Inferring Correlation between User Mobility and App Usage in Massive Coarse-grained Data Traces

Zheng Lu
Electrical Engineering and Computer Science
University of Tennessee
zlu12@vols.utk.edu

Yunhe Feng
Electrical Engineering and Computer Science
University of Tennessee
yfeng14@vols.utk.edu

Wenjun Zhou
Business Analytics & Statistics
University of Tennessee
Knoxville, TN 37996
wzhou4@utk.edu

Xiaolin Li
School of Management
Nanjing University
Nanjing, China
lixl@nju.edu.cn

Qing Cao
Electrical Engineering and Computer Science
University of Tennessee
cao@utk.edu

ABSTRACT

With the rapid growth of smartphone usage, it has been more and more important to understand the patterns of mobile data consumption by users. In this paper, we present an empirical study of the correlation between user mobility and app usage patterns. In particular, we focus on users' moving speed as the key mobility metric, and try to answer the following question: are there any notable relations between moving speed and the app usage patterns? Our study is based on a real-world, large-scale dataset of 2G phone network data requests records. A critical challenge is that the raw data records are rather coarse-grained. More specifically, unlike GPS traces, the exact locations of users are not accurately available. We might infer the approximate location of a user according to his or her interactions with the cell towers, whose locations are known. We address the challenge of user speed estimation by proposing a novel methodology to filter out noises, so that we can achieve reliable and fine-grained speed estimation. We then examine several aspects of mobile data usage patterns, including the data volume, the access frequency, and the app categories, to reveal the correlation between these patterns with users' moving speed.

1. INTRODUCTION

In the past decade, the use of smartphones has grown significantly among consumers. According to a recent report [5], there are 3.4 billion smartphone users worldwide, and the accumulated mobile data traffic has reached 120 ExaByte in 2015. One critical reason for this explosive growth is the popularity of smartphone apps, such as those served by Google Play and Apple Store, whose number has exceeded 1.5 million by July 2015 [22]. It is estimated that people spend as much as 30 hours monthly on these apps on average, a growth of over 65 percent compared to 2013 [13].

Consequently, recent research has invested considerable effort to understand smartphone app usage behavior, as such understandings can help app developers and mobile advertisers tremendously [31, 33]. In the previous work, both temporal patterns (e.g., individual app usage histories) and spatial patterns (e.g., location contexts) have been extensively studied [11]. Their results have enabled novel applications, such as smartphone app launching prediction services [32] and location-aware event recommendations.

In this paper, we focus on one less investigated feature, user mobility, and investigate how this feature correlates with the usage patterns of smartphone app users. Understanding such correlations, if any, could provide useful contextual information for relevant and accurate app recommendation and ad delivery. For example, if we find out hiking hobbyists use certain apps considerably more often, then such apps may be more useful venues for ad delivery for equipment makers for hiking activities.

Unfortunately, previous work on this topic has only investigated this problem in highly limited and controlled contexts, and by taking account into the usage history of a small set of users. For example, a few works have addressed the problem of transportation mode inference, where the goal is to find out whether a user is riding a bus or taking a taxi, among other possibilities. Such work usually assumes that additional hardware (e.g., GPS, sensors) is available, and is carried out for a small group of users in controlled experiments [15, 25, 34, 2, 23, 18, 29, 16]. Later work suggested that it may be possible to use cell tower communications to monitor users' mobility indirectly [17], where efforts have been focusing on inferring users' trajectories [12, 9] or transportation mode [26, 4, 1] only using cell-phone traces (e.g., Call Detail Records, handover data) that do not directly contain location information. Such approaches are more scalable, as they do not require additional hardware resources, and better respect users' privacy. The limitations of these approaches, however, are that they are usually small-scale by nature, and usually has ground-truth data collected for a user as validation methods for their approaches.

Our work is following this later line of research of using

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

large-scale cell-phone tower traces. However, our dataset and the corresponding methodology are significantly different. First, our dataset consists of a truly large population, where we have access to mobile data access histories of millions of users in three cities that cover thousands of square miles. The number of users is perhaps more than quite a few countries in the world. Second, due to privacy concerns, the dataset is fundamentally coarse-grained, meaning that we do not, and can not, collect the ground truth information for these millions of users. Therefore, novel data processing methods are urgently needed. Finally, our research goals are to reveal large-scale, population-level correlations, if any, between user mobility and app usage patterns, a goal that has not been addressed in any of previous research work. We emphasize, however, due to the second limitation on the absence of ground truth, all our conclusions are, at best, educated guesses that are based on real-world data. We believe such results are meaningful and insightful for a wide range of target people: app developers, ad distributors, network operators, and end users.

We address the following two challenges in our work. First, to infer user mobility with cell-phone traces, we need to filter the location history to obtain accurate estimates. In our dataset, the only location information available is the communication history between a customer and a cell tower. Fortunately, we have the precise locations of each cell towers, and by communication principles, we know that a user's phone typically contacts the tower with the best signal reception (usually the nearest one). We have surveyed the previous work on estimating trajectories based on similar datasets [21, 8, 30, 12, 9] or finding mobility motifs [28, 6], but we could not find one that suit our needs as we find their results are clearly still too coarse-grained. One reasons is the dataset difference: their data mostly contain users who perform daily commute or city to city long distance trips, while our data are in dense urban areas where users employ a mixture of transportation modes ranging from walking, bicycles, to buses and cars. Railway transportation is not present in our dataset. Therefore, based on these concerns, we need to develop novel methodology to estimate more complicated and fine-grained user mobility trajectories for our target dataset. Second, to correlate the usage history of apps with mobility patterns successfully, we need to develop a tradeoff between the most popular apps and sparsely distributed ones. More precisely, we find that a majority of users will use those "heavy-hitter" apps no matter what their mobility patterns are. Therefore, inferring such correlations are less meaningful. Instead, we should focus on those app groups where data exhibit differentiated popularity for different groups of users with different moving speeds, a task that is considerably more challenging than simply performing correlation analysis between all apps and users naively. Therefore, our methods need to be customized for this application analysis task.

The main contributions of this paper can be summarized as follows.

- We design and evaluate a novel methodology to infer user speeds with cell-phone traces with low location accuracies. Compared to existing approaches, this methodology achieves far better and fine-grained estimation with adjustable confidence levels. Specifically, to overcome the problem of location accuracy, our methodology involves steps to segment traces by pass-boundary events (i.e., when a user establishes a new connection with a different tower, and performs intra-cell level zooming and analysis to calculate distance estimates. This method is also robust against issues caused by the uncertain nature of wireless communications, e.g., a user located in the overlapped communication coverage area of multiple towers may randomly communicate with each tower, causing cell oscillations that other simple methods cannot easily address.
- With the more accurate speed estimates, we are able to study the correlation of user mobility with app usage patterns in a population scale in an uncontrolled, real-world environment. The results are novel in that no previous work, to the best of our knowledge, has gained similar insights or reported findings in this aspect. Our revealed correlations of user speed and mobile data access patterns include the data volume, the access frequency, the share of each smartphone app category in the total mobile data traffic, and user preferences of apps under different transportation modes.

The rest of this paper is organized as follows. In Section 2, we describe previous works on user mobility inference and geospatial app usage patterns. Section 3 defines our problem and provides details on the mobile data access trace we use in this paper. We describe our speed estimation methodology and design in Section 4. Section 5 explains our findings on correlation of user speeds and mobile data access patterns. Finally, we conclude our work in Section 6.

2. RELATED WORK

In this section, we summarize recent literature on smartphone apps, user mobility, and geospatial analysis of mobile phone apps data.

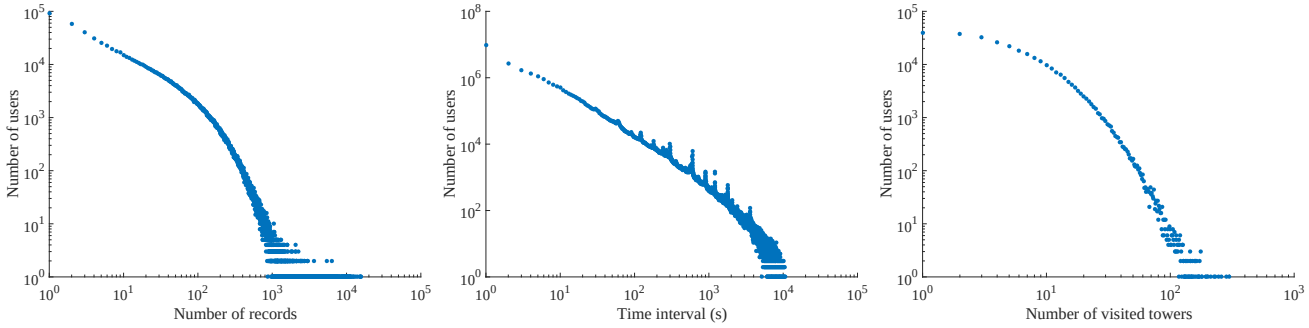
2.1 Smartphone Apps

To study the smartphone app usage behavior of a large group of users, previous work has analyzed mobile data traces generated by smartphone apps in studies of various scales. [33] studied the mobile user behavior by focusing on data usage, mobility pattern and application usage. In [31], the aggregated spatial and temporal prevalence, locality and correlation of smartphone apps at a national scale is investigated, by analyzing the mobile data generated by smartphone apps. Unlike our work, these previous work have not studied the relation of mobile user behavior with more complex user mobility, i.e., user speed.

2.2 User Mobility

Using GPS [15, 25, 34, 2, 23, 18, 29, 16] and embedded sensors [15, 27, 20, 10, 24, 16, 7], a separate body of research is able to use smartphones to infer user mobility patterns accurately in small-scale, controlled experiments, such as inferring transportation modes. Most of these works formulate the problem as a classification problem, where common challenges involve data segmentation [15, 25, 34, 2], feature selection [34, 2, 27, 23]. Multiple methods, such as SVM or linear regressions, are developed to achieve the best accuracy.

Although GPS and sensors are well suited for small scale experiments, they are not scalable as users typically do not want their GPS traces to be shared with others. In recent



(a) Number of Records (b) Avg. Time Interval (c) Number of Towers Visited

Figure 1: Dataset characteristics.

work [17], it is revealed that there is a great potential for using cell-phone data traces such as Call Detail Records (CDRs) for user mobility inference. A large body of research literature exists applying this method for inferring user's trajectories [21, 8, 30, 12, 9, 4, 1] or mobility motifs [28, 6]. For example, [12, 9, 4] inferred user trajectories from cell-phone traces based on how likely a specific route can lead to similar tower access sequences stored in the data traces. In another work [26], it aims to classify a user's transportation mode by clustering travel time distribution. Finally, researchers [1] also proposed approaches that can deal with common zig-zag problems in inferring user mobility from smartphone traces. Different from these existing methods, however, our approach take advantage of scalability of cell-phone data traces and achieves fine-grained user mobility inference on top of it.

2.3 Geospatial App Usage

Studying correlations between app usage and features extracted from phone traces is not new in the literature. Previous work has studied relations of human mobility and social networks using geo-spatial features. For example, [3] found that the short-ranged travels are periodic and not likely to be related to the social network structures, while long-distance travels are heavily related to the social network status of a user. Based on these findings, a model was proposed to predict dynamics of future human movement with a high accuracy. Follow-up works such as [14] studied a similar problem with a different dataset. [19, 33] studied the geospatial relation of the app usage volume. Their works mostly studied the spatial correlation of the smartphone usage, while the user mobility's impact on app usage is still a missing piece of these works. Finally, [11] studied how the proximity, the location and individual differences (e.g., personality) can effect the user's mobile data usage. However, analysis on much complex user mobility such as user speed is still a missing pieces in these works.

3. PROBLEM SETTING

In this section, we provide a description of the dataset, followed by an example of a user's data.

3.1 Dataset Description

Our dataset contains mobile data access history of all active users (during a three-hour period) of a major mobile carrier in three cities of China. For each user, all data request records during the study period are available, where each record consists of the following information:

Figure 2: Communication in a city area.

User ID : the identifier of a user, a hashed value for anonymity;

Tower Location : the geo-coordinates of a cell phone tower with which a user has established handshake and communicated;

Timestamp : the Timestamp consists of the date and time of a mobile data access record;

Data Access Features : contain the app identifier that initiates the data communication, and data volumes.

The dataset is a collection of the aforementioned mobile data access records provided by a cellular network operator in China, collected from three mid-size cities, including both urban and suburban areas, during a three-hour period in the early evening (6pm - 9pm) in 2014. The cities are anonymous in this paper. The dataset includes more than 58 million mobile data access records with a total volume of more than 720 gigabytes, which covers all cell phones that were actively exchanging data with a total of 5199 cell towers in the area during the observation period. The number

of unique users included in this dataset is identified as 0.9 million, by removing duplicates. The total active time of all users accumulates to more than 1 million hours. Figure 2 shows the mobile data access in a city area of our dataset.

3.2 An Example User's Traces

To provide a clear view of our data, we visualize a user's records from our dataset in Figure 3 as a running example. Suppose that the user was taking the path (while using the cell phone) shown with the dashed line. In particular, he started by walking from location 1, to location 2 where he waited for the bus. After a few minutes, he got onto the bus, which took the path towards location 3. Even though we did not know the actual path of the user, his locations could be inferred by the nearby towers to which his data were sent to.

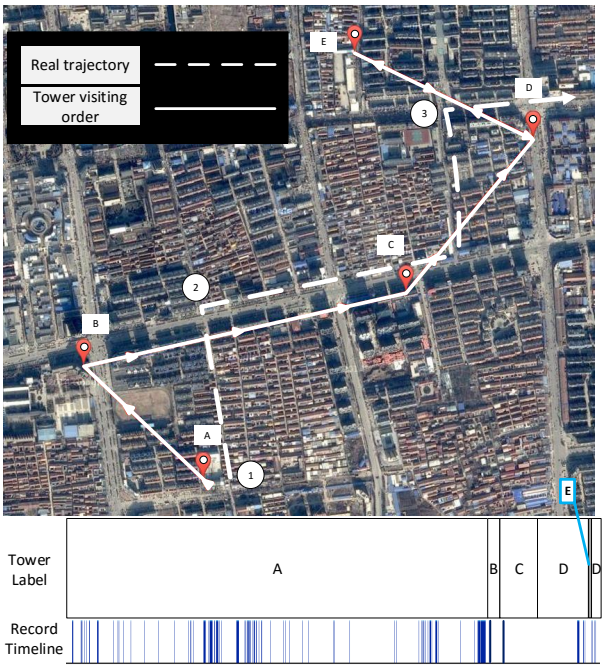


Figure 3: Example data access activities of a user.

Since the cell tower locations are all known, we can display the tower locations in a map that were visited by the user. We use markers to show tower locations and arrowed lines to show the sequence of visiting.

The bottom part of the figure shows the timeline of the user's data access records with pulses. We also show with which tower the user has communicated for each mobile data access record, by providing tower labels above the pulses. Therefore, for this particular user, he/she communicated with tower A for a quite long time, and shortly connected to tower B before switching to tower C. After a while, the user was found in tower D's coverage area. Then he connected to tower E for a very short time as location 3 is equally close to both tower C and tower D, i.e., a possible cell oscillation, and in the end switched back to tower D.

3.3 Data Preprocessing Findings

We first preprocess the data and analyze the characteristics of mobile data access patterns. Distributional characteristics are visualized in Figure 1. In particular, the number of records per user, the average time intervals between consecutive records, and the number of towers visited. We found that our dataset has a highly skewed distribution on the number of records per user, as shown in Figure 1(a), and the time intervals between consecutive records, as shown in Figure 1(b). Here, a higher record density, i.e., more records for a user in a time unit, indicates a better performance to infer user mobility even when the trip length is very short, as we can obtain a better granularity by analyzing these records. Actually, it is the case that most user only traveled a very short trip in terms of the number of visited towers according to Figure 1(c).

Note that our dataset differs from commonly used mobility datasets used in existing work. Compared to movement trajectories like those captured by GPS, we do not know the exact locations of the users, and we only know a user is located nearby a tower to communicate with it. Furthermore, compared to other datasets with call detail records (CDR), our dataset is drawn from a region with more densely populated customers, where each may adopt different mobility methods such as walking, driving, or taking buses. Such differences make it harder to accurately estimate user speed based on existing methods.

4. SPEED ESTIMATION

In this section, we systematically describe our methodology for estimating user mobility speed using coarse-grained tower communication records and timestamps.

4.1 Methodology Overview

Our methods consist of multiple steps, where we first decompose traces of each user into segments to zoom into intra-cell speed estimation. Next, we estimate the distance and the travel time for each segment, where we employ a distance lower bound to filter out low-confidence estimates. In practice, such estimates are usually too noisy to be meaningful or reliable. Finally, we demonstrate how to compensate for speed estimation errors. Figure 4 shows the structural overview of this methodology. The raw data parser on one end of this figure gathers data access records by users, and sorts records of each user by time. On the other end, a list of tower locations from the mobile data access traces is extracted. Note that the system assigns a list for each city during processing steps.

After we have parsed raw traces, we next process them in different steps in parallel. In one of the next steps, we analyze traces of each user, and generate pass-boundary events (BPEs) with the timestamp and location estimates of each record. Based on these events, we can estimate intra-cell travel distances and time accordingly.

In the second sequence of steps, we process the tower list for each city, by generating a Voronoi diagram based on tower coordinates. Here, Voronoi graphs are used to simulate the tower coverage map, based on which we calculate all intra-cell boundary-to-boundary distance lower bounds. We keep such bounds in a separate list for lookup needs.

Finally, at the end of both processing sequences, we aggregate their results to estimate each user's speed distributions. We observe that for some segments, we do not have sufficient location information to accurately estimate a particular user's speed. Under such scenarios, we develop a compensation step where we try to infer the most likely

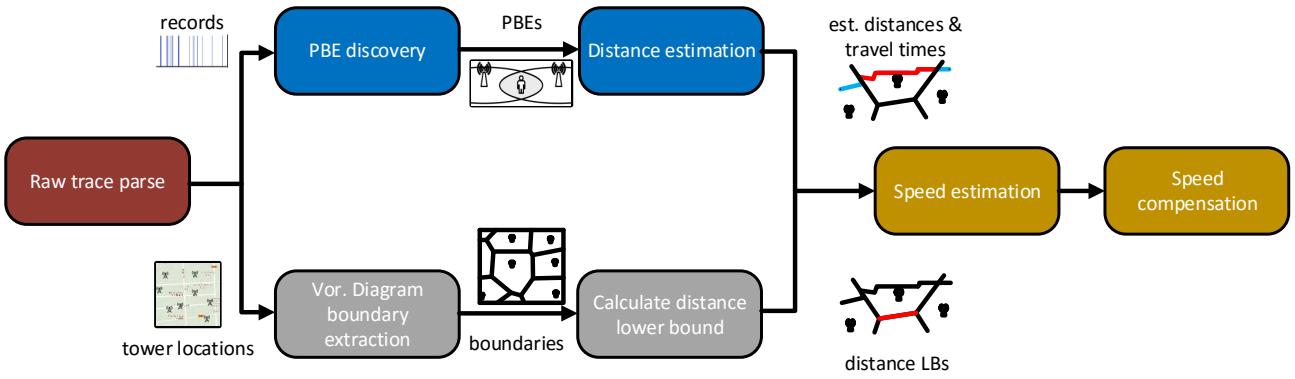


Figure 4: Speed estimation system overview.

speed based on speed distributions of this user in adjacent segments. The assumption is that one user will not change speed too much in short distances. We next discuss each component in more details in the following sections.

4.2 Pass-Boundary Events

As our dataset is coarse-grained, we only have location information of the towers with which a user communicates. As a user could be anywhere inside the tower's coverage area, we need to infer their speeds by exploiting multiple coverage areas. To this end, we first decompose the trace into segments, through what we call "pass-boundary events" (PBE).

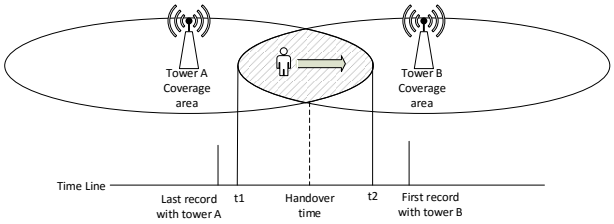


Figure 5: A pass-boundary event.

Formally, a PBE is defined as when a user moving from one tower's coverage cell into an adjacent tower's coverage cell. There are two properties related to a PBE: first, each PBE has a boundary area, which is the overlapping coverage area of two towers; second, each PBE is associated with a time period of the user spent on crossing the boundary area. For example, in Figure 5, the PBE event is associated with a boundary as the shadowed area, while its associated time period is (t_1, t_2) for entering and leaving this shadow area.

We now describe the algorithm on extracting BPEs from the mobile data access traces. For arbitrary two consecutive records r_i and r_j of a user with l_i and l_j as their location estimates respectively, if $l_i \neq l_j$, we define a PBE, denoted by $P_{i,j}$ as follows:

$$P_{i,j} = (r_i, r_j)$$

We denote the boundary of $P_{i,j}$, which is the overlapped area, as (l_i, l_j). We use $({t_i}^{last}, {t_j}^{first})$ as an estimate of the time period of entering and leaving $P_{i,j}$, where ${t_i}^{last}$ is the last record timestamp in l_i and ${t_j}^{first}$ is the first record timestamp in l_j. The length of the time period of $P_{i,j}$ is bounded by ${t_j}^{first} - {t_i}^{last}$.

Algorithm 1: Data segmentation

Data: $Trace$: mobile data trace arranged by users u with record entries e sorted by time. Each e has location estimate l_e and timestamp t_e
Result: R: segments r arranged by users.

```
for each u in Trace do
    l_c ← ∅, r ← ∅, e_lastend ← ∅, e_start ← ∅, e_end ← ∅ ;
    for each e in Trace[u] do
        if l_c ≠ ∅ and l_c ≠ l_e then
            l_r ← (l_{e_lastend}, l_c, l_e), t_r ← (t_{e_start}, te_end) ;
            append r to R[u] ;
        end
        if l_c = ∅ or l_c ≠ l_e then
            e_lastend ← e_end, l_c ← l_e, e_start ← e
        end
        e_end ← e ;
    end
    l_r ← (l_{e_lastend}, l_c, l_e), t_r ← (t_{e_start}, te_end) ;
    append r to R[u] ;
end
return R
```

Once BPEs are defined, we use them as reference points to decompose mobile data records of each user into segments. The detailed segmentation algorithm is shown in Algorithm 1. Specifically, the decomposition works as follows: we first generate the BPEs for each user, and then we consider all records of a user between two consecutive PBEs as one single stretch of *continuous stay* as such records should be communicating with the same tower. Therefore, they should share the same location estimate. Since we do not have observations on the intra-cell trajectories of user mobility, we consider the user to have the single constant speed for each stretch within a cell, i.e., between two PBE events. To estimate this speed, we use two consecutive PBEs. Note that, however, as the first and the last stretches of records only have one PBE each, they will not receive speed estimates.

4.3 Distance and Time Estimation

We next describe how we estimate the speed between

two BPEs. Specifically, we need to estimate the intra-cell boundary-to-boundary distance and the travel time. As the only available information for distance estimation is tower coordinates and tower visiting orders, for a segment with two PBEs $P_{i,j}$ and $P_{j,k}$, we use a straight line trajectory $l_i \rightarrow l_j \rightarrow l_k$ that passes all three tower l_i, l_j and l_k as an estimated trajectory. With the coordinates of towers, the euclidean distance between towers, $d(l_i, l_j)$ and $d(l_j, l_k)$, can be calculated. Since the boundaries are perpendicular bisectors of lines connecting towers (as we use Voronoi diagrams to represent cell coverage areas), the travel distance can be estimated by $\frac{d(l_i,l_j)+d(l_j,l_k)}{2}$. Note that if more information such as the underlying road networks are provided, the road trajectories that has the maximum likelihood to match visited tower sequences can also be used instead of the straight line trajectories.

The travel time of a segment is calculated by the time difference of two related PBEs. Since each PBE has a time interval associated with it for entering and leaving the overlapping area, we can calculate a range of possible values for travel time estimation, including both a tight bound and a relaxed bound. The former one suggests the shortest possible travel time to move through the area, while the latter one indicates the longest possible travel time. For example, for two PBEs $P_{i,j}$ and $P_{j,k}$ with a time interval of $({t_i}^{last}, {t_j}^{first})$ and $({t_j}^{last}, {t_k}^{first})$, respectively, we can easily derive the tight bound as $\Delta t_{tight} = {t_j}^{last} - {t_j}^{first}$ and the relaxed bound is $\Delta t_{loose} = {t_k}^{first} - {t_i}^{last}$.

4.4 Distance Lower Bounds

Algorithm 2: Distance lower bound estimation

Data: TC: a list of tower coordinates.
Result: D_{lb}: a list of all possible boundary-to-boundary distance lower bound estimated from Voronoi diagram.

```
//project to euclidean space with
  equirectangular projection
```

$TL \leftarrow TC$;
$P \leftarrow TL$;
build Voronoi diagram VD with P ;
for *each edge*1 *in* VD **do**
  $pset1 \leftarrow$ Voronoi points of *edge*1 ;
  for *each edge*2 *in* VD **do**
    $pset2 \leftarrow$ Voronoi points of *edge*2 ;
    if $pset1 \cap pset2 \neq \emptyset$ **then**
      $D_{lb}[(pset1, pset2)] \leftarrow |edge1 - edge2|$;
    end
  end
end
return R

In this section, we introduce the concept of distance lower bounds. This is motivated by the observation that it is usually hard to accurately estimate the true distances of users using the coverage areas of given towers for all possible trajectories and represent them with a single distance estimate. To see this, we show an example in Figure 6.

As shown in this figure, the area is divided into coverage areas of three towers A, B, and C. Solid lines represent real user trajectories while dashed lines represent the boundaries of towers. Observe that both user 1 and user 2 pass the three towers in the same order $A-B-C$. The real distance differences, however, are missing due to the limited location estimation accuracy of using tower locations. Therefore, in such cases, a single distance estimate will have to fail due to the wide variety of possible trajectories that can lead to the same tower visiting orders.

Figure 6: Common cases where a single distance estimate will fail.

Faced with this challenge, our next goal is to filter out distance estimates that are not likely to occur in real world scenarios, and provide the trajectory that is most likely as the solution. The major step here is to evaluate the confidence levels of different distance estimates based on estimated trajectories and tower locations, so that such confidence levels can be used as measures for evaluating differences in multiple trajectory lengths. Specifically, for two consecutive boundary events $P_{i,j}$ and $P_{j,k}$, the confidence level of a distance estimate d_{est} is defined as $C_{d_{est}} = \frac{d_{lb}}{d_{est}}$, where d_{lb} is the boundary-to-boundary distance lower bound, i.e., the minimum required distance to travel from the boundary of $P_{i,j}$ to the boundary of $P_{j,k}$, which serves as a conservative estimate for the shortest distance a user may travel. Intuitively, the longer an estimated distance is compared to this lower bound, the less likely it should be as it requires a more complex trajectory shape to be feasible.

We next describe how to calculate the distance lower bound. We first simplify the tower coverage model with the assumption that cell phones only communicate with the nearest tower, and towers have same transmission power. Then, based on the Voronoi diagram formed by towers' locations, we calculate the Voronoi cell shapes with their vertex locations. Figure 7 shows an example of the Voronoi diagram construction with five towers. Each region in the Voronoi diagram represents the coverage area of one tower, while the edges in Voronoi diagrams are central focus lines of overlapping coverage area of towers (not illustrated with Voroni diagrams, but exist in real-world tower communications). The shortest travel distance between boundaries is therefore transformed into the shortest distance between two Voronoi edges, and can be solved using simple geometric methods.

4.5 Virtual Boundaries

Figure 7: Voronoi diagram to represent communication coverage of each tower.

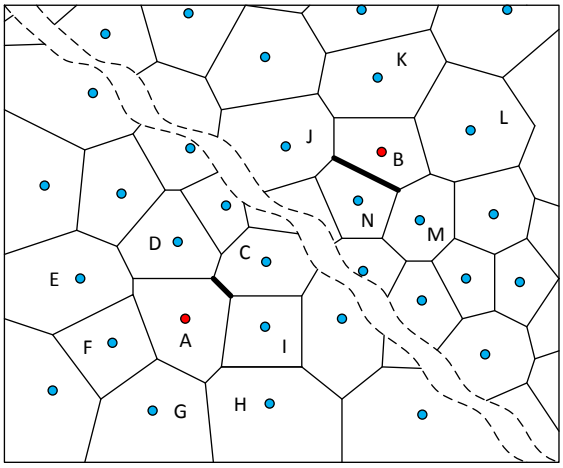


Figure 8: Dealing with virtual boundaries.

A fundamental limitation of using cellphone-tower communication datasets is that records are only collected when mobile data accesses are happening. If the user is keeping silent, there is no way for us to know their locations. In such cases, if the user has traveled across multiple boundaries, we may encounter the following observation: we analyze the consecutive records for this user, and find out that their l_i and l_j may be far away from each other and do not necessarily share a common boundary area. If l_i and l_j are adjacent to each other, we say to $P_{i,j}$ has a real boundary. Otherwise, we refer to it as a virtual boundary.

Different from real boundaries that are treated as an edge in the Voronoi diagram, virtual boundaries are actually distance estimates themselves as users have passed the coverage area of several towers during a PBE within a virtual boundary. Since we do not have any information regarding which towers the user has visited in between, to calculate the distance lower bound of a virtual boundary, we instead use the shortest distance of all possible boundary pairs of l_i and l_j as the best estimate.

We now give an example in Figure 8, where we analyze two consecutive records: r_i for tower A and r_j for tower B. Since tower A and tower B do not share physical boundaries, they only have a virtual boundary between them. To calculate their shortest distance, we calculate the distance for each boundary of tower A to each boundary of tower B, and use the shortest one of all boundary pairs as the estimated distance. In this example, the distance between boundary (A, C) and boundary (B, N) is used as the distance lower bound of the virtual boundary (A, B).

Returning to our earlier analysis, for segments that have PBEs with virtual boundaries, we merge them with adjacent segments if they exist. The distance estimate and lower bound of a segment are the sum of distance estimates and distance lower bounds of both records, and if any, the virtual boundaries between them. Note that as we calculate the sum of distance lower bounds, the resulting distance lower bound is still the minimum distance required to reach one real boundary from the other, even this requires that the trajectory should pass through virtual boundaries between consecutively visited towers.

4.6 Speed Estimation

Now that we have a distance estimate d_{est}, a distance lower bound d_{lb}, and a range of possible travel time $(\Delta t_{tight}, \Delta t_{loose})$ for each segment, we can infer the travel time of a segment estimated by $\Delta t_{est} = \frac{\Delta t_{tight} + \Delta t_{loose}}{2}$. We denote this by Δt_{est}. We next calculate the confidence levels for both distance estimates and travel time estimates as follows:

$$C_{d_{est}} = \frac{d_{lb}}{d_{est}} \quad (1)$$

$$C_{\Delta t_{est}} = \frac{\Delta t_{est}}{\Delta t_{loose}} \quad (2)$$

By setting a threshold for both confidence levels, we can filter out estimates that are not accurate enough. Although we can filter out more inaccurate speed estimates with a much stricter threshold in both confidence levels, we may end up with a limited number of records that have qualified speed estimates. Finally, after setting proper threshold for confidence levels, the speed of the user can be estimated as the following:

$$s_{est} = \frac{d_{est}}{\Delta t_{est}} \quad (3)$$

The detail of the speed estimation algorithm is shown in Algorithm 3.

4.7 Cell Oscillation and Speed Compensation

The distance lower bounds can also help to eliminate the cell oscillation problem, i.e., when a user near boundary area randomly communicates with two or more towers in short periods, generating a sequence of false pass-boundary events. Since the user keeps passing the same boundary, the distance lower bound for such scenarios should always be 0. Therefore, the confidence level of distance estimates will also be 0, which means that we can detect them and filter them out.

Since segments between these false BPEs usually have very short durations due to the nature of how they are generated, we estimate the speed for such segments based on

the assumption that a user's speed does not change dramatically in a very short time period. Therefore, for a segment between false BPEs, if there is a segment that happens to be very close to it and has a qualified speed estimate, we will use its speed estimate as the speed estimate for the segment with false BPEs. Other kinds of low confident level speed estimations can also be compensated by the nearby segments with high confidence levels as long as the confidence level and time period are properly handled.

Algorithm 3: Speed estimation

Data: R: mobile data records arranged by segments r;
r: A mobile data segment which has:
$l_{pre}, l, lpost$: location estimates for segment before r, itself and segment after r, where l_{pre} and l_{post} are acquired during BPE extraction;
d_{est}:distance estimates; (Δt_{tight}, Δt_{loose}): travel time estimates; D_{lb}: a list of boundary-to-boundary distance lower bound estimated from Voronoi diagram.
Param: T_{C_d}, T_{C_t}: confidence level threshold for distance estimates and travel time estimates.
Result: S: Speed estimates for each segment.

for *each* r *in* R **do**
  `//Check if r has real boundary and find its distance lower bound`
  if $((l_{pre}, l), (l, l_{post}))$ *is not in* D_{lb} **then**
    combine r with next record ;
    continue ;
  else
    $d_{lb} \leftarrow D_{lb}[(l_{pre}, l), (l, l_{post})]$;
  end
  `//Calculate travel time estimates`
  $\Delta t_{est} \leftarrow \frac{\Delta t_{tight} + \Delta t_{loose}}{2}$;
  `//Calculate confidence level`
  $Cd_{est} \leftarrow \frac{d_{lb}}{d_{est}}$;
  $C_{\Delta t_{est}} \leftarrow \frac{\Delta t_{est}}{\Delta t_{loose}}$;
  `//Estimate speed if meet threshold`
  if $Cd_{est} \geq T_{C_d}$ *and* $C_{\Delta t_{est}} \geq T_{C_t}$ **then**
    $s_{est} \leftarrow \frac{d_{est}}{\Delta t_{est}}$;
    $S[r] \leftarrow s_{est}$;
  end
end
return S

5. EXPERIMENTAL RESULTS

With our methodology on speed estimates, we next explain our findings on correlations between user mobility and mobile data access patterns in this section. We start with the correlation of the speed and the average mobile data access volumes. Then we reveal the relation of speed and average time intervals between consecutive mobile data accesses. Finally, we illustrate the correlation between speed and the types of app usage that are responsible for generating the corresponding mobile data traffic.

5.1 Experiment Settings

To estimate the speed, our algorithm requires a user has visited at least 3 towers consecutively. In the dataset, we find around 13 million records out of 58 million records can be utilized. In our experiments, to balance the accuracy of speed estimates and the number of mobile data access records that have qualified speed estimates, we set the threshold of both distance ratio d_{ratio} and duration ratio Δt_{ratio} empirically as 0.6. After the filtering, we have around 1 million records out of total 13 million records that meet both criteria. Figure 9(a) shows the cumulative density function (CDF) of both raw speed estimates without filtering and filtered speed estimates. As we can see that the filtered speed estimates are more realistic compared to raw speed estimates. Most of the false high speed estimates and low speed estimates are filtered out by setting thresholds of confidence levels for distance estimates and travel time estimates.

In the following experiments, we only show results in the speed range from 0 km/h to 100 km/h, since there are very few records with a speed estimate above 100 km/h for any meaningful insights.

5.2 Speed and Data Volumes

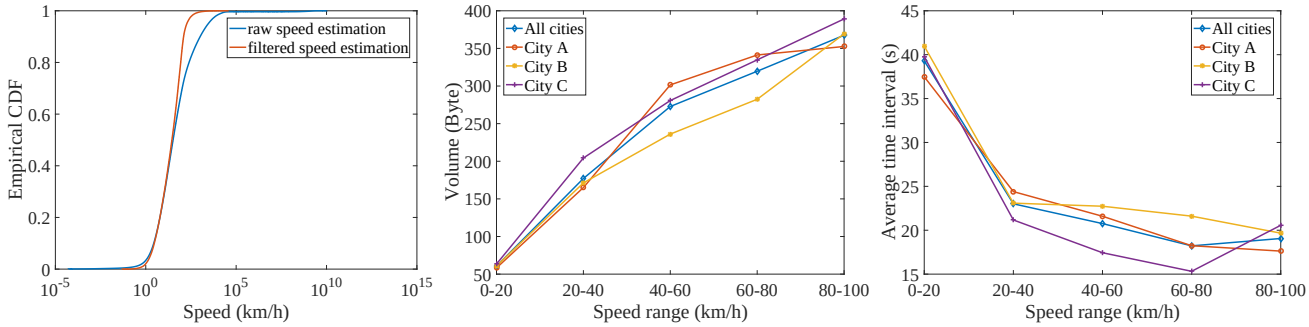
Figure 9(b) shows the results of the correlation of user speed and the average mobile data access volumes per user per second. We demonstrate the data from all three cities combined and each city respectively. The figure shows a clear trend that users are more active in accessing mobile data as the speed increases and the trend holds true for all three cities. In fact, a user with speed estimates of 80-100 km/h could reach an average data volume of 6 times of a low-speed user. Similarly, this trend also holds true for all the cities. Note that these results only show an increase in the mobile data access volume as user speed increases. It does not suggest lower speed users access online contents less frequently. Actually, we believe one reason might be that a large portion of a low-speed user's online needs is already fulfilled by various kind of high-speed connections such as Wifi hotspots. To this end, we reach similar findings with previous work [33] on correlation of user mobility and mobile data access volume, except that the previous work used the number of towers visited by a user as the indicator of user mobility.

5.3 Speed and Access Frequency

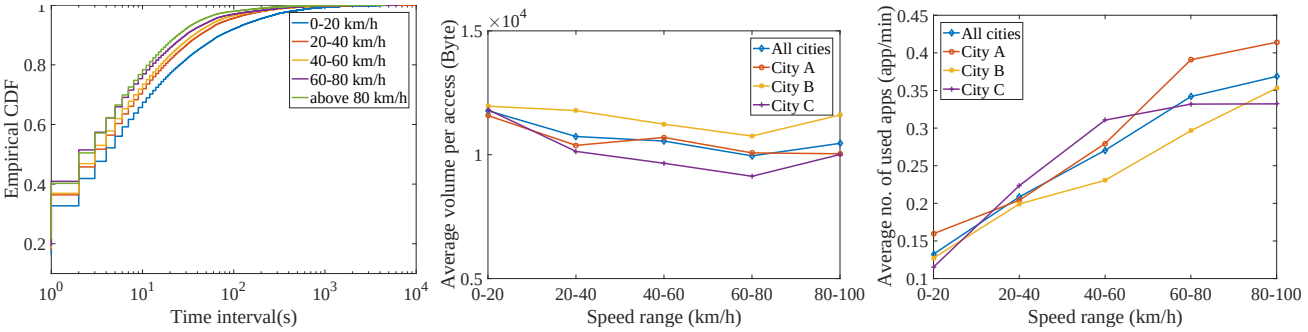
Figure 9(c) shows the correlation of speed and average time intervals between consecutive mobile data access records. The CDF of data time intervals for various speed ranges of all three cities are also shown in Figure 9(d). Note that since the time precision of our data trace is seconds, so there are steps in Figure 9(d). The decrease in time intervals as speed increases suggests that high-speed user accesses mobile data more frequently than low-speed users. A user with a speed estimate of 80-100 km/h access mobile data almost 2 times more frequently than a user with a speed estimate of 0-20 km/h on average. The trend holds for all three cities except that there is an odd point at 80-100 km/h for one city, which may be caused by the lacking of available data.

We show the average volume for each data access in Figure 9(e). As the user speed increases, there is no apparent correlation with average volume for each data access. This suggests that the the increasing in the average volume which is shown in Figure 9(b) is mainly cause by the increased data access frequency, not the volume for each data access.

5.4 Speed and App Choice



(a) Speed Estimates Empirical CDF (b) Data Volume (c) Average Time Interval



(d) Time Interval Empirical CDF (e) Avearge Volume per Data Access (f) Number of Apps Used

Figure 9: The (a) Speed estimates; and its correlation with (b) data volume, (c) average time interval between consecutive data access, (d) time interval between consecutive connections, (e) average data access volume for each data access and (f) average number of apps used.

Figure 9(f) shows the correlation between user speed and average number of apps being used for each user during each data segment per minute. The trend clearly shows that as the speed goes up, the app usage diversity increase rapidly. A user with a speed estimate of 80-100 km/h could reach an average used apps of 8 times for a low-speed user. This trend holds true for all the cities.

We further investigated the trend of the contribution of various app categories on the total mobile data access as the user speed increases. The contribution was defined as the mobile data access of one category versus all categories. According to the mobile service provider, each app in our dataset was assigned to one of 19 categories. Focusing on apps that contributed the most to the total mobile data access volume, we selected the top 8 app categories, as shown in Table 1. The correlation between the user speed and the contribution of each category is shown in Figure 10.

Among the top 8 categories, Microblog, Navigation and Music show a clear upward trend as the speed increases. The impact of navigation has the most steady increase due to the increased needs for such apps when driving. The impact almost doubles for users with speed estimates of 80-100 km/h compared to users with speed estimates of 0-20 km/h. Instant message, Video and App market show a downward trend as the speed increases. The reason could be the users are cost sensitive and strictly control the data usage for large app downloading and video streaming. Browser & Downloading and Reading show a quite stable impact that does not change a lot as the speed increases.

App Category	# Apps	Volume (GB)
Instant Messages	30	97.3
Reading	101	17.6
Microblog	43	13.0
Navigation	38	10.8
Video	63	45.2
Music	33	27.4
App Market	45	37.0
Browser & Download	558	353.5
Others	464	118.9

Table 1: App categories

6. CONCLUSIONS

In this paper, we studied the correlation between user mobility and app usage patterns, and in particular, we focus on users' moving speed as the key mobility metric. We find that there are trends between user speed and the app usage patterns for different categories of apps. In our methodology, we overcome several challenges, including how to estimate speeds accurately with high confidence and reliability. Based on the speed estimations, we are able to reveal the correlation of user mobility with mobile data usage patterns including the data volume, the data access frequency, and the traffic share of apps on the total mobile data traffic. Results showed that with users that have high speed estimation tend to user smartphone more frequently and generate more traffic on the mobile data network. Furthermore, the user speed also plays an important role in the contribution of each smartphone app categories on the total mobile data

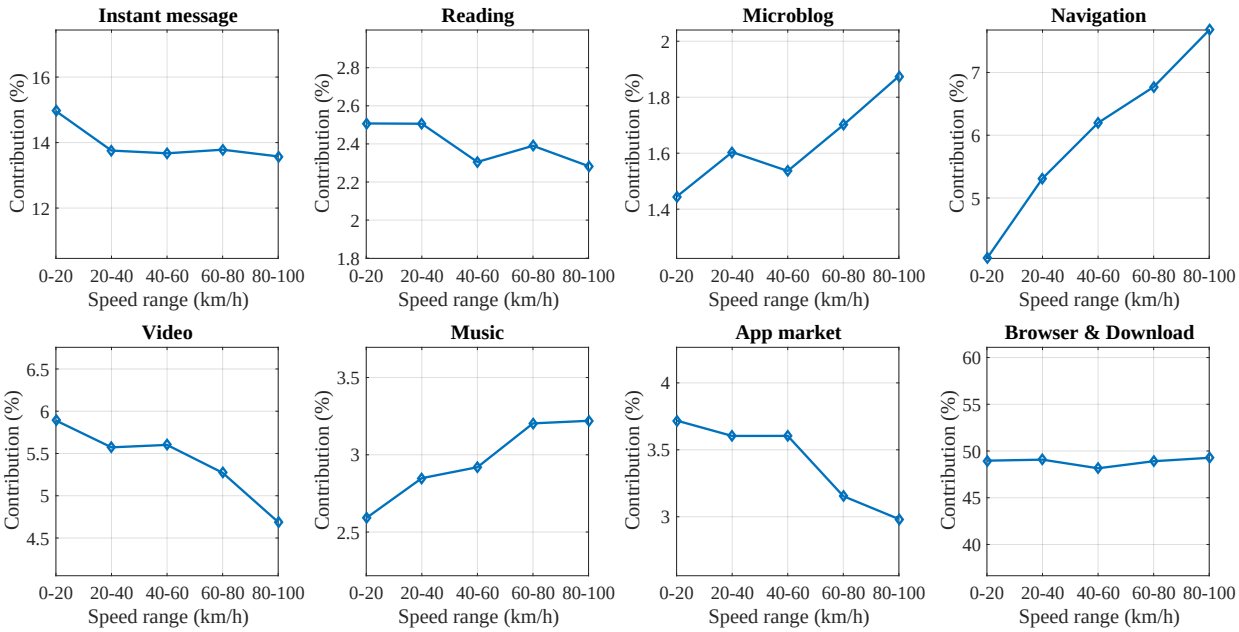


Figure 10: Correlation of user speed and contribution of app categories.

traffic.

7. REFERENCES

[1] S. Bekhor and I. B. Shem-Tov. Investigation of travel patterns using passive cellular phone data. *Journal of Location Based Services*, pages 1–20, 2015.

[2] F. Biljecki, H. Ledoux, and P. Van Oosterom. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2):385–407, 2013.

[3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[4] J. Doyle, P. Hung, D. Kelly, S. McLoone, and R. Farrell. Utilising mobile phone billing records for travel mode discovery. 2011.

[5] Ericsson. Ericsson mobility report, Feb. 2016. http://www.ericsson.com/res/docs/2016/mobility-report/ericsson-mobility-report-feb-2016-interim.pdf.

[6] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[7] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 13:1–13:14, New York, NY, USA, 2013. ACM.

[8] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks*, 64:296–307, 2014.

[9] S. Jiang, G. A. Fiore, Y. Yang, J. Ferreira Jr, E. Frazzoli, and M. C. González. A review of urban computing for mobile phone traces: current methods, challenges and opportunities. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, page 2. ACM, 2013.

[10] V. Manzoni, D. Maniloff, K. Kloeckl, and C. Ratti. Transportation mode identification and real-time co2 emission estimation using smartphones. *SENSEable City Lab, Massachusetts Institute of Technology, nd*, 2010.

[11] L. Meng, S. Liu, and A. D. Striegel. Analyzing the impact of proximity, location, and personality on smartphone usage. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 293–298. IEEE, 2014.

[12] M. Y. Mosny. *Path Estimation Using Cellular Handover*. PhD thesis, Princeton University, 2006.

[13] Nielsen. Smartphones: So many apps, so much time, July 2014. http://www.nielsen.com/us/en/insights/news/2014/smartphones-so-many-apps--so-much-time.html.

[14] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 570–573, 2011.

[15] H. Ohashi, T. Akiyama, M. Yamamoto, and A. Sato. Automatic trip-separation method using sensor data continuously collected by smartphone. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2984–2990, Oct 2014.

[16] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, Mar. 2010.

[17] G. Rose. Mobile phones as traffic probes: practices, prospects and issues. *Transport Reviews*, 26(3):275–291, 2006.

[18] J. Ryder, B. Longstaff, S. Reddy, and D. Estrin. Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 927–931, Aug 2009.

[19] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang. Characterizing geospatial dynamics of application usage in a 3g cellular data network. In *INFOCOM, 2012 Proceedings IEEE*, pages 1341–1349. IEEE, 2012.

[20] D. Shin, D. Aliaga, B. Tunçer, S. M. Arisona, S. Kim, D. Zünd, and G. Schmitt. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems*, 53:76–86, 2015.

[21] Z. Smoreda, A.-M. Olteanu-Raimond, T. Couronné, et al. Spatiotemporal data from mobile phones for personal mobility assessment. *Transport survey methods: best practice for decision making*, 41:745–767, 2013.

[22] Statista. Number of apps available in leading app stores as of july 2015, 2016. http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

[23] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 54–63. ACM, 2011.

[24] C. Tacconi, S. Mellone, and L. Chiari. Smartphone-based applications for investigating falls and mobility. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 258–261, May 2011.

[25] K. Waga, A. Tabarcea, M. Chen, and P. Franti. Detecting movement type by route segmentation and classification. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pages 508–513, Oct 2012.

[26] H. Wang, F. Calabrese, G. Di Lorenzo, and C. Ratti. Transportation mode inference from anonymized and aggregated mobile phone call detail records. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 318–323. IEEE, 2010.

[27] S. Wang, C. Chen, and J. Ma. Accelerometer based transportation mode recognition on mobile phones. In *2010 Asia-Pacific Conference on Wearable Computing Systems*, pages 44–46. IEEE, 2010.

[28] T. Wang, C. Chen, and J. Ma. Mobile phone data as an alternative data source for travel behavior studies. In *Transportation Research Board 93rd Annual Meeting*, number 14-2887, 2014.

[29] P. Widhalm, P. Nitsche, and N. BrÃd'ndie. Transport mode detection with realistic smartphone sensor data. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 573–576, Nov 2012.

[30] P. Widhalm, Y. Yang, M. Ulm, S. Athavale, and M. C. González. Discovering urban activity patterns in cell phone data. *Transportation*, 42(4):597–623, 2015.

[31] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 329–344. ACM, 2011.

[32] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu. Fast app launching for mobile devices using predictive user context. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 113–126. ACM, 2012.

[33] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng. Characterizing user behavior in mobile internet. *Emerging Topics in Computing, IEEE Transactions on*, 3(1):95–106, 2015.

[34] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. Understanding transportation modes based on gps data for web applications. *ACM Transactions on the Web (TWEB)*, 4(1):1, 2010.