Collaborative Opportunistic Sensing with Mobile Phones

Abstract
Mobile phones include a variety of sensors that can be used to develop context-aware applications and gather data about the user's behavior, including the places he visits, his level of activity and how frequently and with whom he socializes. The collection and analysis of these data has been the focus of recent attention in ubiquitous computing, giving rise to the field known as mobile sensing. In this work, we present a collaborative extension to InCense, a toolkit to facilitate behavioral data gathering from populations of mobile phone users. InCense aims at providing people with little or no technical background with a tool that assists in the rapid design and implementation of mobile phone sensing campaigns. By extending the architecture of InCense to support distributed sensing campaigns we are able to incorporate several strategies aimed at optimizing battery, storage, and bandwidth. These issues represent significant challenges in sensing campaigns that generate considerable amounts of data (i.e., collecting audio) or quickly drain the battery in the device (i.e., GPS), given the limitations of mobile devices. In this work, collaborative sensing is used to decide which mobile phone should capture audio when two or more devices are potentially recording a similar audio signal.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

UbiComp ’14 Adjunct, September 13 - 17, 2014, Seattle, WA, USA
Copyright 2014 ACM 978-1-4503-3047-3/14/09...$15.00.
http://dx.doi.org/10.1145/2638728.2638814
Author Keywords
Opportunistic sensing; collaborative sensing; mobile phone sensing; behavioral sensing

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Mobile phones are to become one of the most pervasive technologies ever, which coupled with their communicative and growing processing power, are increasingly becoming essential in our daily activities. The pervasiveness of mobile phones and their built-in sensors have facilitated the emergence of a fairly new research area known as mobile phone sensing. This area has brought about unprecedented opportunities for scientists and practitioners to better understand, and potentially influence, users and contexts.

In this work, we present results of an ongoing project in which we are aiming at sensing the activities of older adults to get acquainted with their behavioral patterns. In doing so, we have faced several battery-saving challenges such as reducing communication and processing costs, or data storage and transmission from data coming from different types of sensors. To address this, we implemented a collaborative, distributed version of the InCense mobile sensing toolkit, an application aimed at facilitating the design and deployment of mobile sensing campaigns. Collaborative sensing mainly refers to battery-saving and performance-wise strategies, which help address technical issues associated to the inherent limitations of mobile devices, such as storage capacity, bandwidth, latency, and battery life. In this work in particular, collaborative sensing is used to decide which mobile phone should capture audio when two or more devices are nearby i.e., they can record a similar audio signal. Addressing these issues is paramount for making continuous opportunistic sensing a reality. We believe this work can help other researchers in the area facing similar challenges (see [1, 2]).

The InCense sensing toolkit
In the last couple of years, we have been working on designing and implementing a general purpose mobile phone sensing platform: InCense. We aim at having an open platform that can be used by people with little or no technical background to deploy sensing campaigns through a visual programming paradigm.

One of the core ideas behind InCense is having loosely-coupled components for carrying out particular tasks, useful for particular sensing campaigns. For example, knowing if a user is at home, detecting others around, or counting steps are components that could be independently created and put together for a particular campaign. These components could be replaced, enhanced, and created from scratch by other researchers and added to a shared repository. In this paper, we will describe high-level details of InCense, to explain how the tool was extended into a collaborative application to deal with some of the performance issues faced in mobile sensing, but further details of InCense can be found in [3, 4].

One of the main advantages of using InCense is that it has low overhead in sensing campaigns that are similar in essence; it runs quietly in the background of a mobile phone; it provides an interface for adding new components; and it can be programmed a priori to
react to certain events that occur to the mobile phone (e.g., receiving a phone call) or pertaining to user behavior (e.g., going out). In addition, the InCense platform provides an ontology representing the components as well as relationships among them [5], which enables us to verify the soundness of a certain campaign. In short, Fig. 1 shows a high level description of its architecture and how its components relate to each other. Currently, data analysis and classification must be carried out separately, meaning that InCense is mainly used for data collection (see [3]).

**Collaborative InCense**

Due to the nature of the sensing campaigns, some of the components are fundamentally mobile and reside in the mobile phones, whereas others reside in non-mobile nodes such as a local server or in the cloud. In scenarios where multiple devices must interact with each other, this feature adds significant complexity to the design of the campaigns, as the researcher has to consider balancing workload on mobile phones and servers in terms of processing and data transfer.

This configuration is not trivial as it involves considering multiple variables related to the diversity of the participating devices such as the processing power, battery consumption, number of devices involved, types of data to be sensed, server location, data transfer rates and costs, etcetera. In order to intrinsically support and consider some of these aspects since the inception of a sensing campaign, several characteristics have been added to a collaborative version of InCense. This is not deeply discussed due to space constraints. However, we will instead discuss a sensing campaign and challenges to reduce battery consumption.

**Sensing campaign: detecting problematic behaviors among elders with dementia**

We designed an observational study of problematic behaviors in institutionalized persons with dementia (PwD). The aim of the study was to identify behavioral and psychological symptoms of dementia (BPSD) such as agitation, anxiety, and depression in order to implement a suitable intervention to support the PwD and her caregiver. These behaviors are characterized by different physiological signs such as repetitive body movements, variations in walking gait, shouting, and/or crying. Thus, detecting these movements through the built-in sensors on mobile phones seems feasible since those bodily expressions are usually very pronounced.

In collaboration with the staff of a geriatric home, seven residents (Females=5, Males=2, ranging from 81 to 94 years old) were selected to participate in the study. During two weeks for four hours a day, participants were closely observed to document BPSD incidents using BPSDiary, an Android application based in the Neuropsychiatric Inventory Questionnaire (NPI-Q) [6], a widely accepted clinical instrument for evaluating psychopathology in dementia. The distress in the PwD caused by the BPSD is usually rated as mild, moderate, or severe. Precise signs exhibited by the PwD, along with the duration of the episode were recorded for each incident. Additionally, we kept track of social interaction with the residence staff and other residents. Throughout the observation subjects were carrying a mobile phone running InCense to collect accelerometer data (e.g., to identify physiological manifestations such as repetitive motion and wandering) and ambient sound (e.g., to detect shouting.
and/or repetitive verbalizations). During the second phase of the study, subjects were shadowed for 2 weeks to observe and record BPSD incidents using the BPSDiary. Once an incident was recorded, the BPSDiary application recommended a suitable intervention based on the actual context and the profile of the PwD.

InCense Sensing Script for Detecting Problematic Behaviors in Older Adults

Fig. 2 shows the InCense configuration for this sensing campaign. This InCense app is divided in two sections, one runs on each of the participant’s mobile phones, and the rest on a server located in the nursing home where behavioral data gathering was conducted. In the mobile app, an InCense sensor captures raw data from the accelerometer (ACC) to be stored in a Data sink (Si) for posterior analysis. Additionally, a collaborative sensing module is used to decide which mobile phone is more appropriate to capture audio when two or more subjects are close enough to record a very similar audio signal. In addition, each mobile phone captures audio raw data that is fed into a filter (F) to extract an Audio Fingerprint (AFP), which is a content-based audio representation that can be used to measure the similarity between audio signals [7]. To this end, the “Estimate Proximity” filter, residing on the Server, receives AFPs from all mobile phones to identify in those recordings fairly similar audio information. Finally, the “Optimum Node” is a filter that decides which mobile phone will carry out the audio recording, and then triggers a condition to deactivate recording in those mobile phones not selected. This is one of the algorithms implemented for collaborative sensing.

Performance improvement during the sensing campaign

1) REDUCING AUDIO PROCESSING AND STORAGE: The goal of this experiment is to use auditory information to detect devices that are recording the same audio information. The setup is twofold. First, we can optimize computing and resource usage in the devices. On the other hand, shared audio cues hint proximity between subjects, and this information can be valuable data per se.

Figure 2. InCense app for detecting behavioral and psychological symptoms of dementia.

Raw audio data is both large and sensitive to noise; hence, we use a compact representation of the audio called an Audio Fingerprint (AFP). This serves two purposes. On the one hand, it reduces the transmission overhead to the context server and being a stable representation of the audio can also be used to discover if two devices recording in the same scene. As shown in [7], computing just the amount of information carried...
by a signal is robust under changes of volume, low pass filtering, compression, and small amounts of white noise. To compute this AFP, we first obtain a vector with the entropies from the incoming audio signal. For this, we compute the entropy every frame of size ~370ms with an overlap of 87.5 percent. The entropy vector is then binary coded to reflect only the modulation of the entropy over time i.e., if the entropy increases from one frame to the next, it is encoded as 1; if it stays the same or decreases, it is encoded as a 0. This AFP is very compact, using only one bit per frame. Two AFP can be compared using the hamming distance. For a WAV audio file of 6 minutes, the size of the AFP is 0.012 percent of the original size, and it takes ~540ms to process 6 minutes of audio (we used a Dell Precision T5500, Processor Intel(R) Xeon(R) CPU 2.00GHZ). An advantage of this AFP is that we do not send the original audio to the server, and we cannot inverse transform data, thus maintaining users’ privacy.

To test our method, we collected two sets of data. The test data 1, was recorded in the lab, whereas the test data 2 was recorded in the older adults’ residence. Both sets were captured with a 24000 Hz sampling frequency with 8 bits of depth. The test data 1 consist of 18 minutes of audio collected with three recording devices (i.e., A, B, and C). In this scenario, we can observe the following combinations of data recordings: A, B, and C are together; A and B are together but not C; A and C are together but not B; B and C are together but not A; and, finally, all the devices are in different locations and therefore recording different data. The test data 2 consists in 6m 30s of audio collected with two recording devices (i.e., A and B). In this scenario we can observe the following combinations of data recordings: A and B are together, and A and B are at different locations.

Audio streams are compared in support windows, which are the minimum lapse of the audio for a comparison. We used 15 seconds support windows in our setup. We compare every 15-second window of one stream with the corresponding neighboring windows of the second string considering a shift threshold of at most 300ms. This shift accounts for a time lapse or phase shift caused by the audio arriving to the recording devices from different relative positions. For each possible alignment we compute the corresponding hamming distance and locate the minimum. Since we are overlapping the measures, every distance computation represents 1.7s of audio.

After this preprocessing, we classify the stream either as accompanied or alone. We did set by hand two thresholds (upper and lower) for the distance; if something is in between it remains unclassified in a first pass. On a second pass, we get rid of small flips in the class value if the time support is smaller than a time threshold.

We tested two pairs of thresholds, TH1 and TH2, as reported in Fig. 3. The upper section of Fig. 3 corresponds to the comparison of A and B of the test data 1 and the lower section of Fig. 3 corresponds to the comparison of A and B of the test data 2. The D row shows the distances, where a darker color represents similitude (i.e., devices are close) and a lighter color represents difference (i.e., devices are at different locations). The GT row corresponds to the ground truth; here the black color means that both devices are together and white means the opposite.
Figure 3. Results comparing devices A-B; test data 1 (above) and test data 2 (below).

The TH1 and TH2 rows correspond to the classification using the two sets of thresholds. Here, the black color refers to a prediction that both devices are together, the white is the opposite, and the gray color represents the undetermined classifications. The THT1 and THT2 rows correspond to using a time threshold on the results of TH1 and TH2 to eliminate the flips.

In Table 1 we can see the accuracy and false negatives of the results using the time thresholds on both TH1 and TH2. The approach achieves an accuracy of over 90 percent due mostly to false negatives in the boundaries. False positives are preferred over false negatives, since we value more a complete audio over power saving, hence false negatives should be kept to a minimum. False negatives mainly occur at the frontiers of devices approaching or separating. The source of error is the time discretization (we used a sliding window of 15s; which can be reduced to obtain a more accurate prediction at the frontier). For this, we suggest not to ask the device to stop recording until we have detected for a given amount of time that both devices are nearby.

One of the main challenges encountered in this experiment is the delay of the incoming audio in the devices due to the positions of the users. Also the reverberation produced in enclosed spaces. An additional challenge was encountered in the older adults’ residence, where we found that the place is mainly connected by a hallway and some audio information could be listened in more than one location. In other words, a shared source of audio (e.g., one person standing in the doorway) may induce a false link between two devices recording in separate rooms. Moreover, two persons may be watching a TV show in separate rooms and a false audio link may be derived.

<table>
<thead>
<tr>
<th>Recording devices</th>
<th>Accuracy (%)</th>
<th>False negatives (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TH1</td>
<td>TH2</td>
</tr>
<tr>
<td>Test data 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-B</td>
<td>97.92</td>
<td>98.08</td>
</tr>
<tr>
<td>A-C</td>
<td>94.37</td>
<td>84.20</td>
</tr>
<tr>
<td>B-C</td>
<td>98.08</td>
<td>97.35</td>
</tr>
<tr>
<td>Test data 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-B</td>
<td>93.75</td>
<td>95.17</td>
</tr>
</tbody>
</table>

Table 1. Accuracy and false negatives in test data 1 and test data 2.
False positives in this experiment imply recording in two phones the same data, while false negatives imply stop recording when two phones are not close.

If they are sending information to the server from different access points, the ambiguity may be solved. On the contrary, this issue can be more complex if they share the access point. Local information about the audio topology of the recording environment may help in the design of a disambiguation method, fingerprinting the audio of popular TV shows may help as well.

**Discussion**

When resources are limited, they need to be regulated in order to help optimize its utilization. This is paramount in environments where resources are particularly scarce. In mobile computing, one of the most valued resources is battery life. Therefore, should we need to save battery, some strategies or alternative algorithms need to be employed (see [8]) and increase the performance of the overall system. Throughout this paper, we have been discussing how, based on actual sensing campaigns, we have been trying to optimize the use of the processing power and storage in order to minimize battery depletion.

In order to step up most of these strategies, much more complex algorithms are needed to make automated, near-optimal decisions during runtime. For instance, the decision whether to process a particular task on a mobile device versus a server depends on a composite interplay of factors that contributes to the complexity of taking these sorts of decisions a priori such as network traffic, current processing load, and the like. These types of decisions are not trivial and can have a great impact on the overall performance of the system. Therefore, they merit careful consideration – and several tests- before implementation since in many cases these decisions can be expensive, and detrimental to the functioning of the system. On the other hand, having a way to sense not only the external variables surrounding the devices but also the current and future internal status of the devices such as the memory available, allocated tasks, battery life, signal strength, and others. This can open the door for far more complex algorithms such as selective task-allocation, which in real-time, based on device capabilities, current and future internal states, and cost-oriented decisions could help decide which device to utilize. For instance, having three mobile devices nearby sensing more or less the same information could result in the following configuration: one mobile device senses all the information (since it has a sensor the others do not e.g., barometer), the second one with better processor and plenty of battery would be in charge of preprocessing data, and a third one will be sending the data over the network. However, different conditions could have resulted in a different configuration.

Much simpler strategies can been applied such as sharing location with other devices in the vicinity either to save battery or because some devices do not have a GPS, but still can obtain their location through Bluetooth. Similar strategies can be also employed to reduce uncertainty, that is, when a device is not sure it ‘heard” (i.e., sensed) something, it could check with a nearby device to see what it heard. Our approach to avoiding duplicate recording of audio could be used either to stop further processing or to deter a device from sending the audio over the network i.e., more
than one device is listening, but only one classifies and sends the results to a server.

Conclusions and future work
We discussed an extension to InCense, a research kit for collecting behavioral data from populations of mobile phone users. This extension was mainly aimed at supporting collaborative sensing campaigns in order to optimize resources such as battery, storage, or bandwidth. In energy-craving environments, like mobile phone sensing, some of these resources are valued and thus they pose serious challenges in sensing campaigns wherein significant amounts of data are to be collected (i.e., audio or video) or the battery is rapidly depleted (e.g., GPS). Some of the challenges in the sensing campaign presented are quite common in mobile phone sensing. There is no risk in loosing data, when a phone is turned off, the server is aware of who will be the proxy for that particular recording.

References