Ubiquitous Inference of Mobility State of Human Custodian in People-Centric Context Sensing

Mattia Gustarini
University of Geneva
mattia.gustarini@unige.ch

Katarzyna Wac
University of Geneva
katarzyna.wac@unige.ch

ABSTRACT
People-centric sensing using people's smartphones offers new research opportunities for large case studies. It presents many challenges, *e.g.*, efficient capture of person's mobility, understanding of context changes and preservation of user privacy. We propose an accurate and energy-efficient method able to capture user's mobility, thus the context changes, while preserving his/her privacy. Our solution can be applied to systems that aim to efficiently sense context on smartphones to study large scale phenomena or perform location management.

Author Keywords Smartphones, Mobility, Battery, Context, Sensing, Efficiency, Performance.

ACM Classification Keywords C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; C.2.4 [**Distributed Systems**]: Distributed Applications.

General Terms Algorithms, Design, Experimentation, Performance.

INTRODUCTION
A growing number of market reports shows that smartphones are becoming pervasive technology, affordable for masses [1]. Each smartphone is equipped with a multitude of sensors, thus possibly enabling a new large-scale research and data collections in the urban environments [2]. The phone owners become sensor *custodians* [3], that use the smartphones in their daily life, and additionally are able to collect significant and meaningful context data for, *e.g.*, the environment they are in. This rise of co-called *people-centric sensing* [3] brings to our attention critical success factors related to: (1) knowledge of the context in which the phones are sensing, (2) respect for the phone's variable availability of limited resources (*e.g.*, device battery, network connectivity), (3) knowledge that phone owners are moving, (implying a need for adaptive sensing mechanisms to efficiently capture the phenomena of interest), while (4) ensuring the privacy of the phone owner. In this paper we focus on these challenges, hence we aim to infer if the phone owner, *i.e.*, sensor custodian is *moving* (*i.e.*, changing locations in time) using as few phone resources as possible - avoiding "power-hungry" sensors (GPS, accelerometer, *etc*.) and preserving the person's privacy. Our goal is to ubiquitously, accurately and in energy-efficient manner infer mobile-fixed context of the human custodian, and to enable *dynamic* changes of the sensors' duty-cycle length, while ensuring a continuous sensing of a phenomenon that is influenced by the mobility of the sensor custodian (*e.g.*, as in the ad-hoc sensing by Lane et al. [6]). When the person is moving, the observed phenomena may mutate faster requiring a different sampling rate, not to lose important data, but when the user is fixed, the rate may be reduced to preserve the phone battery lifetime.

UbiComp'12, September 5-8, 2012, Pittsburgh, USA.
ACM 978-1-4503-1224-0/12/09.

In CenceMe [4] the authors presented how the choice of the sensors' sleeping time and the length of the duty-cycle is crucial for the trade-off between the quality of the data sensed and the phone's battery lifetime. From set of experiments, they have derived a set of (pre-defined) parameterizations for the application. Wang *et al.* [5] present a framework to efficiently recognize the user state (*e.g.*, activity). They proposed to use only power efficient sensors to identify the change of the state and only at that moment enable the "power-hungry" ones to identify the details of the new state. They stated that the designer of a context-aware system must carefully select the right duty-cycle to avoid draining the battery, even with less energy-consuming sensors. Also in this case, the best settings were pre-defined via experimentations.

An other field of interest for our approach relate to location management, like for example one by Papandrea [7], that may provide location updates only when needed. Authors of [7] use different features derived from accelerometer, and classify the fixed/mobile state towards constructing a mobility model based on points of interest. With our algorithm, we contribute to this matter without the use of the accelerometer, which, as we show in this paper, influences considerably the battery lifetime.

The next section of this paper explains our approach and motivates algorithm design, followed by the description of conducted experiments and some preliminary results, which are discussed and concluded in the last sections of this paper.

MOVING OR FIXED? STUDY APPROACH
With our ubiquitous approach we aim to infer in real-time if a smartphone user is in a fixed location or is moving. We

base our prediction on data already collected in the life cycle of the operational phone's OS. We assume no knowledge of the user's past moving/fixed states, *i.e.*, no knowledge upon the history of user's mobility. We employ the algorithm that: (1) retrieves the current network cells' information from the smartphone OS, (2) derives main mobility features and (3) uses them to classify the user's state using a pre-trained classifier. These steps are repeated at a fixed frequency. The algorithm is currently implemented on the Android OS platform.

Before explaining details of each step of the algorithm it is important to note, that the candidate algorithm, as well as its parameters (*e.g.*, time window, features, and an array length) cited in the following subsections were inspired by the literature study and then derived experimentally. Experimentations were based on a long trace of raw mobility data gathered along a travel of multiple hours, and containing data from different urban and suburban locations and in different mobility mode, using *e.g.*, train, tram, bus, car and walking. This trace was used in initial experimentation, to simulate the algorithm with different parameters. Only the parameters leading to the most accurate, energy-efficient algorithm were chosen, as presented. We do not provide details of the initial experimentation due to the space limitations of this paper.

Collection of Network Cells Raw Data

The raw data used by the algorithm is as follows. We retrieve the phone's cellular network (*i.e.*, 2G or 3G) *scan* data, *i.e.*, `cell ids` (CellID) and `signal strength` (RSSI) of the current network's cell and the visible neighbors cells every 2 seconds, in 7 (consecutive) scans (the frequency chosen experimentally to capture variation of cell's RSSI, as derived from assumptions on WIFI RSSI [8]). We retrieve the CellID's RSSIs data from phone's OS and start constructing a table mapping each CellID to an array of its 7 (undefined at first) RSSI values. Each scan operation results in filling out a corresponding RSSI index in the array for a given CellID. We observed that a CellID may be not always present on all the 7 scans. The 7 scans constitute a *measurement*, based on which mobility features are derived. The consecutive windows of 7 scans are not overlapping, *i.e.*, we derive new features each measurement, and a new one is launched every 20 seconds (experimental value).

For each CellID, in addition to the array of its RSSI, we keep its co-called *generation counter* and its *consecutive generations counter*. The first time a CellID provides a value of RSSI in a measurement; its *generation counter* is updated to the value of a *global counter* and its *consecutive generations counter* increases by one. The global counter keeps the *global* number of conducted measurements since the application start. In this way we are able to track which are the CellIDs that contributed at least once in the current measurement.

From Raw Data to Classifier Features

After the collection of the raw data we attempt to obtain the features that are going to be used to label user fixed/moving state. For *each* measurement separately, we derive the following three features using the CellID RSSIs' array and its generation counter, as described in detail in the following sections. At this point, it is important to notice that we tested our system with three variations of each of the features - we have run experiments using the average, the median and the variance of the derived features and for each of them we have selected the one leading to the best accuracy and energy-efficiency algorithm.

1st Feature: Median Overall Live Time of Cells (MOL)

This feature relates to the "visibility" time of the CellID and it is the outcome of a) own experiments related to quantification of the time each CellID is visible when moving and b) from the work of Mun *et al.* [9]. When moving, the time a CellID is visible is shorter than when in a fixed position. Hence, we derive MOL from the median of all CellIDs' *consecutive generations counters*. CellIDs that have disappeared since 5 measurements become obsolete and are omitted. The low values of MOL are observed when moving, and high MOLs, when in a fixed location.

2nd Feature: Average Euclidean Distance of Signals (EUC)

We defined this feature inspired by the work of Sohn *et al.* [10] using Euclidean Distance (ED) of signals to recognize offline mobility modes (*i.e.*, walking, biking, driving). We first take the pair of first two RSSI for each CellIDs and using the formula from [10] we obtain the first ED for RSSI over all the CellIDs for first 2 scans. We repeat the same procedure five times by moving across the CellIDs' RSSI arrays, deriving ED for the RSSIs for each two consecutive scans. Finally we take the average of the six ED derived from the table and we acquire EUC for the current measurement. The EUC decreases when the user is fixed.

3rd Feature: Average Fast Wavelet Transform Distance of Signals (WAV)

This feature is inspired by the general algorithm of Muthukrishnan *et al.* [8] - originally designed for sensing offline the motion using WLAN RSSI. To obtain this feature, each measurements' CellIDs RSSI array is transformed using the Discrete Wavelet Transform. For a given scan within a measurement, transformed RSSIs of CellIDs are compared - we retrieve the minimum and maximum values of RSSI and we compute the range between the two. Then we average all the ranges within a measurement, to derive the WAV feature value. From the preliminary experiments and from [8] we confirmed that the average of the ranges increases when the person is mobile.

Moving or Fixed? Inferring the User State

The final step of our algorithm consists on the inferring of the person state class using the derived features. The classifier has been build using the Weka[1] package and its

[1] http://www.cs.waikato.ac.nz/ml/weka/

model was built offline using the raw, pre-labeled hours of mobility data collected for the simulation of the algorithm. We chose a tree-based classifier model that is deployed on the mobile phone and at the OS runtime it is used to derive inferences upon the current user state being either `mobile` or `fixed`. We are also investigating other classifiers evaluating their speed, accuracy and algorithm complexity. Since the inference of the state is done on the mobile phone (*i.e.*, no data is shared/send to a server) the privacy of the owner is respected.

EXPERIMENTATION

In this section we present first experiments with our algorithm called "MobilitySensor" deployed on an Android OS phone of real user being mobile in the real mobile environments. Our goal is to investigate its accuracy and battery consumption against the use of build-in phone sensors to infer users' mobility state.

Other Inference Approaches & Experimental Setup

To perform the experiments we created four distinct approaches/applications for inferring the user state: (1) based on phone' 3D accelerometer; (2) based on the network-based location; (3) our "MobilitySensor" inferring the state from the pre-trained model based on CellIDs' RSSIs; and (4) based on the phone's GPS.

For (2) and (4) the mobile/fixed label is derived based on comparing the current location with the previously known location. The distance traversed is computed and the sensor accuracy for the previous and current location is noted. If the distance traversed is greater than the worst accuracy, the user state is `mobile`, otherwise `fixed`. If one of the two locations is not available the state is `unknown.`

The mobile/fixed inference from 3D phone's build-in accelerometer was derived as follows. The sensor's sampling frequency was "normal" as provided by the OS (~40Hz). For each window of raw data collected, a mean acceleration vector was derived; if the value is below 0.5, the state is `fixed`, otherwise `mobile`. These bounds were derived experimentally by analyzing the data acquired when phone is in the pocked when the person is fixed, as confirmed in [11].

All the experiments ran using a HTC Sensation with Android OS v. 4.0.3 by one user (discontinuously) for 5 days. To establish the ground truth we created a widget (along the Experience Sampling Method) that the user was instructed to use to timestamp changes in his mobility state. We collected the battery usage statistics per each application separately, as derived from the statistics provided by the phone's OS detailed battery statistics. Every time the OS pushed information upon battery status change, given a twist on the official API, we managed to derive the current battery details; we have not actively pulled this information from the OS.

Each application is activated once every one minute, to infer a user state; this frequency is adapted to the "slowest" GPS-based approach, requiring 40 sec of data collection, *i.e.*, relying on the AGPS timeframe to obtain a stable location with high probability. Our algorithm in theory is able to provide an inference every ~20 seconds (given the measurement time of 14s, features derivation and state classification). For the purpose of this evaluation, our algorithm runs 3 times per a minute, but it provides the inferred state only once. The operation of approaches (1), (2) and (4) is divided in two parts: data collection (40s) and state inference (20s). Once all the algorithms infer the user's state, their applications "sleep" until the next minute bound. We have collected 539 inferences (*i.e.*, minutes) for all approaches, and 750 battery measurements. The user was 52% fixed (*i.e.*, 48% mobile), which leads to a random guess (*i.e.*, fixed state) at a 52% chance level.

Results

Figure 1 provides the results for battery efficiency for all 4 approaches. The results show that GPS based approach is confirmed to be the most "power-hungry". At the second place we find the accelerometer approach followed by the MobilitySensor and then the network one. In general for all approaches, the majority of battery status change events and corresponding energy usage steps are bounded in a short range, assuring a low jitter. Table 1 presents the results for approaches' accuracy; assuming an algorithm could result in a correctly inferred state, or incorrectly inferred state or `unknown` state (as explained, applicable for GPS and network-based approaches only). The accelerometer approach is slightly more accurate (72.96%) than the MobilitySensor (71.24%), but comparing their energy efficiency (Figure 1), we conclude that MobilitySensor is of a higher performance. GPS and network-based approaches are performing poorly in terms of accuracy of correctly inferred user states (38.12%, and 26.43%), both placing themselves below a threshold of random guess (52% chance). In addition to the accuracy, Table 2 presents the

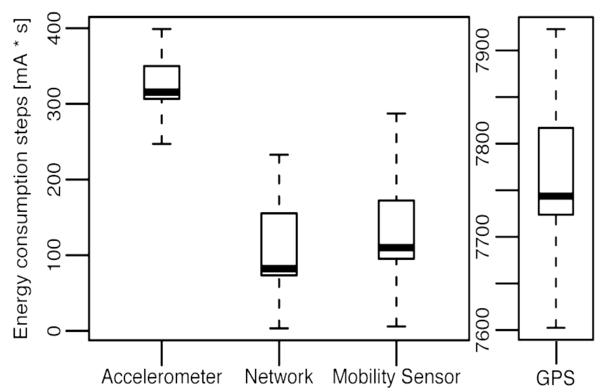

Figure 1. Approaches' Battery Consumption

Approach	Correct	Wrong	Unknown
accelerometer	72.96%	27.04%	0.00%
network location	26.43%	11.65%	61.92%
MobilitySensor	71.24%	28.76%	0.00%
GPS	38.12%	6.82%	55.06%

Table 1. Approaches' Accuracy Results

confusion matrix for mobile-fixed states, *i.e.*, percentage of cases when the `fixed` state is inferred as `mobile` (or inversely). Accelerometer and MobilitySensor perform similarly. GPS is more accurate when inferring the `mobile` state, while the network-based approach has a higher accuracy when inferring if `fixed`.

Approach	**Fixed**	**Mobile**
accelerometer	42.47%	57.53%
network location	34.67%	65.33%
MobilitySensor	46.45%	53.55%
GPS	69.64%	30.36%

Table 2. Approaches' Confusion Matrix

DISCUSSION

In this section we discuss the limitations of each approach used in our study as well as the results presented. The least performing approaches in terms of accuracy are GPS and network-location based approaches. The GPS is underperforming when `fixed` which can be explained by the fact that when fixed, usually one is inside of a building where GPS signal is not available thus it is impossible to get the current user location. The network-based approach poor performance stems from the inaccuracy in the user location's estimation when mobile. The change of position cannot be accurately evaluated, unless it is fast enough such that the radiuses of inaccuracies of separate positions are not overlapping. In the case of the accelerometer-based approach, the confusion stems from the situation when user is performing a physical effort activity while being in a fixed place. Also, when a user is mobile, *e.g.*, sitting in a tram, there are not enough changes in the acceleration, which results in an inaccurate inference. As done in [7], or following the findings in [11], to increase the accuracy of this approach a more detailed analysis and classification is required to distinguish physical activities – implying possibly a shorter sampling window than the one used in our experiments, but possibly resulting in lower battery efficiency, comparing to the MobilitySensor.

MobilitySensor approach is solely based on CellID and RSSI information already available in the OS. Despite its simplicity, the method exhibits high accuracy and low battery usage. We have also identified its shortcomings, for example lack of CellID and RSSI information, when the user is out of coverage of a network, as well as possible sources of errors in the raw data collection stemming from cellular network characteristics. When user is fixed, the method may be inaccurate because some cellIDs are performing what it is call ping/pong behavior [9] - they disappear at random influencing the values of features used for classification. We have already started to implement a solution to avoid this phenomenon. The inaccurate inference when `mobile` stems from the number of CellIDs captured. With less than 3 cells the RSSI variation is not large enough to reflect mobility in the EUC and WAV features. The MOL feature is also influenced in this case, but only when in a big open space (*i.e.*, suburbs) where the coverage area of network cells is larger and there are few of them. This situation seems more difficult to solve experimentally so far. In our current research, we investigate the accuracy of the classifier model in these two above-mentioned special situations. We conclude that new features are required - capturing current network type used, GSM (2G) or UMTS (3G), and the number of CellIDs available. In the future we would like also to conduct larger scale experiments involving more users (currently we present result for one single user) and an improved version of the accelerometer approach (as described above). We aim to have more statistically significant data to show the performance of this approach.

CONCLUSIVE REMARKS

In this paper we propose a novel, real-time ubiquitous approach to infer mobile phone users' mobility state solely from the cellular network data available in their operational phone's OS. No data is shared to external services respecting the user's privacy. We propose how it can be used in the context of people-centric sensing as well as a component of new energy-efficient location managers. We presented the main steps of the algorithm together with preliminary performance results, although involving only one user collecting data for 5 days, in terms of battery consumption and accuracy. Finally we discussed its limitations and stemming from that - future work areas.

REFERENCES

[1] IDC, Smartphone Market, 2012. [Online]. http://www.idc.com/. [Accessed: 09-May-2012].

[2] A. T. Campbell et al., People-centric urban sensing," *WICON*, pp. 18, 2006.

[3] A. T. Campbell et al., The Rise of People-Centric Sensing, *IEEE Internet Computing*, 12(4):12-21, 2008.

[4] E. Miluzzo et al., Sensing Meets Mobile Social Networks: CenceMe Application, in *ACM conference on Embedded network sensor systems*, 2008.

[5] Y. Wang et al., A framework of energy efficient mobile sensing for automatic user state recognition, *MobiSys*, p. 179, 2009.

[6] N. D. Lane et al., Ad-hoc sensor network A Survey of Mobile Phone Sensing, *IEEE COMMAG*, 2010.

[7] M. Papandrea, Enhanced Localization Solution, *PerCom*, pp. 247-252, 2012.

[8] K. Muthukrishnan et al., Sensing motion using spectral and spatial analysis of WLAN RSSI, in *Conf. on Smart sensing and context*, pp. 62-76, 2007.

[9] M. Mun et al., Parsimonious mobility classification using GSM and WiFi traces, *HotEmNets*, 2008.

[10] T. Sohn et al., Mobility detection using everyday GSM traces, *UbiComp 2006,* pp. 212–224, 2006.

[11] D. Fuentes et al., Online motion recognition using an accelerometer in a mobile device, *Expert Systems with Applications*, vol. 39, no. 3, pp. 2461-2465, 2012.