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

ZHENG LU, University of Tennessee
YUNHE FENG, University of Tennessee
WENJUN ZHOU, University of Tennessee
XIAOLIN LI, Nanjing University
QING CAO, University of Tennessee

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 request 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.

ACM Reference format:
Zheng Lu, Yunhe Feng, Wenjun Zhou, Xiaolin Li, and Qing Cao. 2016. Inferring Correlation between User Mobility and App Usage in Massive Coarse-grained Data Traces. 1, 1, Article 1 (January 2016), 22 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

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 exabytes 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 [28]. 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 [18].

Consequently, recent research has invested considerable effort to understand smartphone app usage behavior, as such understandings can help app developers and mobile advertisers tremendously [37, 39]. In the previous work, both temporal patterns (e.g., individual app usage histories) and spatial patterns (e.g., location contexts) have been extensively studied [16]. Their results have enabled novel applications, such as smartphone app launching prediction services [38] and location-aware event recommendations.

DOI: 10.1145/nnnnnnn.nnnnnnn

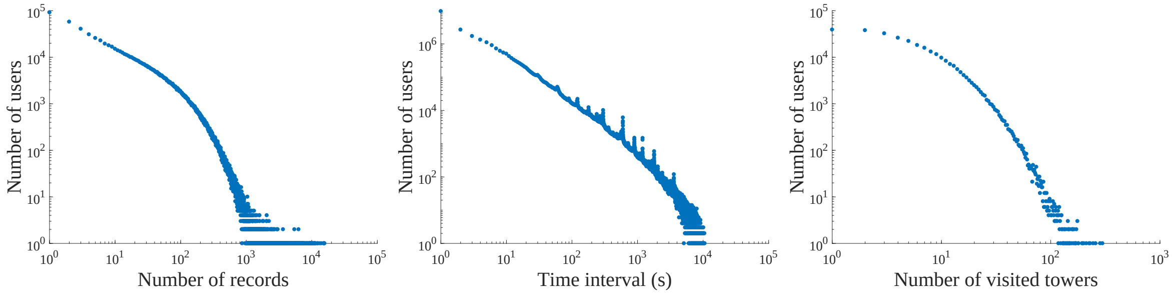
As the cellular datasets become more and more available, itfis vital to design processing algorithms to extract fine-grained user mobility traces from these datasets. In this paper, we focus on user mobility, and investigate how this feature correlates with the usage patterns of smartphone app users. 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. **possible other applications:**

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 [3, 20, 21, 24, 29, 31, 35, 41]. Later work suggested that it may be possible to use cell tower communications to monitor users' mobility indirectly [23], where efforts have been focusing on inferring users' trajectories [10, 17] or transportation mode [2, 32] 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 latter line of research of using 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 the population of certain 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 [9, 10, 17, 27, 36] or finding mobility motifs [6, 34], but we could not find one that suits our needs as we find their results are clearly still too coarse-grained. One reason is the dataset difference: their data mostly are sparse compared to ours. For example, one dataset contains users who perform daily commute or city to city long distance trips. In contrast, 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 a 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 various groups of users with different moving speeds, a task that is considerably more challenging than simply



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

Fig. 1. Dataset characteristics.

performing correlation analysis between all apps and users without differentiation. Therefore, our methods need to be customized for the needs of 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 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. **We briefly evaluate our speed estimation algorithm in Section 4.7.** Section 6 explains our findings on the correlation of user speeds and mobile data access patterns. **We state the limitations of our work in Section 4.7.** Finally, we conclude our work in Section 8.

2 RELATED WORK

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

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. [39] studied the mobile user behavior by focusing on data usage, mobility pattern and application usage. In [37], 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.

Using GPS [3, 20, 21, 24, 29, 31, 35, 41] and embedded sensors [8, 15, 20, 21, 26, 30, 33], 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 [3, 20, 31, 41], feature selection [3, 29, 33, 41]. 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 work [23], 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 [2, 9, 10, 14, 17, 27, 36] or mobility motifs [6, 34]. For example, [10, 17] 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 [32], it aims to classify a user's transportation mode by clustering travel time distribution. Finally, researchers [2] 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 the calability of cell-phone data traces and achieves fine-grained user mobility inference on top of it.

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, a work [4] 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 [19] studied a similar problem with a different dataset. [25, 39] 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. [16] studied how the proximity, the location and individual differences (e.g., personality) can effect the user's mobile data usage. Finally, [40] showed the apps access pattern under various user mobility properties such as number of visited cell phone towers and radius of gyration. However, analysis of much complex user mobility such as user speed is still a missing piece in these works.

(M5) Although recent studies [1, 13] shows that WiFi handles half of the mobile data traffic, cell phone network have much better coverage and user mobility diversity. It is more meaningful to study user mobility under cellular traffic.

3 DATA DESCRIPTION

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

3.1 Dataset

(M1) Our dataset contains mobile data access history of all active users (during a sunday evening from 6pm to 9pm) of a major mobile carrier in three cities of China in September 2014. The data was collected by the carrier at the cell tower side. For each user, all data request records during the study period are available, where each record consists of the user ID (a hashed value for anonymity), the tower ID (from which we were able to look up its geo-coordinates), the timestamp, the app identifier, and other data access features such as data volumes.

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 around 900 thousand. The total active time of all users accumulates to more than 1 million hours. Figure 2 shows a

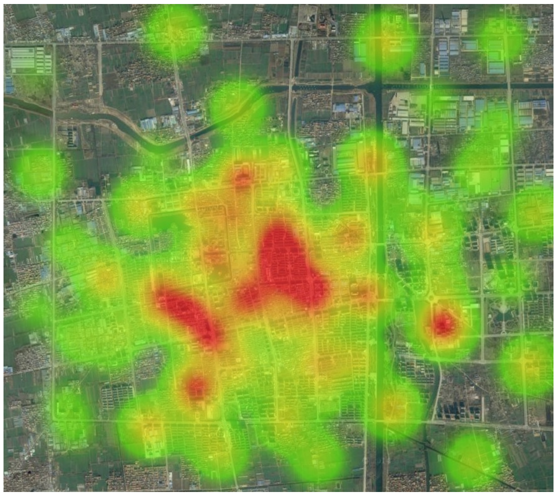


Fig. 2. Communication density in a city area

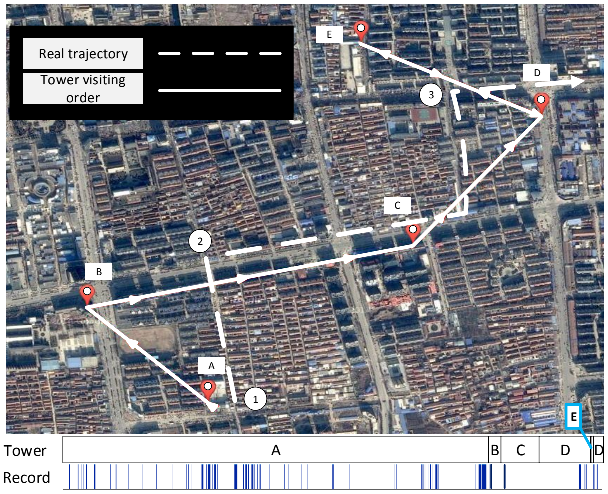


Fig. 3. Example data access activities of a user

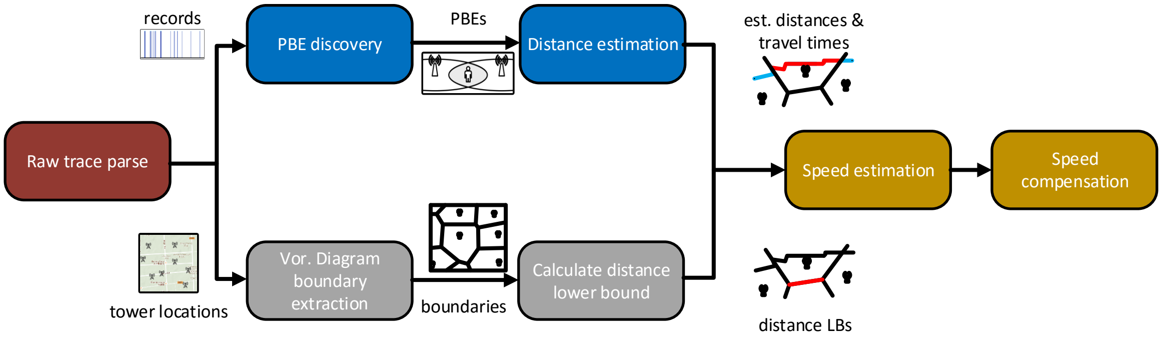


Fig. 4. Speed estimation system overview.

heatmap of the mobile data access in a city area of our dataset. The dataset contains both user initiated network access and background network access.

3.2 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 of 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 moving 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.

3.3 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, she started by walking from location 1, to location 2 where she waited for the bus. After a few minutes, she got onto the bus, which took the path towards location 3. Even though we did not know the actual path of the user, her locations could be inferred by the nearby towers to which her communication data were sent to.

Since the cell tower locations are all known, we can display the tower locations on 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 Figure 3 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, 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 she 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.

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 (PBE) 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 diagrams 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 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 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, which we call "pass-boundary events" (PBE).

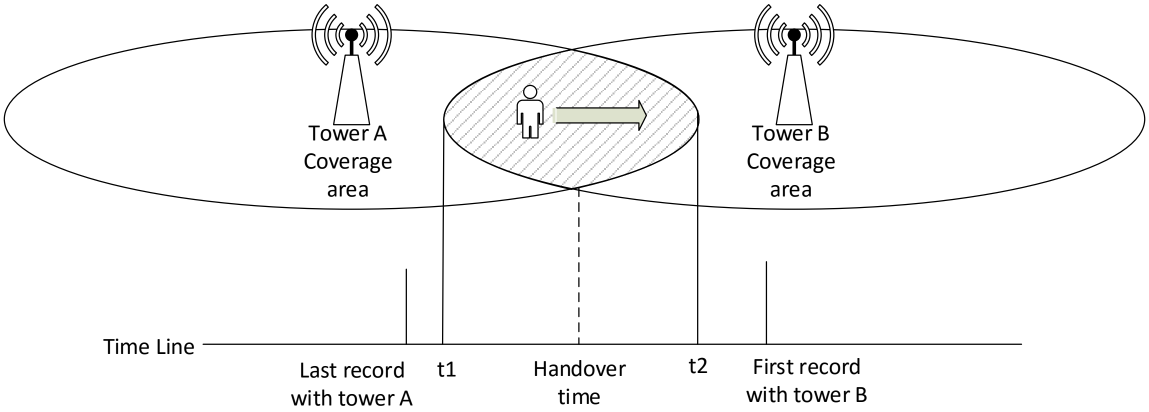


Fig. 5. A pass-boundary event.

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 PBEs 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 ${t_j}^{first} - {t_i}^{last}$. Once PBEs 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 PBEs 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, however, that as the first and the last stretches of records only have one PBE each, they will not have speed estimates.

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. We use l_c to represent location of current entry.

Result: R: segments r arranged by user and time. Its location l_r is defined by location of two adjacent PBE. Its time t_r is defined by the time of first data entry and time of last data entry in this segment.

```
for each $u$ in $Trace$ do
    $l_c \leftarrow \emptyset, r \leftarrow \emptyset, e_{lastend} \leftarrow \emptyset, e_{start} \leftarrow \emptyset, e_{end} \leftarrow \emptyset$ ;
    for each $e$ in $Trace[u]$ do
        if $l_c \neq \emptyset$ and $l_c \neq l_e$ then
            $l_r \leftarrow (l_{e_{lastend}}, l_c, l_e), t_r \leftarrow (t_{e_{start}}, t_{e_{end}})$ ;
            append $r$ to $R[u]$ ;
        end
        if $l_c = \emptyset$ or $l_c \neq l_e$ then
            $e_{lastend} \leftarrow e_{end}, l_c \leftarrow l_e, e_{start} \leftarrow e$
        end
        $e_{end} \leftarrow e$ ;
    end
    $l_r \leftarrow (l_{e_{lastend}}, l_c, l_e), t_r \leftarrow (t_{e_{start}}, t_{e_{end}})$ ;
    append $r$ to $R[u]$ ;
end
return $R$
```

4.3 Distance and Time Estimation

We next describe how we estimate the speed between two PBEs. 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 $l_i \longrightarrow l_j \longrightarrow l_k$ 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

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.

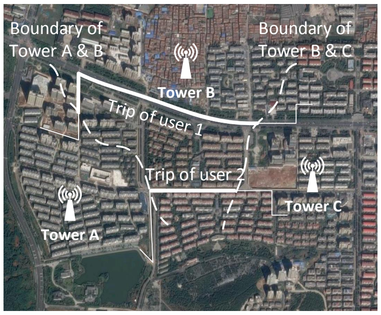


Fig. 6. Common cases where a single distance estimate would fail

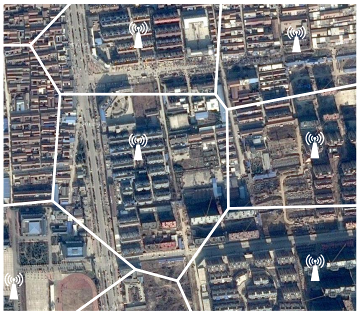


Fig. 7. Voronoi diagram to represent communication coverage of each tower

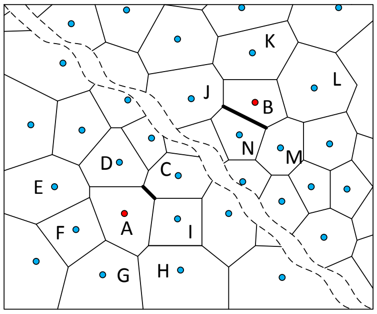


Fig. 8. Dealing with virtual boundaries

As shown in Figure 6, 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.

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.

In order to calculate the distance lower bound, we first simplify the tower coverage model with the Voronoi diagram. 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 (such areas are hidden in simple Voroni diagrams, but they widely 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. The detailed algorithm is shown in Algorithm 2.

4.5 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.

ALGORITHM 2: Distance lower bound estimation

```
Data: $TC$: a list of tower coordinates.
Result: $D_{lb}$: a list of all boundary-to-boundary distance lower bounds estimated from Voronoi diagram.
$TL \leftarrow TC$; $P \leftarrow TL$ ;
Build Voronoi diagram $VD$ with $P$ ;
for each $edge1$ in $VD$ do
    $pset1 \leftarrow$ Voronoi points of $edge1$ ;
    for each $edge2$ in $VD$ do
        $pset2 \leftarrow$ Voronoi points of $edge2$ ;
        if $pset1 \cap pset2 \neq \emptyset$ then
            $D_{lb}[(pset1, pset2)] \leftarrow |edge1 - edge2|$;
        end
    end
end
return $R$
```

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 from 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 represented as $(\Delta t_{tight}, \Delta t_{loose})$ 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:

$$C_{d_{est}} = \frac{d_{lb}}{d_{est}} \tag{1}$$

$$C_{\Delta t_{est}} = \frac{\Delta t_{est}}{\Delta t_{loose}} \tag{2}$$

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:

$$s_{est} = \frac{d_{est}}{\Delta t_{est}} \tag{3}$$

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

ALGORITHM 3: Speed estimation

```
Data: $R$ from Algorithm 1 and $D_{lb}$ from Algorithm 2.
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 $S$
```

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 PBEs 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 PBEs, 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 PBEs. 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.

5 (M3) SHORT EVALUATION

Before we preceed to our findings on correlations between user mobility and mobile data access patterns, we perform a short evaluation of our speed estimation algorithm in this section. Since the ground truth of user mobility of our dataset is not available to us, we use a similar but much smaller dataset [12] collected by Intel Placelab to perform a the evaluation of our speed estimation algorithm. This small dataset contains both cell phone data and GPS traces. We use the GPS as ground truth for the evaluation.

5.1 Dataset and Experiment Settings

The dataset we used was collected in Seattle area in September 2004. Both cell phone data and GPS data were collected with a Nokia 6600 cell phone and a separate GPS unit. Similar to our dataset, the cell phone data contains only the ID of the cell tower. Note that as the dataset is collected more than ten years ago, so we can only find coordinations for part of the tower IDs contains in the dataset. We thus discarded part of the datasets that contains tower IDs with unknown coordinates. The data we used in our evaluation contains several trips with a total length of 40 minutes and 2007 records.

We use the same parameter setting as latter we use in our dataset analysis. Thresholds of both distance ratio d_{ratio} and duration ratio Δt_{ratio} are ste as 0.6. We are able to get speed estimation for 1955 records out of the total 2007 records. After the filtering, we have 571 that meet both criteria. We also disable both distance lower bound filter and travel time filter by setting thresholds to 0 and use the raw speed estimates as our baseline for comparison. Note that, as we previously stated, if more information such as the underlying road networks are provided, our distance and travel time filters can also be applied to the road trajectories that has the maximum likelihood to match visited tower sequences instead of the straight line trajectories.

5.2 Speed estimation accuracy

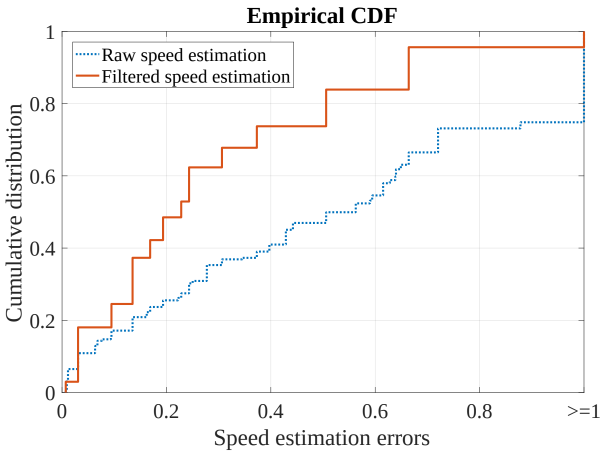


Fig. 9. The CDF of raw speed estimation and filtered speed estimation

Figure 9 shows the cdf of errors for both filtered speed estimates and raw speed estimates. Here we define the speed estimation errors as $e = |(s_{est} - s_{gt})|/s_{gt}$, where s_{est} is speed estimated by cell

phone data and s_{gt} is ground truth speed calculated by GPS data. By using the absolute value, e is always a positive value. As the value of e can reach to as high as a few hundreds due to cell oscillation problem, we set the limit of e to 1 in Figure 9 to better show the difference of speed estimation errors in reasonable error range.

As we can see in Figure 9, filtered speed estimation significantly outperforms raw speed estimation in terms of accuracy. About 50% of the filtered speed estimates have less than 0.2 error rate while only 25% of raw speed estimates achieve the same level of accuracy. More than 80% of filtered speed estimates have error rates less or equal to 0.5, while for raw speed estimates this number is only less than 50%. For the high error rate part, there is only less than 5% of filtered speed estimates have higher than 1 error rates compared to more than 15% of raw speed estimates have such high error rates.

5.3 Speed filtering

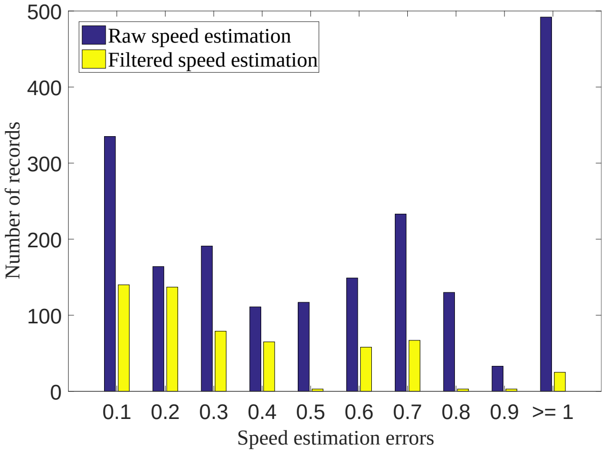


Fig. 10. Number of records in each error bracket for raw speed estimation and filtered speed estimation

Figure 10 shows the number of records in each speed estimation error bracket. For example, there are around 350 data records have raw speed estimates with an error rate of 0.1. Among these 350 data records, only around 150 data records have filtered speed estimates. The rest 200 data records are filtered our by our speed estimation algorithm. Is worth noting that our speed estimation algorithm does not improve speed estimation accuracy on a single data record. Actually, it filtered out those speed estimates that have possible high error rates by setting up threshold for both distance and travel time. In Figure 10, we can see that approximately 50% of records that have raw speed estimation errors less than 0.5 have been filtered out while more than 85% of records that have raw speed estimation errors greater than 0.5 have been filtered out.

6 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.

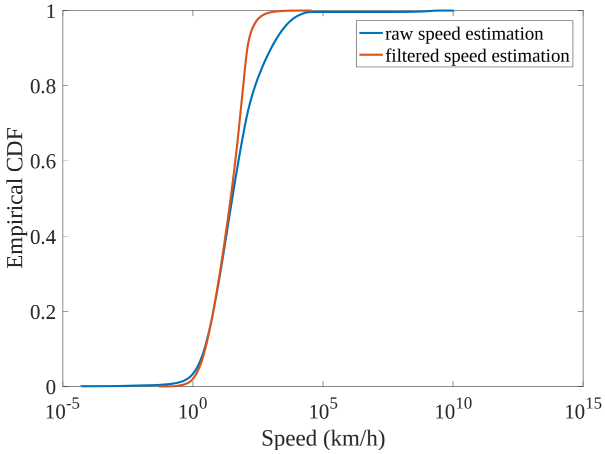


Fig. 11. Empirical CDF of speed estimates.

6.1 Experiment Settings

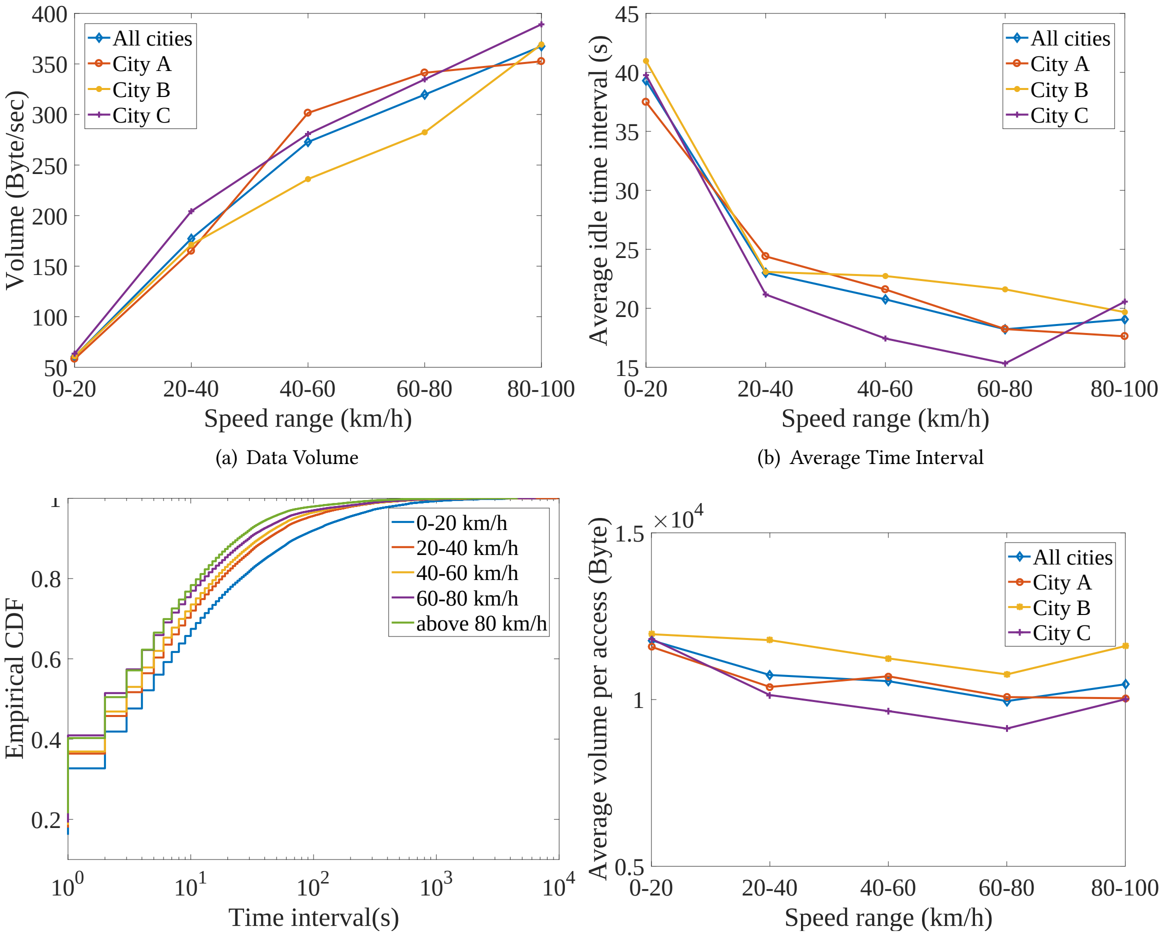
To estimate the speed, our algorithm requires a user has visited at least 3 towers consecutively. In the dataset, we find that around 13 million records out of 58 million records can be utilized. Although the dataset contains both user initiated network access and background network access, we find it very hard to separate them reliably. 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 11 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.

(R1-4) We implement our speed estimation algorithm and mobile data usage pattern analysis algorithm in Python. We use Voronoi package from Scipy [11] to construct Voronoi maps with tower coordinates and Shapely [7] for geometry calculations. All analysis are carried out on a single Cloudlab [22] c8220 server with two 10-core 2.2GHz E5-2660 processors and 256GB memory.

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.

6.2 Speed and Data Volumes

Figure 12(a) 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



(a) Data Volume

(b) Average Time Interval

(c) Time Interval Empirical CDF

(d) Average Volume per Data Access

Fig. 12. The correlation of speed estimates with (a) data volume, (b) average idle time interval between consecutive data access, (c) idle time interval between consecutive connections, (d) average data access volume for each data access.

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 [39] on the 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.

6.3 Speed and Access Frequency

Figure 12(b) shows the correlation of speed and average idle 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 12(c). Note that since the time precision of our data trace is seconds, so there are steps in Figure 12(c). 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 twice 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 12(d). As the user speed increases, there is no apparent correlation with average volume for each data access. This suggests that increasing in the average volume which is shown in Figure 12(a) is mainly cause by the increased data access frequency, not the volume for each data access.

6.4 Speed and App Usage

According to the mobile service provider, each app in our dataset was assigned to one of 18 categories, as shown in Table 1.

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
Game	106	9.2
Payment	18	1.2
Comic	12	0.8
Email	10	1.5
P2P	8	3.9
VOIP	17	0.3
Multimedia Messages	2	0.3
Browsing	558	353.5
Finance	25	0.7
Security	22	5.2
Others	244	95.8

Table 1. App categories

We first divided the estimated user speed by 50km/h into two classes. The low speed class includes transportation modes such as walking, cycling, bus and other low speed vehicles. The high speed class includes mainly high speed vehicles. We showed the number of data access per app category in Figure 13 and data volume per app category in Figure 14. In both figures, low speed users have more data access due to large user base. The share for each category holds similar trend for both high and low speed users. We will discuss more details on the contribution of each category to total data access in Figure 18.

Figure 15 shows the correlation between user speed and the average number of unique apps and app categories 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 increases rapidly. A user with a speed estimate of 80-100 km/h could use as many as 2 times apps per unit of time compared to a low-speed user. An explanation might be that for users with high

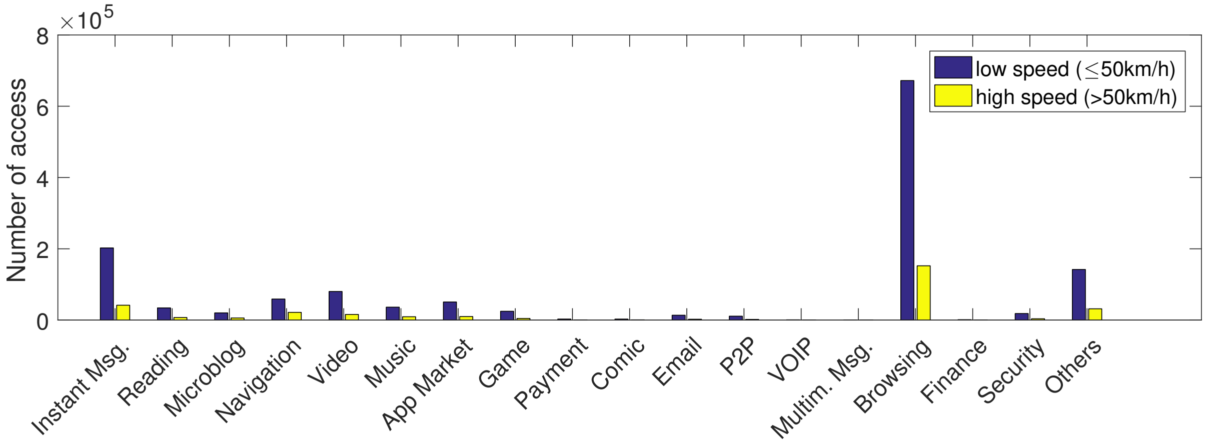


Fig. 13. The distribution of the number of access per app category with high and low speed users

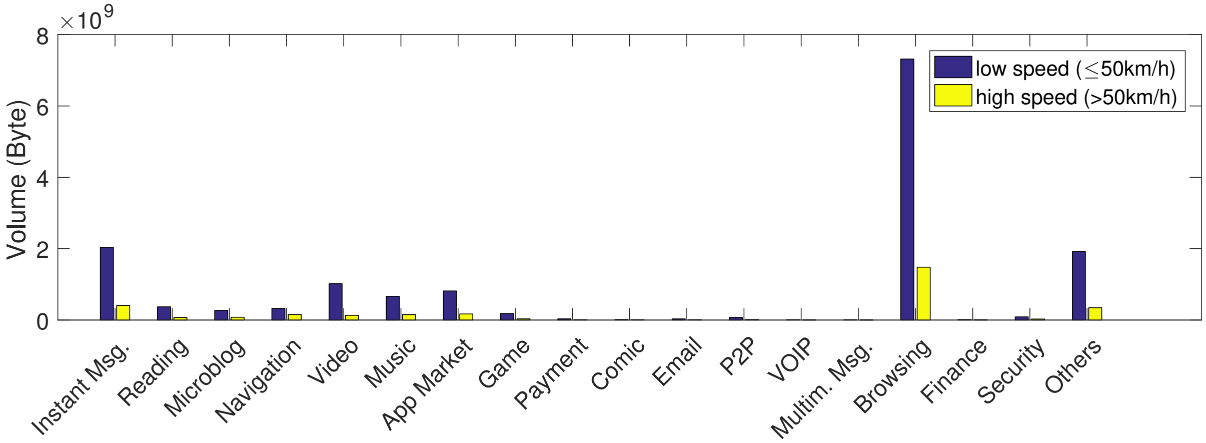
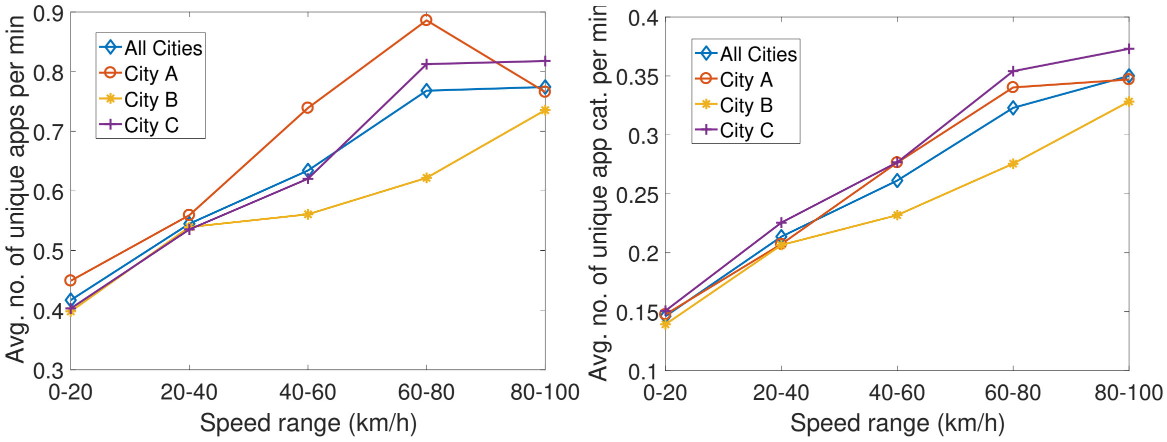


Fig. 14. The distribution of data volume per app category with high and low speed users

mobility, they may use their phones more often, use more kinds of apps, and be less likely to focus on one app for prolonged periods of time.

In Figure 16, we show the frequency of app switch and app category switch. They both follow the same trend that lower speed users tend to switch apps and app categories more frequently. Combined with Figure 15, we can reach the conclusion that although low speed users switch apps more frequently, they only switch between fewer number of apps.

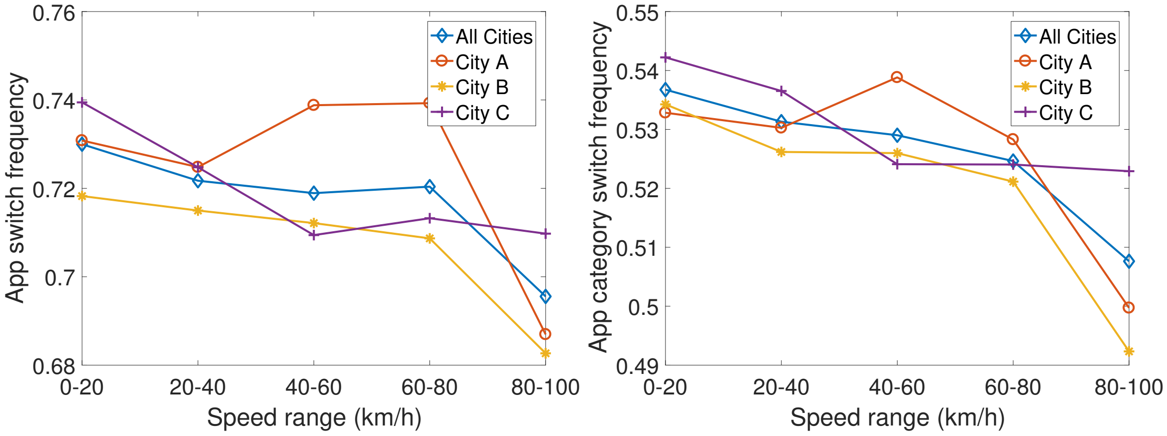
We then further extended our study by investigating the correlation of the number of concurrently running apps and app categories with estimated user speed. The result is shown in Figure 17. Although the actual launch time and close time for each running app are unavailable, we use the network access data to infer such information. For each app, we treat the time of the first data access as the launch time of an app. If the app stops accessing network for a certain period (close threshold), we record the last network access time as the close time for the app. We chose 10 seconds as the close threshold to plot Figure 17, but we had tested various close



(a) Number of unique apps used per minute

(b) Number of unique app categories used per minute

Fig. 15. Correlation of user speed and the number of unique apps and app categories used



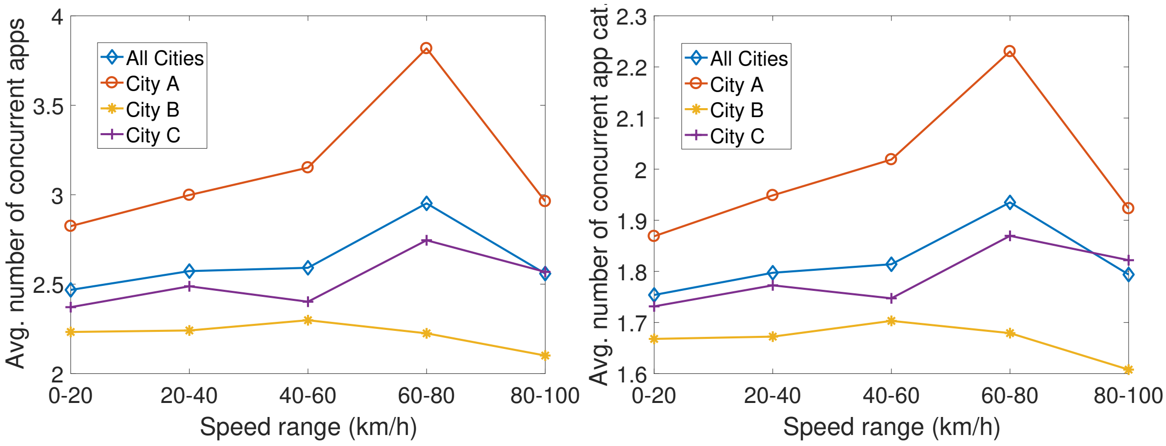
(a) App switch frequency

(b) App category switch frequency

Fig. 16. Correlation of user speed and the frequencies of app and app category switch

threshold and found similar trends except differences in value. In most cities, the peak happens at the 60-80 km/h and decreases as the speed keeps increasing. Currently, we do not have a direct explanation for this trend. We will continue this study in future works.

In the last, 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. Focusing on apps that contributed the most to the total mobile data access volume, we selected the top 8 app categories. The correlation between the user speed and the contribution of each category is shown in Figure 18.



(a) Average number of concurrently running apps (b) Average number of concurrently running app categories

Fig. 17. Correlation of user speed and the number of concurrently running apps and app categories

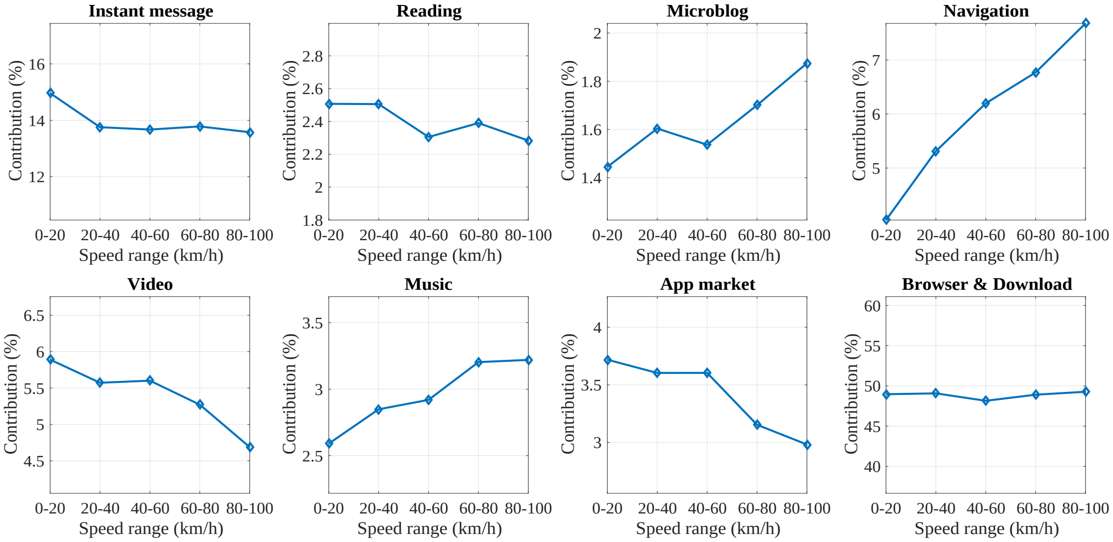


Fig. 18. Correlation of user speed and contribution of app categories.

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.

7 LIMITATION

We discuss three limitations of our work in this section. The first one is that our speed estimation method does not consider WiFi traffic. The second one is that our dataset only have limited time length and location area. The last one is that our speed estimation algorithm filtered out users that do not frequently move nor frequently use cell phone data networks.

7.1 (M4) WiFi data for speed estimation

Our speed estimation algorithm is solely based on the coarse-grained location information from cell phone data access traces. Although WiFi data is becoming more and more popular and its availability has greatly increased during the last decade, cell phone network still have better coverage compared to WiFi networks. Moreover, users on cell phone networks usually have better speed diverges compared to users using WiFi. So it's more valuable to study the usage pattern under different speed.

The coverage patterns of WiFi APs are very different from cell phone towers. They are usually heavily overlapped with each other in certain area and do not have any coverage on other areas. Due to these reasons, our speed estimation algorithms cannot be directly applied to WiFi data. However, WiFi APs usually have smaller coverage than cell phone towers. Compared to cell phone data, location information in WiFi data usually have better accuracy. It is possible to improve the accuracy of speed estimation of our algorithm with additional WiFi data. We are planning to pursue this line of research in our future work.

7.2 (M2b, R1-3) Limitation of dataset

There are two main limitations of our dataset. First, it only contains cell phone data of a three-hour period during a Sunday evening. With such a short time period, we are not able to perform in-depth study on temporal or spatial analysis, e.g., there is a significantly lack of data in office area. Second, although it contains data from three cities, they are all from same country and they are actually quiet near from each other. So, the result presented in Section 6 might be biased because of these two limitations. Nevertheless, the speed estimation method can be easily applied to similar datasets from other areas of any given time period. Actually, in our evaluation section, we use a smaller dataset from a whole different area to verify the performance of our speed estimation algorithm. Similar procedure can be used to study usage patterns with temporal and spatial contexts in those datasets without much difficulty. To this end, our experimental study serves more like a show case that reveals there are certain patterns between user's mobile data usage and their speed.

7.3 (M2c) Limitation of speed estimation

Our speed estimation algorithm utilize passing-boundary events as building blocks. This technique require users have visited at least three towers (can be repeated, i.e., A to B to A) in their trace. Also, the threshold in travel time will filtered out users that do not frequently access mobile networks. These limitations are introduced because of the nature of this kind of datasets, i.e., coarse-grained location information and irregular time intervals. With tower coordinates as the only location information, it is almost impossible to infer user location in a certain cell phone tower coverage area. So for a trace that have only visited less than three towers, the uncertain of travel distances will undoubtedly

make the speed estimates very inaccurate. Similarly, the uncertain of travel time of users who do not frequently use mobile network will also introduce large errors for speed estimates.

8 CONCLUSIONS

In this paper, we studied the correlation between user mobility and app usage patterns. In particular, we focused on users' moving speed as the key mobility metric. A key challenge addressed by our methodology is 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 played an important role in the contribution of each smartphone app categories on the total mobile data traffic.

REFERENCES

[1] Paul Baumann and Silvia Santini. 2014. How the availability of Wi-Fi connections influences the use of mobile devices. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 367–372.

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

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

[4] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *KDD*. 1082–1090.

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

[6] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility markov chains. In *MPM*. 3.

[7] Sean Gillies and others. 2013–. Shapely. (2013–). https://pypi.python.org/pypi/Shapely [Online; accessed ¡today¿].

[8] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2013. Accelerometer-based Transportation Mode Detection on Smartphones. In *SenSys*. 13:1–13:14.

[9] Sahar Hoteit, Stefano Secci, Stanislav Sobolevsky, Carlo Ratti, and Guy Pujolle. 2014. Estimating human trajectories and hotspots through mobile phone data. *Computer Networks* 64 (2014), 296–307.

[10] Shan Jiang, Gaston A Fiore, Yingxiang Yang, Joseph Ferreira Jr, Emilio Frazzoli, and Marta C González. 2013. A review of urban computing for mobile phone traces: current methods, challenges and opportunities. In *UrbComp*.

[11] Eric Jones, Travis Oliphant, Pearu Peterson, and others. 2001–. SciPy: Open source scientific tools for Python. (2001–). http://www.scipy.org/ [Online; accessed ¡today¿].

[12] Anthony LaMarca, Yatin Chawathe, Jeffrey Hightower, and Gaetano Borriello. 2004. CRAWDAD dataset intel/placelab (v. 2004-12-17). Downloaded from http://crawdad.org/intel/placelab/20041217/placelab. (Dec. 2004). DOI: http://dx.doi.org/10.15783/C7KS30 traceset: placelab.

[13] Kyunghan Lee, Joohyun Lee, Yung Yi, Injong Rhee, and Song Chong. 2010. Mobile data offloading: How much can WiFi deliver?. In *Proceedings of the 6th International COnference*. ACM, 26.

[14] Ilias Leontiadis, Antonio Lima, Haewoon Kwak, Rade Stanojevic, David Wetherall, and Konstantina Papagiannaki. 2014. From cells to streets: Estimating mobile paths with cellular-side data. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 121–132.

[15] Vincenzo Manzoni, Diego Maniloff, Kristian Kloeckl, and Carlo Ratti. 2010. *Transportation mode identification and real-time CO_2 emission estimation using smartphones*. Technical Report. SENSEable City Lab, Massachusetts Institute of Technology.

[16] Lei Meng, Shu Liu, and Aaron D Striegel. 2014. Analyzing the impact of proximity, location, and personality on smartphone usage. In *INFOCOM WKSHPS*. 293–298.

[17] Mathé Young Mosny. 2006. *Path Estimation Using Cellular Handover*. Bachelor of Science Thesis. Princeton University.

[18] Nielsen. 2014. 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.

[19] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. An Empirical Study of Geographic User Activity Patterns in Foursquare. In *ICWSM*. 570–573.

[20] H. Ohashi, T. Akiyama, M. Yamamoto, and A. Sato. 2014. Automatic trip-separation method using sensor data continuously collected by smartphone. In *ITSC*. 2984–2990.

[21] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using Mobile Phones to Determine Transportation Modes. *ACM Trans. Sen. Netw.* 6, 2 (2010), 13:1–13:27.

[22] Robert Ricci, Eric Eide, and The CloudLab Team. 2014. Introducing CloudLab: Scientific Infrastructure for Advancing Cloud Architectures and Applications. *USENIX ;login:* (2014).

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

[24] J. Ryder, B. Longstaff, S. Reddy, and D. Estrin. 2009. Ambulation: A Tool for Monitoring Mobility Patterns over Time Using Mobile Phones. In *CSE*, Vol. 4. 927–931.

[25] M Zubair Shafiq, Lusheng Ji, Alex X Liu, Jeffrey Pang, and Jia Wang. 2012. Characterizing geospatial dynamics of application usage in a 3G cellular data network. In *INFOCOM*. 1341–1349.

[26] Dongyoun Shin, Daniel Aliaga, Bige Tunçer, Stefan Müller Arisona, Sungah Kim, Dani Zünd, and Gerhard Schmitt. 2015. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems* 53 (2015), 76–86.

[27] Zbigniew Smoreda, Ana-Maria Olteanu-Raimond, Thomas Couronné, and others. 2013. Spatiotemporal data from mobile phones for personal mobility assessment. *Transport Survey Methods: Best Practice for Decision Making* 41 (2013), 745–767.

[28] Statista. 2016. 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/.

[29] Leon Stenneth, Ouri Wolfson, Philip S Yu, and Bo Xu. 2011. Transportation mode detection using mobile phones and GIS information. In *GIS*. 54–63.

[30] C. Tacconi, S. Mellone, and L. Chiari. 2011. Smartphone-based applications for investigating falls and mobility. In *PervasiveHealth*. 258–261.

[31] K. Waga, A. Tabarcea, Minjie Chen, and P. Franti. 2012. Detecting movement type by route segmentation and classification. In *CollaborateCom*. 508–513.

[32] Huayong Wang, Francesco Calabrese, Giusy Di Lorenzo, and Carlo Ratti. 2010. Transportation mode inference from anonymized and aggregated mobile phone call detail records. In *ITSC*. 318–323.

[33] Shuangquan Wang, Canfeng Chen, and Jian Ma. 2010. Accelerometer based transportation mode recognition on mobile phones. In *APWCS*. 44–46.

[34] Tingting Wang, Cynthia Chen, and Jingtao Ma. 2014. Mobile phone data as an alternative data source for travel behavior studies. In *Transportation Research Board 93rd Annual Meeting*.

[35] P. Widhalm, P. Nitsche, and N. Brändie. 2012. Transport mode detection with realistic Smartphone sensor data. In *ICPR*. 573–576.

[36] Peter Widhalm, Yingxiang Yang, Michael Ulm, Shounak Athavale, and Marta C González. 2015. Discovering urban activity patterns in cell phone data. *Transportation* 42, 4 (2015), 597–623.

[37] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. 2011. Identifying diverse usage behaviors of smartphone apps. In *IMC*. 329–344.

[38] Tingxin Yan, David Chu, Deepak Ganesan, Aman Kansal, and Jie Liu. 2012. Fast app launching for mobile devices using predictive user context. In *MobiSys*. 113–126.

[39] Jie Yang, Yuanyuan Qiao, Xinyu Zhang, Haiyang He, Fang Liu, and Gang Cheng. 2015. Characterizing User Behavior in Mobile Internet. *IEEE Transactions on Emerging Topics in Computing* 3, 1 (2015), 95–106.

[40] Lin Yang, Mingxuan Yuan, Wei Wang, Qian Zhang, and Jia Zeng. 2016. Apps on the move: A fine-grained analysis of usage behavior of mobile apps. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 1–9.

[41] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. 2010. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web* 4, 1 (2010), 1.