[71] E. Miluzzo, C.T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A.T. Campbell. Darwin Phones: The Evolution of Sensing and Inference on Mobile Phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10)*. ACM, 2010.

[72] Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, and Andrew T. Campbell. Cenceme - injecting sensing presence into social networking applications. In Gerd Kortuem, Joe Finney, Rodger Lea, and Vasughi Sundramoorthy, editors, *EuroSSC*, volume 4793 of *Lecture Notes in Computer Science*, pages 1–28. Springer, 2007.

[73] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, New York, NY, USA, 2008. ACM.

[74] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. Eyephone: activating mobile phones with your eyes. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 15–20, New York, NY, USA, 2010. ACM.

[75] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Péter Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 55–68, New York, NY, USA, 2009. ACM.

[76] Sense Networks. Website, 2008. `http://www.sensenetworks.com/`.

[77] Nokia. SensorPlanet. `http://www.sensorplanet.org/`.

[78] Nokia. *Workshop on Large-Scale Sensor Networks and Applications*. February 03-06 2005, Kuusamo, Finland.

[79] R. Norris, D. Carroll, and R. Cochrane. The Effects of Physical Activity and Exercise Training on Psychological Stress and Well-being in an Adolescent Population. *Journal of Psychosomatic Research*, 36(1):55–65, 1992.

[80] N. C. Oza and Stuart Russell. Online bagging and boosting. In *AISTAT*, pages 105–112, 2001.

[81] R.S. Paffenbarger Jr, R. Hyde, A.L. Wing, and C. Hsieh. Physical activity, all-cause mortality, and longevity of college alumni. *New England journal of medicine*, 314(10):605–613, 1986.

[82] K. Patrick, F. Raab, M.A. Adams, L. Dillon, M. Zabinski, C.L. Rock, W.G. Griswold, and G.J. Norman. A Text Message–based Intervention for Weight Loss: Randomized Controlled Trial. *Journal of Medical Internet Research*, 11(1), 2009.

[83] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaoka i Wang, Dieter Fox, and Henry Kautz. Opportunity knocks: A system to provide cognitive assistance wi th transportation services. In *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 433–450, Berlin / Heidelberg, 2004. Springer.

[84] D. Peebles, H. Lu, N.D. Lane, T. Choudhury, and A.T. Campbell. Community-guided learning: Exploiting mobile sensor users to model human behavior. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI-10)*, 2010.

[85] F.J. Penedo and J.R. Dahn. Exercise and well-being: a review of mental and physical health benefits associated with physical activity. *Current Opinion in Psychiatry*, 18(2):189, 2005.

[86] June J. Pilcher, Douglas R. Ginter, and Brigitte Sadowsky. Sleep quality versus sleep quantity: Relationships between sleep and measures of health, well-being and sleepiness in college students. *Journal of Psychosomatic Research*, 42(6):583 – 596, 1997. Nocturnal Penile Tumescence: Measurement and Reasearch.

[87] Ming-Zher Poh, Kyunghee Kim, Andrew D. Goessling, Nicholas C. Swenson, and Rosalind W. Picard. Heartphones: Sensor earphones and mobile application for non-obtrusive health monitoring. *Wearable Computers, IEEE International Symposium*, 0:153–154, 2009.

[88] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43:51–58, May 2000.

[89] Bodhi Priyantha, Dimitrios Lymberopoulos, and Jie Liu. Littlerock: Enabing energy efficient continuous sensing on moble phones. Technical Report MSR-TR-2010-14, Microsoft Research, 2010.

[90] Rajib Rana, Chun Tung Chou, Salil Kanhere, Nirupama Bulusu, and Wen Hu. Ear-phone: An end-to-end participatory urban noise mapping. In *IPSN '10: Proceedings of the 9th international conference on Information processing in sensor networks*, New York, NY, USA, 2010. ACM.

[91] S. Reddy, D. Estrin, and M. Srivastava. Recruitment Framework for Participatory Sensing Data Collections. *Proceedings of Eighth International Conference on Pervasive Computing*, 2010.

[92] Oriana Riva and Cristian Borcea. The urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2):41–49, 2007.

[93] P. Rozin. The meaning of food in our lives: a cross-cultural perspective on eating and well-being. *Journal of Nutrition Education and Behavior*, 37:S107–S112, 2005.

[94] A. Schmidt, K. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced interaction in context. In *HandHeld and Ubiquitous Computing*, pages 89–101. Springer, 1999.

[95] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[96] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F.L. Wong. Sensay: A context-aware mobile phone. In *Proceedings of the 7th IEEE international Symposium on Wearable Computers*, page 248. Citeseer, 2003.

[97] Thad Starner. *Wearable Computing and Conetextual Awareness*. PhD thesis, MIT Media Laboratory, April 30 1999.

[98] Maja Stikic and Bernt Schiele. Activity recognition from sparsely labeled data using multi-instance learning. In *Proceedings of LoCA '09*, pages 156–173, Berlin, Heidelberg, 2009. Springer-Verlag.

[99] Maja Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*, pages 81–88, Washington, DC, USA, 2008. IEEE Computer Society.

[100] Maya Stikic, Kristof Van Laerhoven, and Bernt Schiele. Exploring semi-supervised and active learning for activity recognition. In *In Proc. of IEEE International Symposium on Wearable Computing*, 2008.

[101] David Minnen Thad, David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering characteristic actions from on-body sensor data. In *In Proc. of IEEE International Symposium on Wearable Computing*, pages 11–18, 2006.

[102] Arvind Thiagarajan, Lenin Ravindranath Sivalingam, Katrina LaCurts, Sivan Toledo, Jakob Eriksson, Samuel Madden, and Hari Balakrishnan. VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones. In *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, CA, November 2009.

[103] Antonio Torralba and Kevin P. Murphy. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):854–869, 2007. Senior Member-Freeman, William T.

[104] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[105] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of CHI '04*, pages 319–326, New York, NY, USA, 2004. ACM.

[106] Luis von Ahn, Benjamin Maurer, Colin Mcmillen, David Abraham, and Manuel Blum. re-CAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, pages 1160379+, August 2008.

[107] Y. Wang, J. Lin, M. Annavaram, Q.A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 179–192. ACM, 2009.

[108] Woodrow W. Winchester, III. Cover story catalyzing a perfect storm: mobile phone-based hiv-prevention behavioral interventions. *interactions*, 16(6):6–12, 2009.

[109] Ian H. Witten and Eibe Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[110] C. Tomasi Y. Rubner and L. J. Guibas. The earth movers distance as a metric for image retrieval. *IJCV*, 40(2):99–121, 2000.

[111] H.K. Yaggi, A.B. Araujo, and J.B. McKinlay. Sleep duration as a risk factor for the development of type 2 diabetes. *Diabetes Care*, 29(3):657, 2006.

[112] T. Yan, V. Kumar, and D. Ganesan. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 77–90. ACM, 2010.

[113] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In *UbiComp*, pages 312–321, 2008.

[114] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 791–800, 2009.

[115] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen. Discovering personally meaningful places: An interactive clustering approach. *ACM Trans. Inf. Syst.*, 25(3):12, 2007.

[116] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool publishers, 2009.

[117] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

Appendix: Refereed Publications as a Ph.D. Candidate

My refereed publications as a Ph.D. candidate are listed below. Work in preparation and technical reports are omitted.

Journal Publications

Hong Lu, Nicholas D. Lane, Shane B. Eisenman, Andrew T. Campbell, *"Bubble-Sensing: Binding Sensing Tasks to the Physical World"* In Journal of Pervasive and Mobile Computing, 2010.

Shane Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald Peterson, Gahng Seop Ahn, and Andrew T. Campbell, *"BikeNet: A Mobile Sensing System for Cyclist Experience Mapping"*, ACM Transactions on Sensor Networks (TOSN), Vol. 6, n. 1, December 2009.

Magazine Publications

Nicholas D. Lane, Ye Xu, Hong Lu, Shane Eisenman, Tanzeem Choudhury, Andrew Campbell, *"Cooperative Communities (CoCo): Exploiting Social Networks for Large-scale Modeling of Human Behavior"*. In IEEE Pervasive Computing, November, 2011.

Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, Andrew T. Campbell, *"A Survey of Mobile Phone Sensing"*. In IEEE Communications Magazine, September 2010.

Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristof Fodor, Gahng-Seop Ahn, *"People Power - The Rise of People-Centric Sensing"*, In IEEE Internet Computing Special Issue on Mesh Networks, 2008.

Conference/Workshop Publications

Nicholas D. Lane, Mashfiqui Mohammod, Mu Lin, Xiaochao Yang, Hong Lu, Shahid Ali, Afsaneh Doryab, Ethan Berke, Tanzeem Choudhury, Andrew T. Campbell, *"BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing"*. In Proc. of International ICST Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Dublin, Ireland, 23-26 May 2011.

Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, Andrew T. Campbell, *"The Jigsaw Continuous Sensing Engine for Mobile Phone Applications"*. In Proc. 8th ACM Conference on Embedded Networked Sensor Systems (SenSys 2010), Zurich, Switzerland, November 3-5, 2010.

Emiliano Miluzzo, Michela Papandrea, Nicholas D. Lane, Hong Lu, Andrew T. Campbell, *"Pocket, Bag, Hand, etc. - Automatically Detecting Phone Context through Discovery"*. In Proc. of First International Workshop on Sensing for App Phones (PhoneSense'10), Zurich, Switzerland, November 2, 2010.

Nicholas D. Lane, Dimitrios Lymberopoulos, Feng Zhao, Andrew T. Campbell, *"Hapori: Context-based Local Search for Mobile Phones using Community Behavioral Modeling and Similarity"*. In Proc. of 12th International Conference on Ubiquitous Computing (Ubicomp 2010) , Copenhagen, Denmark, September 2010.

Emiliano Miluzzo, Nicholas D. Lane, Hong Lu, Andrew T. Campbell, *"Research in the App Store Era: Experiences from the CenceMe App Deployment on the iPhone"*. In Proc. of The First International Workshop Research in the Large: Using App Stores, Markets, and other wide distribution channels in UbiComp research, September 26, 2010, Copenhagen, Denmark.

Daniel Peebles, Hong Lu, Nicholas D. Lane, Tanzeem Choudhury, Andrew T. Campbell, *"Community-Guided Learning: Exploiting Mobile Sensor Users to Model Human Behavior"*. In Proc. of 24th AAAI Conference on Artificial Intelligence (AAAI '10), Atlanta, Georgia, USA, July 11-15, 2010.

Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, Andrew T. Campbell, *"SoundSense: Scalable Sound Sensing for People-Centric Sensing Applications on Mobile Phones"*, In Proc. of 7th ACM Conference on Mobile Systems, Applications, and Services (MobiSys '09), Krakov, Poland, June 22-25, 2009.

Emiliano Miluzzo, Nicholas D. Lane, Kristof Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, Andrew T. Campbell, *"Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application"*, In Proc. of Seventh ACM Conference on Embedded Network Sensor Systems (Sensys 2008), Raleigh, NC, Nov. 5 - Nov. 7, 2008.

Emiliano Miluzzo, James M. H. Oakley, Hong Lu, Nicholas D. Lane, Ronald A. Peterson, Andrew T. Campbell, *"Evaluating the iPhone as a Mobile Platform for People-Centric Sensing Applications"*, In Proc. of Intl Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08). Raleigh, NC, USA, Nov. 4, 2008.

Emiliano Miluzzo, Nicholas D. Lane, Andrew T. Campbell, Reza Olfati-Saber, *"CaliBree: a Self-Calibration System for Mobile Sensor Networks"*, In Proc. of International Conference on Distributed Computing in Sensor Networks (DCOSS 2008), Santorini Island, Greece, June 11-14, 2008.

Shane B. Eisenman, Nicholas D. Lane, and Andrew T. Campbell, *"Techniques for Improving Opportunistic Sensor Networking Performance"*, In Proc. of International Conference on Distributed Computing in Sensor Networks (DCOSS 2008), Santorini Island, Greece, June 11-14, 2008.

Mirco Musolesi, Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, Tanzeem Choudhury, Andrew T. Campbell, *"The Second Life of a Sensor: Integrating Real-world Experience in Virtual Worlds using Mobile Phones"*, In Proc. of Fifth Workshop on Embedded Networked Sensors (HotEmNets 2008), June 2008, Charlottesville, Virginia, USA.

Nicholas D. Lane, Hong Lu, Shane B. Eiseman and Andrew T. Campbell, *"Cooperative Techniques Supporting Sensor-based People-centric Inferencing"*, In Proc. of Pervasive 2008, Sydney, Australia, May 19-22, 2008.

Hong Lu, Nicholas D. Lane, Shane B. Eisenman, Andrew T. Campbell, *"Bubble-Sensing: A New Paradigm for Binding a Sensing Task to the Physical World using Mobile Phones"*, In Proc. of International Workshop on Mobile Devices and Urban Sensing, St. Louis, April 21, 2008.

Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, Andrew T. Campbell, *"Urban Sensing Systems: Opportunistic or Participatory?"*, In Proc. of Ninth Workshop on Mobile Computing Systems and Applications (HotMobile 2008), Silverado Resort, Napa Valley, CA, USA,

Feb. 25-26, 2008.

Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, Andrew T. Campbell, *"CenceMe - Injecting Sensing Presence into Social Networking Applications (Invited paper)"* In Proc. of Second European Conference on Smart Sensing and Context (EuroSSC 2007), Lake District, UK, October 23-25, 2007.

Nicholas D. Lane, Shane B. Eisenman, Emiliano Miluzzo, Mirco Musolesi, Andrew T. Campbell, *"Urban Sensing: Opportunistic or Participatory?"*, In Proc. of First Workshop on Sensing on Everyday Mobile Phones in Support of Participatory Research, Sydney, Australia, November 6, 2007.

Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell, *"The BikeNet Mobile Sensing System for Cyclist Experience Mapping"*, In Proc. of Sixth ACM Conference on Embedded Network Sensor Systems (Sensys 2007), Sydney Australia, November 6 - 9th 2007.

Nicholas D. Lane, Hong Lu and Andrew T. Campbell, *"Ambient Beacon Localization: Using Sensed Characteristics of the Physical World to Localize Mobile Sensors"*, In Proc. of Fourth IEEE Workshop on Embedded Networked Sensors (EmNets 2007), 25-26 June, 2007 Cork, Ireland.

Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell, *"MetroSense Project: People-Centric Sensing at Scale"*. In Proc. of First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications (WSW'2006), Boulder CO, Oct 31, 2006.

Emiliano Miluzzo, Nicholas D. Lane, and Andrew T. Campbell, *"Virtual Sensing Range (Poster Abstract)"*, In Proc. of Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006), Boulder, Colorado, USA, November 1-3, 2006.

Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald Peterson, *"People-Centric Urban Sensing (Invited Paper)"*, In Proc. of Second ACM/IEEE Annual International Wireless Internet Conference (WICON 2006), Boston, Massachusetts, USA, August 2-5, 2006.

Nicholas D. Lane and Andrew T. Campbell, *"The Influence of Microprocessor Instructions on the Energy Consumption of Wireless Sensor Networks"*, In Proc. of Third IEEE Workshop on Embedded Networked Sensors (EmNets 2006), Cambridge, Massachusetts, USA May 30-31, 2006.