

Practical Context Awareness: Measuring and Utilizing the Context Dependency of Mobile Usage

Ahmad Rahmati, *Member, IEEE*, Clayton Shepard, *Student Member, IEEE*, Chad Tossell, Lin Zhong, *Member, IEEE*, Philip Kortum

Abstract— Context information brings new opportunities for efficient and effective applications and services on mobile devices. A wide range of research has exploited context dependency, i.e. the relations between context(s) and the outcome, to achieve significant, quantified, performance gains for a variety of applications and services. These works typically have to deal with the challenges of multiple context sources leading to a sparse training data set, and the challenges of energy hungry context sensors. Often, they address these challenges in an application specific and ad-hoc manner. We liberate mobile application designers and researchers from these burdens by providing a methodical approach to these challenges. In particular, we 1) define and measure the context-dependency of three principal types of mobile usage (visited websites, phone calls, and app usage) in an application agnostic yet practical manner, providing insight into the performance of potential application. 2) Address the challenge of data sparseness when dealing with multiple context sources in a systematic manner. 3) Present SmartContext to address the energy challenge by automatically selecting among context sources while ensuring a minimum accuracy for each estimation. Our analysis and findings are based on one year of usage and context traces collected in real-life settings from 24 iPhone users. We present findings regarding the context dependency of three types of mobile usage from 24 users, yet our methodology and the lessons we learn can be readily extended to other types of usage as well as system resources. Our findings guide the development of context aware systems, and highlight the challenges and expectations regarding the context dependency of mobile usage.

Index Terms—Mobile Computing, Mobile Applications, Human Factors.

1. INTRODUCTION

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

Context has in the past been widely exploited to provide more usable mobile applications and services, such as content adaptation [3, 4], user interaction [5], and information delivery [6, 7]. Context has also been widely exploited to provide enhanced system efficiency and performance, such as for energy management [8, 9] and network selection [10]. These designs exploit the context dependency of mobile usage and mobile resources for specific purposes, and show significant, quantified, performance gains.

Context aware systems often have to deal with two fundamental challenges. First: dealing with multiple sources of context is challenging; due to the curse of dimensionality [11], simply treating them as a multidimensional vector results in a sparse training set. Second: liberal application of context can quickly drain the devices battery, as some context sensors are extremely energy hungry. To address the sparseness challenge, existing work often limit the number

of context sources, e.g. to one [8] or two [9], and/or employ ad-hoc or expert solutions to combine multiple sources of context, e.g. [10]. To address the energy challenge, they often employ ad-hoc schemes along one or more of these lines: reducing the frequency of accessing costly context [12-14], avoiding them altogether [10, 12, 13], or substituting them with other context [15-18].

Ad-hoc and application specific approaches towards these challenges mean that the designers need to design and evaluate a new solution for every context-based system. Furthermore, before designing and evaluating their application or service, its designers can only guess its performance outcome. Our work is liberating in this regard. We provide a methodological solution for using multiple and various sources of context, while managing their energy costs. We provide a formal yet practical definition of context dependency, which provides insight into the performance of applications while remaining application agnostic. We measure the context dependency of three principal types of mobile usage using unprecedented real-life context and usage traces collected from 24 iPhone users over one year. The mobile usage we focus on are visited websites, phone calls, and app usage¹. We utilize context information from sensors built into the phone (i.e. real-time clock, Cell ID, Accelerometer, and GPS), as well as the phone's last known usage state (i.e. application, web, and phone use). Yet, our methodology and the lessons learned can be extended to other context and usage. In particular, we make four major contributions towards quantifying and measuring the context dependency of mobile usage: First, in Section 3, we identify *estimation accuracy* based on maximum a posteriori (MAP) estimation as an application

- A. Rahmati is with Broadcom Corporation, 190 Mathilda Pl. Sunnyvale, CA. E-mail: ahmad@rahmati.com.
- C. Shepard, L. Zhong, and P. Kortum are with Rice University, Houston, TX 77005. E-mail: {cws, lzhong, pkortum}@rice.edu.
- C.C. Tossell is with the USAF Research Lab, Wright-Patterson Air Force Base, 2620 Q St. Bldg. 852, OH 45433. E-mail: chadtossell@yahoo.com.

¹ Note that we use the words app and application differently. App refers to applications that are installed on the phones, either built-in or obtained from the App Store. Application refers to its more general meaning, i.e. use
xxx-xxx/0x/\$xx.00 © 200x IEEE

case. Similarly, application agnostic means not dedicated to a single service or purpose.

Table 1. Data samples collected from the 24 users, during one year of logging

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

agnostic yet practical, measure for context dependency, In contrast with other theoretical metrics that are applicable to multinomial data, such as entropy as a measure of uncertainty and pseudo R square as a measure of correlation, estimation accuracy provides practical insight into the performance of many potential applications, while remaining application agnostic. To allow the efficient calculation of posterior probabilities, we present and compare the performance of several forms categorizing and binning of context measurements into a limited number of categories, for both continuous and discrete context sources. Furthermore, we address the challenge of data sparseness when dealing with multiple sources of context by comparing classifier combination methods.

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

Third, in Section 5, we present SmartContext, a framework to dynamically or statically optimize the cost / accuracy tradeoffs of context awareness, while ensuring a minimum accuracy for each estimation event. SmartContext takes advantage of the classifier combination algorithms we have explored that have little overhead. We show that by utilizing energy hungry context only at uncertain times, SmartContext can achieve an estimation accuracy within 1% of the maximum possible accuracy, while significantly reducing energy costs by 60% or more.

Fourth, in Section 6, we present and evaluate several sample applications that benefit from context dependency of mobile usage. These applications highlight the practical value of estimation accuracy as a measure of context dependency, and attest to the effectiveness of context for estimating usage. Our best performing methods, i.e. using Supervised Binning and Bayesian combination, consistently outperform common non-context-based methods.

2. DATA COLLECTION

Studying context dependency can be extremely challenging, as it needs a large trace collected in real life user studies. In this section, we describe the methodology used to collect and analyze the usage and context data from 24 iPhone users. We have already presented the details of the data collection in [19]. In this section, we provide information relevant to this study.

2.1. Field Study Participants

The 24 participants were studied continuously for one year, from February 2010 to February 2011. All of them were undergraduate students at a small private university, located in a major metropolitan area of the USA. In general, they were representative of college students in terms of age (average age: 19.7, deviation: 1.1) and gender. They lived on campus and had a PC or laptop at their residence, in addition to access to the university's computing labs.

As compensation, each participant received a free iPhone as well as free service throughout the duration of the study, including 450 voice call minutes per month, unlimited data, and unlimited SMS. We helped all participants port their phone numbers to and they were required to use the outfitted iPhones as their primary phone. They were not given specific instructions on how to use the device, other than to use it as they would normally use their phone.

2.2. Logger Design and Implementation

While extensive logging of PC usage has been reported in past literature, privacy concerns and battery lifetime limitations, have limited the scope of mobile phone based studies. Indeed, privacy concerns and/or significantly reduced battery lifetime is likely to impact usage, thus the usage data would not accurately reflect real life user behavior [20, 21]. Our study mitigates these concerns by limiting energy consumption and addressing user privacy concerns through one way hashing and on device data processing, as well as by *partitioning*, i.e. dividing the research team so that the data analysis and logger development team do not know or directly interact with the participants, in order to avoid linking data to the actual users. The key component of the study is an in-device, programmable logging software that collects iPhone usage and context *in situ*. To run the iPhone logger continuously in the background, we had to jailbreak the iPhones. The main logger daemon is written as a bash shell script and utilizes components written in various languages, including C, Perl, awk, SQL, and objective C, altogether comprising ~2000 lines of code.

The logger records a plethora of context information. For this work we focus on logs regarding usage and context. The visited websites, app used, and phone calls are recorded by the phone's operating system, and our logger piggy-backs on the phone's logs by periodically recording them. Further, whenever the phone's CPU is not asleep, at 15 minute intervals, the logger records the GPS location, cell ID, and a 15 second recording from the accelerometer at 25 Hz. The GPS location data is collected using Apple's framework, which reports the GPS location if available, and, if not, the estimated location based on visible cell tow-

ers and strengths. The logger attempts to retrieve the location until the reported accuracy is less than 100m, or the location has been updated (by the framework) 3 times, in order to avoid draining the battery, yet still retrieve accurate location data. For cell ID location, we query the phone’s GSM modem using the AT command set, returning the currently associated cell ID.

The collected data is recorded on the phones, and transferred nightly to our servers on a secure connection. Our logger has recorded thousands of usage samples through the study, as presented in Table 1. Due to the extremely large size of the traces, it is often necessary to process them sequentially. Therefore, we developed most of the tools to process them using the Perl language and Bash scripts. We also took advantage of several open source tools for this purpose, including Cluster 3.0 [22, 23].

2.3. Collected Data

Usage

We look into the three principal types of mobile usage: phone calls, web usage, and app usage. We limit the number of usage categories considered to 100, similar to what we did with discrete context. This simplifies data processing, and can even increase accuracy by reducing usage cases with too few samples. We chose the number 100 based on the CDF of usage, covering 87%, 93%, and 99% of web, phone, and app usage.

We consider web usage as the independent websites a user visits, as presented by their domain names. We consider each visited domain as one entry, irrespective of the number of web pages under that domain. This would include the top level domain (TLD) and the first hierarchical subdomain, e.g. `www.example.com/url/` would be counted as `example.com`. For phone calls, while our logger does not record actual phone numbers, it records a one-way hash that uniquely identifies each phone number. We consider all phone calls including ones that have a length of zero, indicating no conversation. For application usage, we consider all applications the user utilizes, including built-in ones and those obtained from the App Store. We do not consider the home screen as an application, even though it is implemented as an application on the iPhone platform.

Context

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

For time&day, we separate weekends and weekdays, but otherwise treat days as the same. Separating weekends from weekdays not only makes intuitive sense, but our testing indicated that it performed better than treating all days the same. Therefore, with a one-minute resolution, time&day is a continuous number between 0 and 2880, to account for a two day period (a weekday and a weekend). For movement, we calculate the log of the power of the accelerometer readings. The reason we utilize the log of power, instead of absolute power, is the distribution of



Figure 1: Equal width discretization (top) would result in more natural boundaries. Equal frequency discretization (bottom) would use bins more efficiently and prevent too few samples in some bins.

power readings that is close to the power law. More than 99% of the $\log(p)$ entries fall between 0.1 and 10000, and the range is therefore limited accordingly.

For GPS location, we utilize the most accurate location provided by the iPhone API, which is provided in the geographic coordinate system, i.e. latitude and longitude. For cell ID location, we utilize the (single) cell ID reported by the phone.

3. QUANTIFYING CONTEXT-DEPENDENCY

As previously mentioned, context dependency can be broadly defined as a set of strict or probabilistic rules and relations between two often discrete variables context(s) and outcome [2]. Multiple theoretical, application agnostic metrics exist for measuring the relationship of such variables. These include entropy as a measure of uncertainty, and Pseudo R Square as a measure of correlation [24]. Yet, neither entropy nor Pseudo R Square can provide practical insight into the performance of context-aware applications.

We present estimation accuracy, based on maximum a posteriori probability (MAP) estimation, as our measure of choice for context dependency. Estimation accuracy can provide practical insight into the performance of many potential context-aware applications, while remaining application agnostic. In this section, we provide practical methods to calculate the a posteriori probability of an outcome (g) given context (x), or $P(g|x)$, from one or multiple continuous or discrete (multinomial) context sources.

3.1. Formal Definition

The use of context information can help increase the estimation accuracy of MAP estimation. MAP estimation works as follows. Assumes g takes value from a finite set $\{g_1, g_2, \dots, g_k\}$. Knowing the posterior probability of every possible outcome, g , under contextual information $x=(x_1, x_2, \dots, x_n)$, the optimal estimation for the outcome, \hat{g} , is

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

where $P(g|x)$ is the a posteriori probability. Now the expected estimation accuracy is $\sum_x P(g = \hat{g}(x)|x)P(x)$, which should be higher than $P(g = \hat{g})$, the expected estimation accuracy without the contextual observation x .

For many applications, the cost of a false negative is considerably higher than false positive. Therefore, providing multiple responses (i.e. best guesses) can be beneficial. Such responses would be in the form of $\hat{g} = \hat{g}_1 \cup \hat{g}_2 \cup \dots$. For example, in application preloading, the system could preload multiple applications to reduce the chance of not having the next application preloaded. We can use the same definition to allow multiple responses. In this case, the expected estimation accuracy would be

$$\sum_x P(g \in \hat{g}(x)|x)P(x).$$

3.2. Calculating Posterior Probabilities

The key to MAP estimation is the accurate calculation of the a posteriori probability distribution $P(g|x)$.

It may initially appear straightforward to calculate $P(g|x)$; simply dividing the number of times each possible outcome g_i has occurred under context conditions x by the number of times x has been observed in total. Recall that g takes value from a finite set, $\{g_1, g_2, \dots, g_k\}$. However, if the number of times x has been observed is small, the estimates of $P(g|x)$ are unreliable [25]. Due to the large number of possible context combinations, and the possibility of having few, or no prior samples in a given context, posterior probability estimates may become inaccurate or impossible. This is true even for individual context sources, but can become significantly worse if multiple context sources are treated as multiple dimensions, due to the curse of dimensionality [11]. In this subsection, we present the methods we use to address this challenge, for individual and multiple context sources.

Individual Context

For individual context, we employ Laplace Correction [26, 27] to reduce the negative impact of too few observations under some context conditions. Instead of calculating

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

we employ Laplace Correction and calculate $P(g_i|x)$ as

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

where m is the number of possible outcomes of g .

Note that Laplace Correction only smooths out the estimate of $P(g_i|x)$ when there are a small number of context samples. The effect of Laplace Correction is negligible for those if are significantly more samples than outcomes, i.e. $\text{count}(x) \gg m$.

Multiple Context

As presented earlier, treating the context space as a multi-dimensional space, with each dimension corresponding to one context source, will result in an unacceptably sparse data set due to the curse of dimensionality. In order to address this challenge for calculating posterior probability, $P(g|x)$, we employ classifier combination techniques. They enable us to treat each context source as a separate one-dimensional predictor, and combine multiple $P(\hat{g}_i|x_n)$ into $P(\hat{g}|x_1, x_2, \dots, x_n)$. In other words,

$$P(\hat{g}|x_1, x_2, \dots, x_n) \\ = \text{combination}(P(\hat{g}_1|x_1), P(\hat{g}_2|x_2), \dots, P(\hat{g}_n|x_n))$$

We explore three prominent classifier combination techniques. The first is *Simple* or *Naïve Bayesian*, which works under the assumption that different sources of context are conditionally independent. The Bayesian rule calculates

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

where

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

The assumption that different sources of context are independent does not necessarily hold. Even so, Simple Bayesian is known to often perform well even without this condition [28, 29]. We therefore evaluate the performance of Simple Bayes alongside other methods. Similar to individual context, we utilize Laplace Correction to reduce the ill effects of too few samples in calculating each $P(x, |g)$.

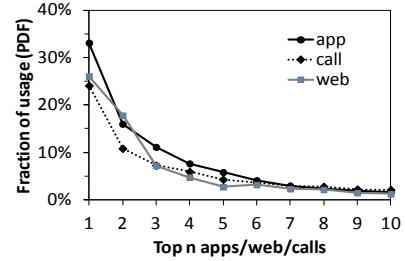


Figure 2: Estimating mobile usage is challenging due to the Power Law distribution of usage (average for all users); even given context based evidence, it is improbable for the posterior probability of a usage to rise above the more common usage.

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

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

For example, if one classifier selects outcome A with an 80% posterior probability, and two other classifiers select outcome B with 70% and 60% posterior probabilities, outcome A will be selected, and a posterior probability proportional to 80% is reported. The Maximum Rule is known to be highly sensitive to noise [30], when one classifier may be producing a high confidence due to noisy data or too few samples.

The third combination technique we explore is the *Mean Rule*, also known as the *Average Rule*. It calculates the probability of each outcome as the average of the reported probabilities by each of the classifying methods. Formally, $P(g_i|x_1, x_2, \dots, x_n) = \text{mean}(P(g_i|x_1), P(g_i|x_2), \dots, P(g_i|x_n))$. The Mean Rule is especially resilient to noise [31], and most useful when the classifiers are highly correlated.

3.3. Discretizing Context

Based on our definition of context dependency, we need to discretize continuous sources of context. Continuous sources of context inherently have a single or multidimensional structure to them, where samples close to each other are related. Such a structure allows for the use of unsupervised clustering techniques to discretize them. There have been many methods in literature for unsupervised clustering of single and multi-dimensional entries, and consequently creating the discrete cases. Furthermore, there is often the choice of *equal frequency* vs. *equal width* discretization, as shown in Figure 1. An equal number of samples per cluster would guarantee efficient use of clusters, and prevent a too few number of samples in some clusters producing inaccurate results. On the other hand, the equal sample constraint may artificially limit the boundaries in the clustering algorithm, resulting in inefficient clusters. Both equal width and equal frequency discretization are straightforward for one dimensional context, such as time and movement (accelerometer power).

For continuous context in multiple dimensions, i.e. location in our case, we refer to clustering literature to find a suitable unsupervised clustering algorithm. The resulting clusters would in turn become the categories. We have two requirements for such an algorithm. We chose the popular

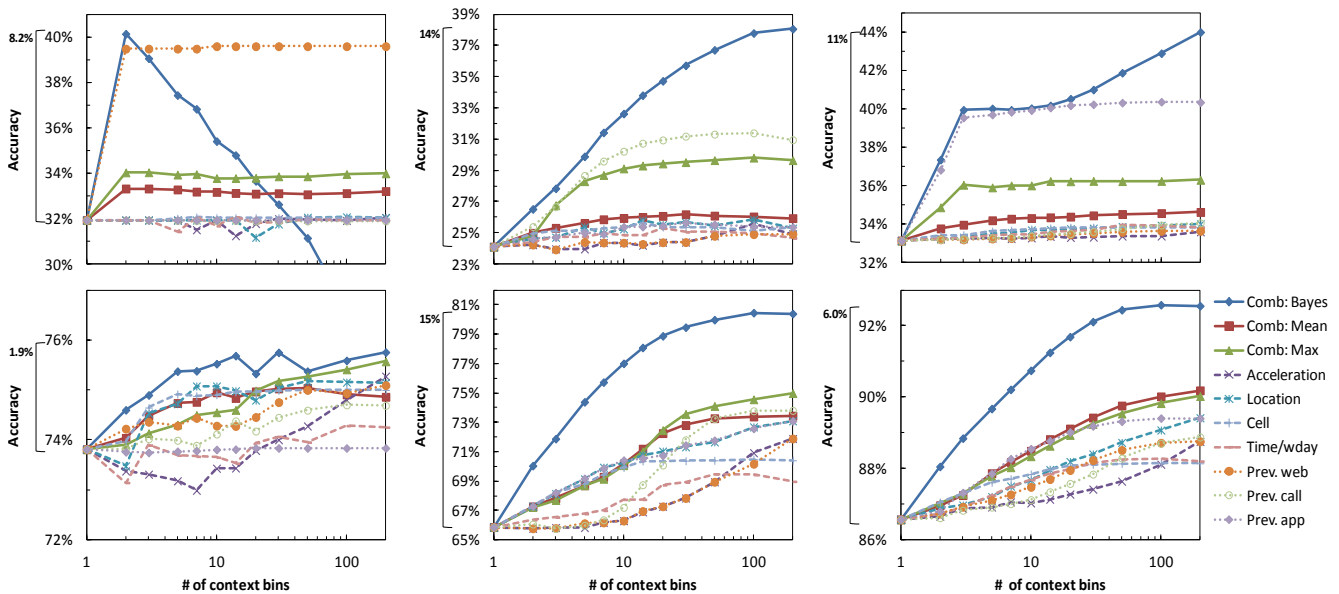


Figure 3: Context dependency of web (left), phone (middle), and application (right) usage, presented as the accuracy of prediction, for 1 (top), and 10 (bottom) responses. One bin means no context information was used.

k-mean clustering method because it satisfies our constraints; it clusters close samples together by minimizing the Within-Cluster Sum of Squares (WCSS), and it clusters entries into our desired number of clusters (k).

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

3.4. Binning Context

Even for context that is already categorical, in order to efficiently calculate $P(g|x)$, it is necessary to limit the number of possible categories for x , i.e. by grouping together some of the categories. We use *binning* to refer to the process of reducing the number of context categories. We define Simple Binning as follows. For binning categorical context into n bins, we simply choose the most popular $n-1$ categories, and group all other categories as the n 'th bin. This is especially reasonable if the distribution of context follows the power law, which is often the case. For consistency, for continuous context, we define Simple Binning into n bins as discretizing the context into n categories.

The number of bins chosen for any context involves an inherent tradeoff; more bins can allow finer molding of bins and more accurate posterior probability calculations, but at the same time would result in fewer samples per cate-

gory, increasing noise and reducing the accuracy of posterior probability calculations. Clearly, not only does the number of bins affect accuracy, but also how the bins are formed can affect accuracy. Supervised Binning can potentially make the best of both worlds in the above tradeoff by identifying and binning together categories that have similar outcomes. Therefore, supervised binning can allow finer molding of bins, increasing accuracy, without reducing the number of samples per bin that would increase noise and reduce accuracy. Supervised Binning can also be performed on continuous context, by first discretizing it into a larger number of categories.

There are two methods to perform Supervised Binning. First, one can either use the derived partitions from a classifier tree as the bins. An optimal classifier tree using (1 - estimation accuracy) as the loss function would be the optimal binning of context [11], but building it in our case is computationally prohibitive². The second method, which we use, performs clustering on the outcome distances to determine the bins. We use k-mean clustering based on the 2-norm distance of the normalized Laplace-corrected usage vectors to create the bins. Each usage vector is in the form of $\{P(g_1|x), P(g_2|x), \dots, P(g_{100}|x)\}$. One must note that in order to preserve the integrity of results when binning, it is necessary to separate the training data used for creating the bins from the testing data.

4. CONTEXT DEPENDENCY OF PHONE USAGE

In this section, using the context and usage traces, and our formal definition of context dependency, we present a series of interesting findings regarding the context dependency of web, phone, and app usage. As presented in Section 3, we utilize prediction accuracy as the metric for evaluating context dependency. We note that context-based

² One can, however, use heuristic methods such as CHi-squared Automatic Interaction Detector (CHAID) [32] to build a sub-optimal decision

tree.

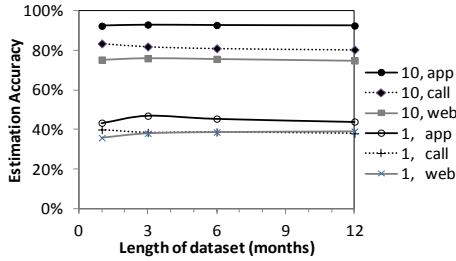


Figure 4: Affect of seasonal variation of mobile usage on its context dependency is small, and one to three months of training logs is sufficient. Estimation accuracy for with one and ten responses, calculated on trace durations of one, three, six, and 12 months.

prediction of usage is extremely challenging due to the distribution usage that closely resemble the power law. We can see in Figure 2 that the most popular usage of each user constitutes a major proportion of their usage, and much higher than the next most popular usage, and so forth. Consequently, even given context based evidence, it is improbable for the posterior probability of any usage to rise above the more common usage.

As mentioned in Section 2, applications and services in which the cost of a false negative is considerably higher than false positive can benefit from multiple responses in the form of $\hat{g} = \hat{g}_1 \cup \hat{g}_2 \cup \dots$. For example, for an app preloading application, the system might preload multiple apps to reduce the chance of a miss. Accordingly, we consider the case for multiple responses as well as the single response case.

4.1. Individual Context

We have studied the performance of individual context by measuring the estimation accuracy versus number of context bins, as shown in Figure 3. One context bin would mean no context, i.e. always returning the most likely result(s). We utilize leave-out-one cross-validation (LOOCV) to preserve the integrity of our results. LOOCV removes the test sample from the training data set used to calculate posterior probabilities. We utilize Simple Binning in order to keep the meaning of bins easy to understand. Recall from Section 3.4 that for categorical context, Simple Binning utilizes the top $n-1$ categories and an ‘other’ category. For continuous context, it simply discretizes it into n bins. In this section, we present our findings regarding the performance of each context, as well as the effect of the number of bins. Note that different context have a widely diverse range of effectiveness, depending on usage and the number of acceptable responses. Also, we see that more bins initially improves performance, but after a point will hurt performance. The reason is that even though more bins can allow finer molding of the model, hence a more accurate calculation of $P(g|x)$, it would result in fewer samples per bin, increasing noise and reducing accuracy.

An important finding not inherently obvious in the figures is that an increase in the number of context bins is useful only as long as there are a reasonable number of samples to reliably calculate the posterior probability of each bin. As a rule of thumb, there should be more than ten samples per bin, even though we are mitigating the ill effects of too few samples using Laplace correction. As shown in Table

1, there are on average 700 website visits per user. Therefore, it is unsurprising that in particular for equal cluster size contexts, there are diminishing results in going over 10-50 context bins. On the other hand, since there are over two thousand phone call samples, increasing the number of context bins is fruitful up to 100-200 bins, where the returns are diminished. This shows that the number of context bins should be not preset, but dynamically adjusted by the system to ensure a specific number of samples per bin. We next provide findings specific to each context type shown in Figure 3, using the better performing discretization method for each context, i.e. with and without the equal frequency constraint (Section 3.3) when applicable.

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

Movement: Similar to time&day, equal frequency discretization of accelerometer power performs slightly better than equal width discretization. Interestingly, a relatively high number of bins (e.g. 100) are most effective here. This is in contrast with our original expectations that a small number of bins, e.g. to account for moving and non-moving states, would be sufficient. This finding suggests that accelerometer power can and should be used *as a signature to classify a user’s detailed state*, and not merely as an indicator for whether they are moving or not. Previous research has shown a similar phenomenon with ambient sound for the purpose of room level localization, i.e. SoundSense [33].

GPS Location: In contrast to the single dimension contexts, location performed best without the equal frequency cluster constraint. We believe this is due to the equal frequency constraint artificially breaking down meaningful clusters in order to satisfy the sample size constraint. Interestingly, without the equal frequency constraint, a larger number of bins do not reduce performance. We believe this is due to the extra clusters mostly absorbing outliers, instead of breaking down meaningful clusters.

Cell ID Location: As each cellular cell spans a large coverage area, most of our users’ lives were under a small number of cell IDs. Therefore, there is little to gain from increased number of context bins. Note that as cell ID is already discrete, discretization doesn’t apply here.

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

4.2. Combinations of Context

By observing the performance of the three types of usage, we can see that combination methods are very useful when a number of meaningful context sources are present. The max-rule consistently outperforms the mean-rule for our

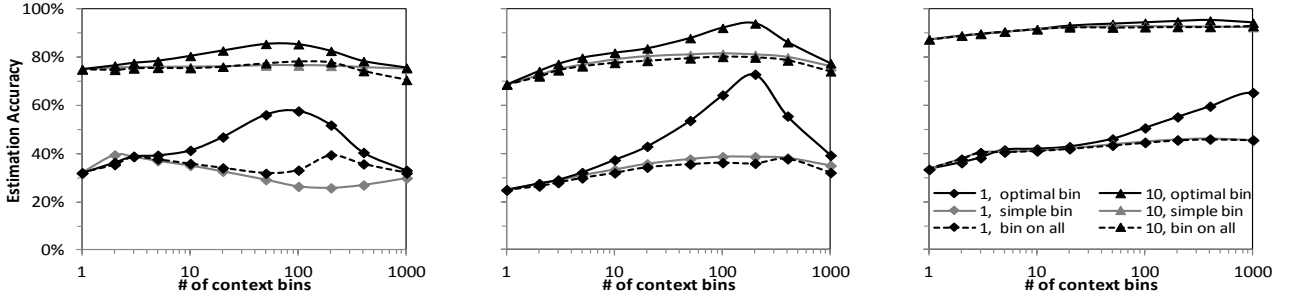


Figure 5: Supervised Binning, performed individually for each user, can greatly increase the accuracy of context-based usage estimation. Estimation accuracy of web (left), phone call (middle), and application usage (right), calculated using the Bayes method, for one, three, and ten acceptable responses. Compared to Supervised Binning on all users’ data, and individualized simple binning. One bin means no context information was used.

data. The mean-rule is known to perform well when classifiers are noisy and highly dependent [31]. Therefore, we conclude that different context sources are likely not noisy or highly dependent. The surprising fact that Bayes often outperforms other combinations is another indicator that our context sources are either not highly (conditionally) dependent, and/or their dependencies are distributed evenly [29]. On the other hand, even though we use Laplace Correction to reduce the impact of data sparseness, the performance of Bayes is reduced when there are more bins (with fewer samples and therefore more noise). Indeed, it is well known that Bayes is highly susceptible to noise. This is especially notable in web usage.

We note that as expected, and as confirmed by the traces (not shown), treating multiple context sources in a multi-dimensional manner results in virtually no improvement in estimation accuracy. This is unsurprising, as even with only 10 context bins, there were less than 1% of samples in any given context. More importantly, most samples belonged to bins that each contained less than 0.1% of samples.

4.3. Seasonal Variation

Our long-term traces allow us to answer an important hypothesis regarding context dependency: how significant is the effect of the user’s seasonal and temporal behavior variation on estimation accuracy? In other words, whether shorter durations may show higher context dependency due to the significant seasonal changes in user behavior [19], or would fewer samples in shorter durations reduce estimation accuracy.

We find that the context-dependency of usage remains relatively constant even for durations of one to three months, as shown in Figure 4. Furthermore, the fact that the performance is not significantly reduced for shorter durations indicates that a smaller data set, e.g. a month or more, would be sufficient for context-awareness. To this end, we present the same analysis as the prior section for the Bayes combination method, but instead of only calculating it for the entire 12 month duration of the study, we calculate it over one, three, and six month durations as well. For each duration and usage, we select the best number of bins, and present the average estimation accuracy of the multiple partitions resulting from each duration, e.g. the results from the four 3-month durations are averaged together.

4.4. Supervised Binning

As mentioned previously, there is an inherent tradeoff in choosing the number of context bins; more bins can allow finer molding of the model, and more accurate results, but it would at the same time result in fewer samples per category, increasing noise and reducing accuracy. Supervised Binning has the potential to make the best of both worlds, by identifying and binning together categories that have similar outcomes. We have studied the efficacy of binning, and have shown that it can greatly increase the accuracy of context based usage estimation.

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

We present significant performance increase of Supervised Binning in Figure 5. We use the Bayes combinatory method, since it produced the best results (Section 4.2). For fair comparison, we also show the performance of Simple Binning calculated using the same two-fold cross validation.

One interesting question is whether it is necessary to perform supervised binning individually per user, or are there inherent features in context that are common between users, and can allow the supervised binning to be performed once for all users, i.e. using data from all users. Our results confirm the former; even for our small, relatively homogeneous population, supervised binning using data from all of our users fails to improve accuracy over simple binning.

4.5. User Diversity

The long-term traces allow us to analyze the diversity in the context dependency of our users, i.e. whether some users have more diverse usage, and whether some users’ usage is more context dependent. We utilize the Kernel Density Estimation (KDE) to present the distribution of estimation accuracy among our participants, and compare it to the case without context information (only one bin), for one and ten acceptable responses, as shown in Figure 6. The estimation accuracy is calculated using our best methods, i.e. Bayesian combination and Supervised binning,

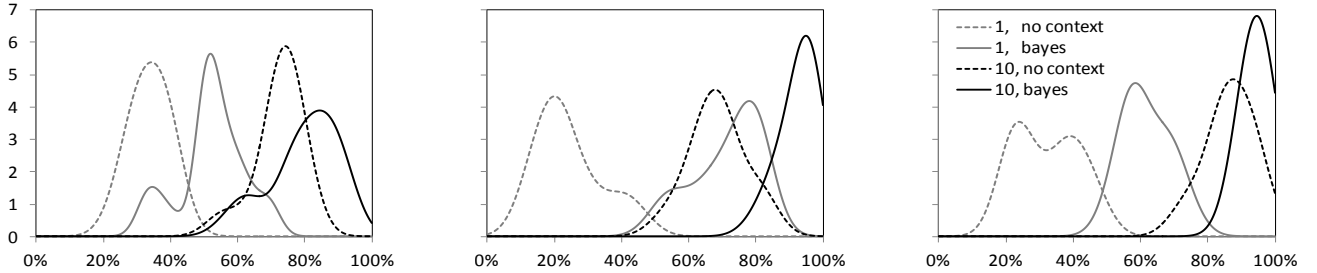


Figure 6: User diversity in context dependency is shown by the Kernel Density Estimation (KDE) distribution of estimation accuracy for one and ten responses. Web (left), phone (middle) and application usage (right).

and we empirically set the KDE bandwidth to 0.05.

Figure 6 shows that even though our 24 users were from undergraduates from the same school, there is considerable diversity among participants’ usage patterns. Yet, all of them show significant context-dependency in their usage. Furthermore, the top one and ten usage cases, which serve as the baselines for context dependency, constitute a significant share of usage for all participants. Finally, we note that among the three principal usage we studied, the estimation accuracy for web usage had a much higher diversity of context dependency, compared to its non-context case. This shows that there is more diversity in the context dependency of web usage, compared to phone and app usage.

4.6. Prior Usage Context

The methodological classifier combination approach to the data sparseness challenge of context awareness, presented in Section 4.2, allows the use of prior usage context in addition to sensor context. We refer to usage context as the last used websites, phone calls, and apps, and sensor context refers to the measurements of the device’s sensors, i.e. time&day, cell ID, motion, and GPS. In this subsection, we examine the effectiveness of prior usage context of varying depth for estimation accuracy. The depth of prior usage context is defined as how many prior usages are considered and combined along with the sensor context. A zero depth means no usage context is utilized. For the depth of one, the last used prior usage for web, phone, and apps is used, similar to other sensor context. So far, we have only utilized prior usage with a depth of one. Higher depth usage context can be treated in two ways. First, n ’th depth usage context can be simply presented as yet another single dimension usage context, which will be handled and combined similar to other context, as described in Section 3. Second, is to treat each n -depth usage context as an n -dimensional vector. Compared to the first approach, this method suffers from sparseness, due to the curse of dimensionality.

We have found that for both of these methods, estimation accuracy drops when the depth of historical context is increased beyond one. This limitation stems from inherent limitations of classifier combination for dependent data. Figure 7 shows the best estimation accuracy achieved through Bayesian combination and Supervised Binning, for different depths of prior usage context for phone usage. Web and app usage were similar, and are therefore not shown. It shows that after the depth of one, the gains start to diminish. This is due to the limitations of classifier combination; the additional information provided by depths

higher than one cannot offset the error induced by their dependence. Also, for the multidimensional treatment of n -depth context, due to its sparseness for most of the samples, Laplace Correction simply returns the average a posteriori probability as the conditional a posteriori probability. In turn, this adds dependency and reduces performance.

In order to better support historical usage context, for depth of greater than one, it is obvious that the dependency challenge needs to be addressed. Therefore, we propose and evaluate the following method for incorporating historical context: Treat usage context with a depth higher than one as multidimensional, but for each classification event and each type of usage, choose the highest depth that has more than m training samples. This method is referred to as auto-depth. The constant m is the minimum number of training samples deemed sufficient for an accurate a posteriori probability calculation. We set m to 10, based on the findings in Section 4. We find that the automatic depth selection approach works well, in terms of avoiding a performance drop, i.e. more information doesn’t hurt. However, we observed no measurably significant performance increase after the depth of one.

5. COST-AWARE CONTEXT COMBINATION

Our findings so far attest to the performance and usefulness of context-based usage estimation. However, obtaining and processing context information can incur significant energy costs. Many ad-hoc methods, sometimes themselves context based, have been employed to reduce energy cost of context awareness while satisfying the system designers cost / accuracy trade-off. These methods typically reduce the frequency of accessing energy hungry context sources, or to avoid using them altogether, substituting them by lower cost context sources.

In this section, we address this challenge through a methodological framework, SmartContext. It takes advantage of the methodology presented in Section 4, and builds upon the general problem of budgeted observation selection in the operations research community, to automatically optimize the energy cost of context-based estimation, while satisfying the accuracy requirements and tradeoffs that the designer sets for each estimation event. We focus on energy costs, but other definitions of cost may be used as well.

5.1. Assumptions and Requirements

SmartContext takes advantage of and is compatible with any classifier combination method, as long as the combination method can provide the MAP estimate and estimation

Table 2. Ordering & measured energy cost of context

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

accuracy using any combination of context sources with small processing overhead. All three classifier combination methods we evaluated have these features, but we will use Bayesian as it performed best.

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

5.2. Operation

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

Once the ranking is determined, the energy aware combination of context works as follows. For each classification event, SmartContext starts combining multiple sources of context information one by one, in the ordering determined in step one. This can be done with minimum processing overhead, and for any combination of context sources, as explained in Section 4.2. After running the classifier combination with each additional context source, it checks the criteria of the requesting application or service, for that estimation event. In the evaluation presented here, a fixed minimum estimation accuracy for every estimation event is utilized. However, the application or service may set a different accuracy requirement for each estimation event, or even consider the expected cost of accessing the next context source in determining when to settle with the current estimation accuracy and stop accessing more context sources. SmartContext assures the target estimation accuracy for each estimation event, as long as it is possible to reach that accuracy, while spending no excess cost in acquiring unnecessary context. In other words, in some conditions, no additional costly context is used, while in more uncertain conditions, SmartContext may use up to all the available context sources. The pseudo-code description of SmartContext is shown in Figure 10.

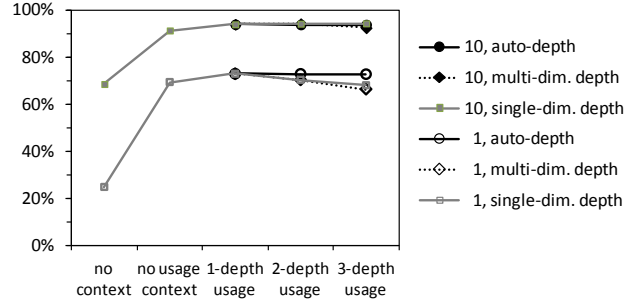


Figure 7: Automatic depth selection is necessary to efficiently utilize prior usage context with depth greater than one. Effectiveness of prior usage context in increasing estimation accuracy, for phone usage, vs. depth of prior usage context for one and ten acceptable responses, averaged among the 24 participants.

5.3. Ranking of Context Sources

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

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

Review of existing methods

Solutions toward this challenge are based on submodularity, an important property of the information gain from multiple observations [34]. Submodularity is also intuitively named as the diminishing returns property. It states that the information gain from an observation helps more if one has a smaller set of observations so far. Vice versa, the information gain from an observation helps less if it is added to a larger set. This can be formally presented as follows. The set function $F : V \rightarrow \mathbb{R}$ is submodular if

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

for all $A \subset A' \subseteq V$, $X \notin A$, i.e. adding X to a smaller set, A , helps more than adding it to a larger set, A' . The general problem of maximizing submodular functions is NP-hard [35], and general algorithms are unable to provide guarantees in terms of processing time [36], unless there are certain assumptions, e.g. selecting a subset among a fixed tree ordering of possible observations [37]. However, artificially imposing such a dependency tree is heuristic in nature and can reduce performance, e.g. in [38]. Further, calculating the maximum-likelihood dependence tree requires assumes pre-measured mutual information between unit cost observations are available [39], neither of which are applicable to our case.

Therefore, it is common practice to use the greedy (myopic) solution towards this selection problem [40]. The submodularity property ensures that such greedy solutions are near-optimal, typically with provable constant factor performance guarantees. The greedy solution, assuming a unit cost for all observations, selects the observation with the most information at every step, i.e. the marginal increase $F(A \cup X) - F(A)$ is maximized. For this case, in [34], Nemhauser et al. prove that any set of equal-cost observations selected in this manner performs, at worst, a factor of

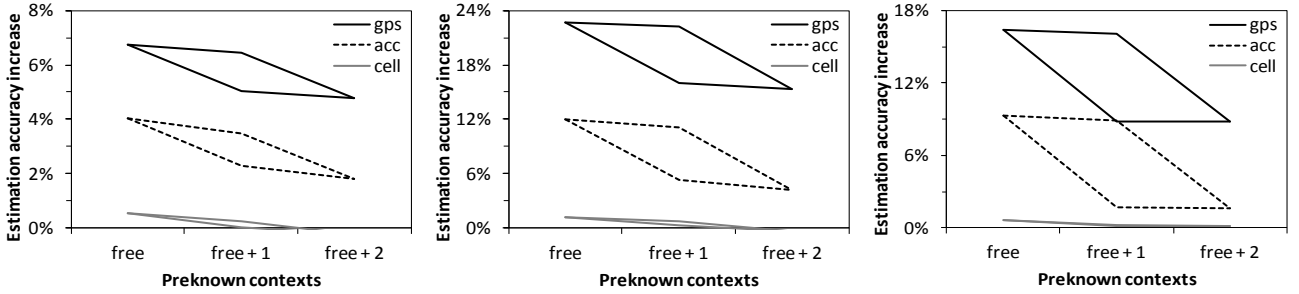


Figure 8: Submodularity of estimation accuracy; the average estimation accuracy gain resulting from combining a certain context decreases if more contexts were known (combined) beforehand. Free indicates time&day and previous usage. Free + 1 and free + 2 indicate one or both of the remaining two contexts (except the one to be added). Left: web usage. Middle: phone call usage. Right: app usage

$(1 - 1/e)$, compared to the optimal set. More recently, the operations research community has proved the same bound when observations have different costs. Krause et al. prove that for independent costs, the greedy solution selects the observation with the maximum *cost-effectiveness* at every step, i.e. the marginal increase divided by cost of observation, $(F(A \cup X) - F(A))/cost(X)$, is maximized [40]. Furthermore, they prove that approximation algorithms are unable to provide guarantees better than a constant factor of $(1-1/e)$, i.e. $(k \cdot (1-1/e))$. We therefore base our work on the solution provided by Krause et al. [40].

Ranking Mobile Context

SmartContext is based upon the greedy method described in the earlier section, guaranteeing [40] a performance bound of $(1 - 1/e)$. However, the performance guarantee requires two assumptions. First, the costs of context sources (observations) must be independent from each other. Indeed, mobile context sources typically have independent energy costs, as was our case. Second, the submodularity or diminishing returns property must hold for our utility function (estimation accuracy). While this appears a reasonable assumption, it is necessary to verify it. Existing work typically use entropy as their utility function, and either assume that it is submodular [34], or prove that it is submodular under an assumption of independence [40]. We experimentally verify the overall submodularity of estimation accuracy. For this purpose, it is necessary to show that the estimation accuracy gain resulting from adding (combining) any context source decreases if more context sources were known (combined) beforehand. Note that since SmartContext assumes that free context is always utilized, it is necessary to verify the submodularity only among costly context. Figure 8 shows the estimation accuracy gain for Cell ID, acceleration, and GPS location. Therefore, we conclude that the greedy approach works well for context awareness. In this case, the best ordering is obtainable by ranking the context sources according to their cost effectiveness. In this case, the cost effectiveness of each context source is the marginal estimation accuracy increase divided by its expected energy cost, i.e., $(F(A \cup X) - F(A))/cost(X)$. The expected energy cost can be pre-measured by the system designer, as in our case, or can be measured automatically in software as in [41]. The energy costs of acquiring context on the iPhone 3GS are presented in Table 2. The estimation accuracy of each context source

is shown in Figure 8. The resulting ranking is shown in Table 2. Note that due to the often significant difference in the energy cost of context sources on mobile devices, their ranking becomes close to the order of their energy cost.

Finally, we note that due to the relatively limited number of costly context sources on mobile devices, it is also possible to simply perform a thorough search, calculating the performance of SmartContext under all possible orderings of context sources. For our case, there are three costly context sources, resulting in a total of $3! = 6$ possible rankings. Unsurprisingly, for each of the experimental cases of Section 5.4, the rankings we obtained using the through search are the same as the greedy ranking.

5.4. Evaluation

We have evaluated SmartContext using the in-situ traces. Figure 9 shows, for different (minimum) target accuracies, how often each context source is utilized for Cell ID, Accelerometer, and GPS, and how often the target accuracy is achieved, as well as the overall average achieved estimation accuracy. Note that SmartContext always uses time&day and previous usages, as we assume they are available for free. We can see that even without energy hungry context sources, and only using free context, it is often possible to achieve good estimation accuracy, and the additional accuracy provided by the costly sources are incremental. This is expected, because each source of context has a small incremental value, as shown in Section 4.1, and because submodularity ensures diminishing returns, as shown in Section 5.3 and Figure 8.

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

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

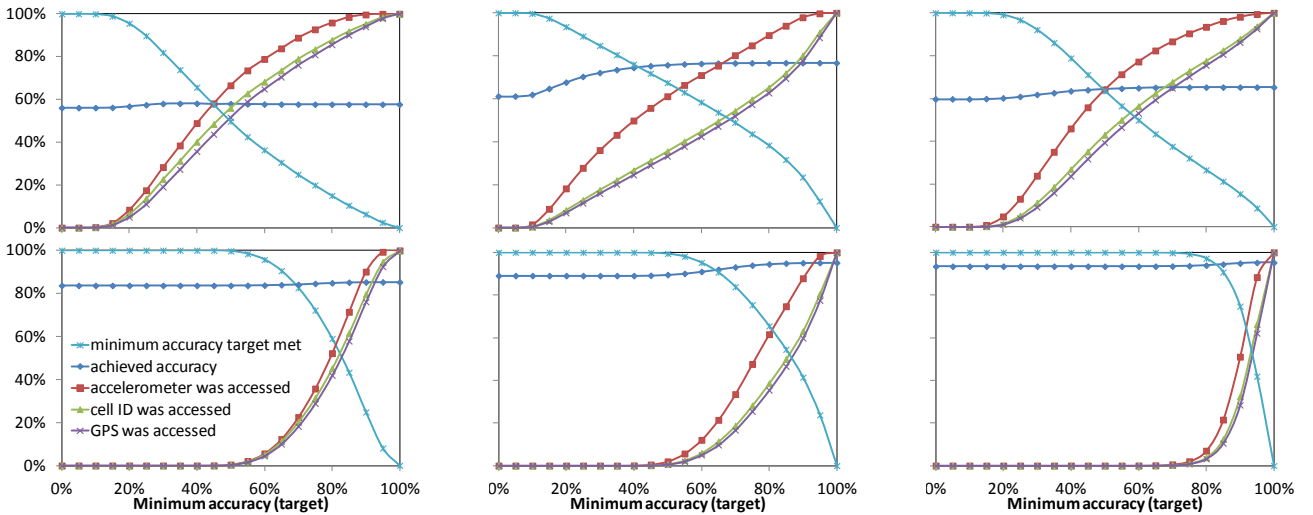


Figure 9: Performance of SmartContext. For a range of minimum accuracy targets, how often costly context (i.e. accelerometer, cell ID, GPS) is accessed, how often the minimum accuracy target is met, and the overall average estimation accuracy. Web (left), phone (middle), and application (right) usage, for 1 (top), and 10 (bottom) responses.

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

Figure 10: Pseudo-code for SmartContext

6. SAMPLE APPLICATIONS

The context dependency of mobile usage not only provides insight to the social behavior of humans, but can be utilized in many applications, such as those in Figure 12. In this section, we provide several potential applications to highlight the efficacy and simplicity of our approach. Their performance gains are based on our best performing methods, i.e. using Supervised Binning and Bayesian combination.

Web bookmarks

It is known that a few websites account for most of the typical user's usage [42]. Accordingly, some browsers, e.g. Opera, offer a list of favorites or home screen that is configurable by the user or automatically generated. This would provide with access to the user's most common websites. Others provide a user configurable home page that is automatically loaded when the browser is run. Similar to Section 4, we focus on individual domains, instead of pages within a domain.

A context-aware web favorites list is a sample application that can present more likely choices to the user according to their context. Our findings, presented in Figure 11, show that a context-based solution for providing the user with either a single default home page, or a list of 10 websites, significantly outperforms an ideal static selection, with a

miss rate of 15% vs. 25% for a list of 10 websites, and 42% vs. 68% for the single home page, and less than half the ideal static solution. Interestingly, the ideal static list of 10 favorite websites outperforms the 10 most recently visited.

Phone favorites list

In order to assist users in making phone calls, phones typically provide the user with a redial button, a list of recent phone calls, and/or a user configurable favorites list. For example, the iPhone used in our study provided a list of recent phone calls as well as a user configurable favorite contacts list. A context-aware phone favorites list is a sample application that can present more likely choices to the user according to their context.

On average, a static list of each user's top 10 contacts has a miss rate of 32%, and a recent call list has a miss rate of 28%. On the other hand, a context aware favorites list can reduce the users' need to go through their phonebook by approximately five fold, to 6%. Furthermore, the miss rate of a redial button is 64%, but the context-based dial button has a miss rate of 27% (Figure 11).

App Quicklaunch and Preloading

Most phones often have a list of apps that are more readily available for users to run, i.e. quicklaunch. The iPhone provides room for four such apps, which are readily available on any page of the home screen, and users can also organize their apps so the most common are placed in the first page. A context-aware quicklaunch list is a sample application that dynamically updates the quicklaunch list according to the users' context. Our findings show that it would have a miss rate of 16%, compared to the 39% miss rate of the ideal static quicklaunch, an improvement of three times. For 10 apps, the miss rate is just 4%, compared to 13% for the static case (Figure 11).

Preloading is another possible application, where context-based estimation of the application to be used can enhance performance. App preloading, including context-based methods have been widely studied in the past [43]. We have measured the app launch times on the substantially



Figure 12: Sample applications that can benefit from a dynamic context-aware selection vs. currently static selections. Left: Browser displays a list of bookmarks when launched. Center: Favorite phone contacts. Right: Apps on the home screen, and the always-visible quick launch bar (bottom row).

faster iPhone 4. The measurements were repeated three times for each app, and we excluded content load times, if applicable. For the without preloading case, each app's process was manually terminated between the runs. For the preloaded case, we started the app once, and closed it before the measurement run. Without preloading, the average load time was 2.0 seconds (median = 2.1, deviation = 0.5). With preloading, the load times were 0.6 seconds for all the measured apps. These measurements show that, on average, preloading can improve app load times over three fold. We note that the iPhone and many other platforms utilize a most recently used algorithm to keep multiple apps in memory, given memory limitations. We compare our performance to the MRU algorithm, and show that the miss rate for 10 preloaded apps, the improvement is from 9% to 4% (Figure 11). Furthermore, for 3 preloaded apps, the miss rate is reduced from 31% to 17%, almost half.

7. RELATED WORK

Prior work (e.g. [2]) also define context dependency as a set of strict or probabilistic rules and relations between context(s) and the outcome. Others design and implement frameworks for sensing and processing context information [44, 45]. For a survey, see Baldauf et al. [46]. These work attest to the significance and usefulness of context. Context information, has in the past been widely used for specific applications such as implicit user interaction (e.g. by Schmidt [5]) and information delivery (e.g. a reminder system by in [6], a tourist guide in [7], and content adaptation in [3] and [4]). For a survey of such cases, see Chen and Kotz [47]. Others have presented system mechanisms based on context information, e.g. estimating and predicting wireless network conditions [10], network routing [48], battery management [8, 49], and energy efficient GPS duty cycling [12, 13]. Further, Eagle and Pentland have shown that device usage patterns are indeed structured and predictable [50]. These designs and others depend on the context dependency of device usage, and show significant, quantified, performance gains by exploiting context. A number of other work depend on knowledge regarding usage to perform. For example, the work in [10] predicts network conditions to choose the best network interface, but assume network usage is pre-known, even though it depends on the behavior of applications, services, and the

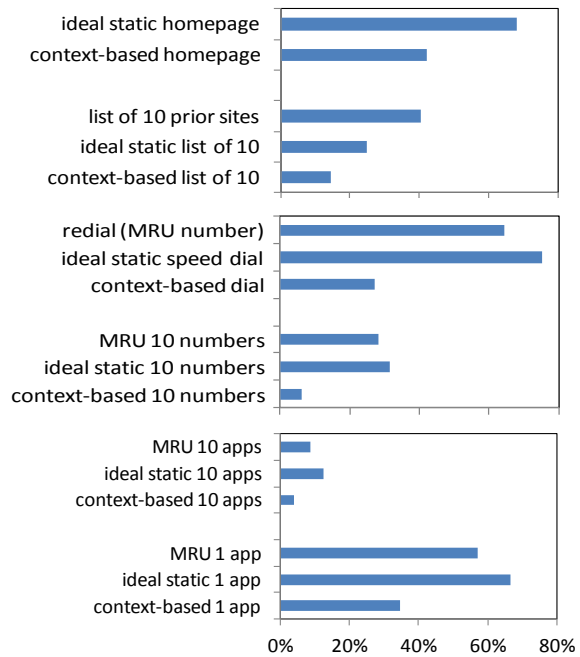


Figure 11: Performance of context-based applications, presented as miss rates. Top: web bookmarks. Middle: phone favorites. Bottom: application launching or preloading

user. As another example, the authors of [8] show that battery consumption is context dependent. The authors of [9] further research this problem by focusing on phone call usage, and show that call lengths, and therefore their energy consumption, are context dependent. Further, Eagle and Pentland have shown that device usage patterns are indeed structured and predictable [50]. The usefulness of context has been so significant that many researchers have designed and implemented frameworks for the specific task of sensing and processing context [44-46]. Our work presents a methodological solution for using multiple and various sources of context while managing their energy costs, and presents a formal definition of context dependency as well as practical methods to calculate it. We abstain from focusing on a single application or service, yet we provide practical insight into the relationship between context-dependency and the performance achievements of individual applications. A number of recent research have dealt with reducing the cost of acquiring context. These work attest to the challenge of energy efficiency in context awareness, but typically focus on single applications and/or static configurations. They use one or more of the following three techniques to reduce energy cost, while retaining acceptable performance. First: *frequency reduction*, as in [12-14] reduces the sampling frequency of energy hungry context sensors. Second: *sensor substitution* utilizes lower energy cost context instead of energy hungry ones, as in [10, 12, 13]. Third: *sensor elimination* attempts to utilize a subset of sensors. We take the third approach in SmartContext, but unlike previous work that focus on and take advantage of the properties of a specific application, such as activity detection [15-18], we provide a generic framework for system designers to dynamically or statically optimize the cost /

accuracy trade-offs of context awareness. A more general problem of selecting the most effective subset of sensors, also referred to as observations or predictors, has been the focus of decades of research in the artificial intelligence and operations research communities. These work focus on the optimization of information, typically defined as either joint entropy or information gain (delta entropy). It is common practice to use the greedy (myopic) solution towards this selection problem [40, 51, 52], with guaranteed performance bounds, due to the processing complexities of finding the optimal solution [35-37]. SmartContext builds upon the greedy solution of Krause et al. [40], but is adapted to using estimation accuracy instead of information gain.

There has been several research utilizing phone logging, e.g. [53-56]. Compared to our traces, they collect very limited context information due to privacy concerns and battery lifetime limitations. We have overcome these challenges by the careful design and implementation of the study, and have collected unprecedented data.

Finally, there has also been considerable research on determining user state from context information e.g. physical activity [57]. In this work, we abstain from extracting user state, either directly or as an interim stage, and instead focus on the relationship between device usage and context information.

8. SUMMARY AND CONCLUSION

We have found that estimation accuracy based on MAP estimation is a practical application agnostic measure for context dependency, yet can provide insight into the real-life performance of many applications, several of which are briefly presented here. These applications attest to the effectiveness of context for estimating usage, and highlight the practical value of estimation accuracy as the measure of choice for context dependency.

We have also found that due to the power law distribution of usage, estimating mobile usage is very challenging. Yet, we show that careful yet methodological treatment of multiple sources of context, e.g. combination, discretization, binning, can greatly increase estimation accuracy. In particular, we have found that 1) it is necessary to maintain a reasonable number of usage samples in each category, i.e. no less than ten, and equal frequency discretization of single dimension context helps achieve this. 2) Classifier combination methods can address the data sparseness challenge when utilizing multiple context sources, and Bayesian combination works best, even though the contexts are dependent. 3) Individualized supervised binning greatly improves estimation accuracy by keeping a more samples in each bin while allowing the fine molding of bins.

Finally, even though the energy cost of some context sources can be a substantial challenge for context based applications. We address this challenge through the SmartContext framework, which ensures using only as much context sources to meet a minimum accuracy set by the application designer for each estimation event. We show that SmartContext can achieve an estimation accuracy within 1% of the maximum possible, while reducing energy costs by 60% or more.

9. REFERENCES

- [1] Schilit, B., Adams, N. and Want, R., Context-aware computing applications. in *Proc. Workshop on Mobile Computing Systems and Applications*, 1994.
- [2] Gu, T., Pung, H.K. and Zhang, D.Q., A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 2005.
- [3] Lemlouma, T. and Layaïda, N., Context-aware adaptation for mobile devices. in *Proc. Int. IEEE Conf. on Mobile Data Management*, 2004.
- [4] Kane, S.K., Wobbrock, J.O. and Smith, I.E., Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. in *Proc. ACM Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*, 2008.
- [5] Schmidt, A., Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 2000.
- [6] Marmasse, N. and Schmandt, C., Location-Aware Information Delivery with ComMotion. in *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, 2000.
- [7] Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C., Developing a context-aware electronic tourist guide: some issues and experiences. in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2000.
- [8] Banerjee, N., Rahmati, A., Corner, M.D., Rollins, S. and Zhong, L., Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. *Proc. Int. Conf. Ubiquitous Computing (Ubicomp)*, 2007.
- [9] Ravi, N., Scott, J., Han, L. and Iftode, L., Context-aware Battery Management for Mobile Phones. *Proc. Int. Conf. on Pervasive Computing and Communications (PerCom)*, 2008.
- [10] Rahmati, A. and Zhong, L., Context-Based Network Estimation for Energy-Efficient Ubiquitous Wireless Connectivity. *IEEE Transactions on Mobile Computing*, 2011.
- [11] Hastie, T., Tibshirani, R. and Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [12] Zhuang, Z., Kim, K.-H. and Singh, J.P., Improving energy efficiency of location sensing on smartphones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.
- [13] Paek, J., Kim, J. and Govindan, R., Energy-efficient rate-adaptive GPS-based positioning for smartphones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.
- [14] Wang, Y., Krishnamachari, B., Zhao, Q. and Annavaram, M., Markov-optimal sensing policy for user state estimation in mobile devices. in *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2010.
- [15] Panuccio, P., Ghasemzadeh, H., Fortino, G. and Jafari, R., Power-aware action recognition with optimal sensor selection: an AdaBoost driven distributed template matching approach. in *Proc. ACM Workshop on Mobile Systems, Applications, and Services for Healthcare*, 2011.
- [16] Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., Tröster, G. and Verdone, R., Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection. *Wireless Sensor Networks*, 2008.
- [17] Kang, S., Lee, J., Jang, H., Lee, H., Lee, Y., Park, S., Park, T. and Song, J., SeeMon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2008.
- [18] Wang, Y., Lin, J., Annavaram, M., Jacobson, Q.A., Hong, J., Krishnamachari, B. and Sadeh, N., A framework of energy efficient mobile sensing for automatic user state recognition. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2009.
- [19] Shepard, C., Rahmati, A., Tossell, C., Zhong, L. and Kortum, P., LiveLab: measuring wireless networks and smartphone users in the field. *SIGMETRICS Perform. Eval. Rev.*, 2010.
- [20] Rahmati, A., Qian, A. and Zhong, L., Understanding human-battery interaction on mobile phones. in *Proc. ACM Int. Conf. Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2007.
- [21] Rahmati, A. and Zhong, L., "Studying smartphone usage: lessons from a four-month field study", in *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, July 2013.
- [22] Hoon, M.J.L.d., Imoto, S., Nolan, J. and Miyano, S., Open Source Clustering Software. 2004.
- [23] Cluster 3.0. Open Source Clustering Software <http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm>.
- [24] Scott Long, J. and Freese, J. *Regression models for categorical dependent variables using stata*. Stata Press, 2006.

- [25] Kuncheva, L. *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.
- [26] Provost, F. and Domingos, P., Tree Induction for Probability-Based Ranking. *Machine Learning*, 2003.
- [27] Zadrozny, B. and Elkan, C., Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *Proc. Int. Conf. on Machine Learning (ICML)*, 2001.
- [28] Domingos, P. and Pazzani, M., Beyond independence: Conditions for the optimality of the simple Bayesian classifier. in *Proc. Int'l Conf. Machine Learning*, 1996.
- [29] Zhang, H., The optimality of naive Bayes. *Proc. Int Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2004.
- [30] Duin, R. and Tax, D. Experiments with Classifier Combining Rules. in *Multiple Classifier Systems*, Springer Berlin / Heidelberg, 2000, 16-29.
- [31] Kittler, J., Hatef, M., Duin, R.P.W. and Matas, J., On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [32] Hill, T. and Lewicki, P. *Statistics: Methods and Applications*. StatSoft, Inc., 2005.
- [33] Lu, H., Pan, W., Lane, N.D., Choudhury, T. and Campbell, A.T., SoundSense: scalable sound sensing for people-centric applications on mobile phones. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2009.
- [34] Nemhauser, G.L., Wolsey, L.A. and Fisher, M.L., An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 1978.
- [35] Ko, C.-W., Jon, L. and Queyranne, M., An Exact Algorithm for Maximum Entropy Sampling. *Operations Research*, 1995.
- [36] Goldengorin, B., Tijssen, G.A., Tso, M. and A. The maximization of submodular functions : old and new proofs for the correctness of the dichotomy algorithm, University of Groningen, Research Institute SOM (Systems, Organisations and Management), 1999.
- [37] Krause, A. and Guestrin, C., Optimal nonmyopic value of information in graphical models-efficient algorithms and theoretical limits. *Proc. IJCAI*, 2005.
- [38] Guestrin, C., Krause, A. and Singh, A.P., Near-optimal sensor placements in gaussian processes. in *Proceedings of the 22nd international conference on Machine learning*, 2005.
- [39] Chow, C. and Liu, C., Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 1968.
- [40] Krause, A. and Guestrin, C., Near-optimal nonmyopic value of information in graphical models. in *Proc. of Uncertainty in AI*, 2005.
- [41] Dong, M. and Zhong, L., Self-constructive high-rate system energy modeling for battery-powered mobile systems. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2011.
- [42] Tossell, C., Kortum, P., Rahmati, A., Shepard, C. and Zhong, L., "Characterizing web use on smartphones", in ACM SIGCHI Conf. Human Factors in Computing Systems (CHI), 2012.
- [43] Yan, T., Chu, D., Ganesan, D., Kansal, A. and Liu, J., Fast App Launching for Mobile Devices using Predictive User Context. in *Under Review*, 2012.
- [44] van Kranenburg, H., Bargh, M.S., Iacob, S. and Peddemors, A., A context management framework for supporting context-aware distributed applications. *Communications Magazine, IEEE*, 2006.
- [45] Lee, Y., Iyengar, S.S., Min, C., Ju, Y., Kang, S., Park, T., Lee, J., Rhee, Y. and Song, J., MobiCon: a mobile context-monitoring platform. *Commun. ACM*, 2012.
- [46] Baldauf, M., Dustdar, S. and Rosenberg, F., A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2007.
- [47] Chen, G. and Kotz, D. A survey of context-aware mobile computing research, Citeseer, 2000.
- [48] Cugola, G., Migliavacca, M., Roedig, U. and Sreenan, C., A Context and Content-Based Routing Protocol for Mobile Sensor Networks. *Wireless Sensor Networks*, 2009.
- [49] Ravi, N., Scott, J., Han, L. and Iftode, L., Context-aware Battery Management for Mobile Phones. in *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, 2008.
- [50] Eagle, N. and Pentland, A., Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 2009.
- [51] Van Der Gaag, L. and Wessels, M., Selective evidence gathering for diagnostic belief networks. *AISB Quarterly*, 1993.
- [52] Dittmer, S.L. and Jensen, F.V., Myopic value of information in influence diagrams. in *Proc. conf. Uncertainty in Artificial Intelligence (UAI)*, 1997.
- [53] Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B. and Landay, J.A., MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. in *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2007.
- [54] Eagle, N. and Pentland, A., Reality mining: sensing complex social systems. *Personal Ubiquitous Computing*, 2006.
- [55] Rahmati, A. and Zhong, L., Usability evaluation of a commercial Pocket PC phone: A pilot study. *Proc. ACM Int. Conf. Mobile Technology, Applications and Systems (Mobility)*, 2007.
- [56] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R. and Estrin, D., Diversity in Smartphone Usage. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.
- [57] Bouten, C.V.C., Koekkoek, K.T.M., Verduin, M., Kodde, R. and Janssen, J.D., A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily Physical Activity. *IEEE Trans. Bio-medical Engineering*, 1997.

Ahmad Rahmati received the BS degree in computer engineering from the Sharif University of Technology in 2004 and the MS and PhD degrees from the Department of Electrical and Computer Engineering at Rice University in 2008 and 2012. He is a senior staff scientist at the Mobile and Wireless Group at Broadcom. His publications received the ACM MobileHCI Best Paper Award in 2007 and have been featured twice as the spotlight paper of the IEEE Transactions on Mobile Computing, in 2010 and 2011. His research interests include mobile and wireless system design and applications, context-aware computing through sensing and statistical learning on large data, as well as human factors and HCI. He is a member of the IEEE.

Clayton Shepard received the BS degree in 2008 and the MS degree in 2012 from Rice University, where he is currently working toward the PhD degree. He is a member of the Rice Efficient Computing Group, led by Dr. Lin Zhong. In 2008, he was a visiting researcher with Motorola's Advanced Research and Technology lab, where he also interned in 2007. His research interests include mobile systems and ubiquitous low-power computing. His current research focus is many-antenna base stations. In 2011 and 2012, he interned with Bell Labs, Alcatel-Lucent. He received the NDSEG Fellowship Award in 2011. He is a student member of the IEEE.

Chad C. Tossell received the BS degree in psychology from the University of California, Berkeley in 2003, the MS degree in applied psychology from Arizona State University in 2006, and the PhD degree in psychology from Rice University in 2012. He currently leads training research for the US Air Force (USAF) Research Laboratory at Wright-Patterson Air Force Base, Ohio. He oversees research and development efforts aimed at enhancing human and team performance within command and control, intelligence, and cyberspace domains. His other research interests include the use and usability of mobile systems, personalization of technology, and the human factors of voting with handheld devices.

Lin Zhong received the BS and MS degrees from Tsinghua University in 1998 and 2000, respectively, and the PhD degree from Princeton University in September 2005. He is an associate professor in the Department of Electrical & Computer Engineering, Rice University. He received the US National Science Foundation CAREER award and Best Paper Awards from ACM MobiSys 2011, IEEE PerCom 2009, and ACM MobileHCI 2007. A paper he coauthored was identified as one of the 30 most influential papers in the first 10 years of the Design, Automation & Test in Europe conference. His research interests include mobile computing, human-computer interaction, and nanoelectronics. He is a member of the IEEE.

Philip Kortum received the PhD degree from the University of Texas at Austin in 1994. He is a faculty member in the Department of Psychology at Rice University in Houston, Texas. Prior to joining Rice, he was at SBC Laboratories (now AT&T Laboratories) for almost a decade doing human factors research and development in all areas of telecommunications. He continues to work on the research and development of user-centric systems in both the visual (web design, equipment design, and image compression) and auditory domains (telephony operations, mobile computing, and interactive voice response systems).