

Discovering SpatioTemporal Mobility Profiles of Cellphone Users *

Murat Ali Bayir[†]

Computer Science & Engineering
University at Buffalo, SUNY
14260, Buffalo, NY, USA
mbayir@cse.buffalo.edu

Murat Demirbas

Computer Science & Engineering
University at Buffalo, SUNY
14260, Buffalo, NY, USA
demirbas@cse.buffalo.edu

Nathan Eagle

MIT Media Laboratory
Massachusetts Institute of Technology
02139, Cambridge, MA, USA
nathan@mit.edu

Abstract

*Mobility path information of cellphone users play a crucial role in a wide range of cellphone applications, including context-based search and advertising, early warning systems, city-wide sensing applications such as air pollution exposure estimation and traffic planning. However, there is a disconnect between the low level location data logs available from the cellphones and the high level mobility path information required to support these cellphone applications. In this paper, we present formal definitions to capture the cellphone users' mobility patterns and profiles, and provide a complete framework, **Mobility Profiler**, for discovering mobile user profiles starting from cell based location log data. We use real-world cellphone log data (of over 350K hours of coverage) to demonstrate our framework and perform experiments for discovering frequent mobility patterns and profiles. Our analysis of mobility profiles of cellphone users expose a significant long tail in a user's location-time distribution: A total of 15% of a user's time is spent on average in locations that each appear with less than 1% of time.*

1 Introduction

Cellphones have been adopted faster than any other technology in human history [7], and as of 2008, the number of cellphone subscribers exceeds 2.5 billion, which is twice as

many as the number of PC users worldwide¹. To capture a slice of this lucrative market, Nokia, Google, Microsoft, and Apple have introduced cellphone operating systems (Symbian, Android, Windows Mobile, OS X) and open APIs for enabling application development on the cellphones. Recently, cellphones have also attracted the attention of the networking and ubiquitous computing research community due to their potential as sensor nodes for city-wide sensing applications [1, 5, 10, 12, 14, 15, 21].

Mobility path information of cellphone users play a central role in a wide range of cellphone applications, such as context-based search and advertising, early warning systems [3, 20], traffic planning [9], route prediction [16, 17], and air pollution exposure estimation [6]. Cellphones can log location information using GPS, service-provider assisted faux GPS or simply by recording the connected cellular tower information. However, since all these location logs are low level data units, it is difficult for the cellphone applications to access meaningful information about the mobility patterns of the users directly. To make mobility data more readily accessible to cellphone applications, higher level data abstractions are needed.

To address this problem, we focus on the problem of discovering spatiotemporal mobility patterns and mobility profiles from cellphone-based location logs. In particular, the contributions and novelities of this paper are listed as follows:

1. In order to capture the mobility behaviors of cellphone users at a level of abstraction suitable for reasoning and analysis, we introduce formal definitions for the concepts of *mobility path* (denoting a user's travel from

*This work was partially supported by NSF Career award #0747209
†corresponding author.

¹www.wirelessintelligence.com

one end-location to another), *mobility pattern* (denoting a popular travel for the user supported by her mobility paths), and *mobility profile* (providing a synopsis of a user’s mobility behavior by integrating the frequent mobility patterns, contextual data, and time distribution data for the user). Although human mobility has been studied in different contexts in previous works [11, 13, 18, 23, 24], those works were restricted to small scale environments such as building or a campus area and relied on WLAN technologies. In contrast, this paper focuses on analyzing human mobility in city wide level by using cellular networks.

2. We design and implement a complete framework, the **Mobility Profiler**, for discovering mobility profiles from raw cell tower connection data. Our framework addresses a commonly encountered phenomenon in real-world cellular networks, *cell tower oscillation*, where even when the user is static she may be assigned to a number of neighboring cell towers for load-balancing purposes or due to changes in the ambient RF environment. Our framework removes oscillation side-effects by determining oscillating cell tower pairs from the cellphone logs and grouping them in a single cluster. Furthermore our framework exploits the geometric nature of the problem to improve the performance of the discovery process: our framework first constructs a cell tower topology from the available mobility paths and then uses this topology to expedite the pattern discovery process by eliminating a majority of candidate path sequences as unrealizable (due to the topological constraints). In the same vein, our framework introduces new support criterias based on string matching to increase the algorithm’s performance during support checks for the mobility patterns.
3. We validate and demonstrate our framework by using the “Reality Mining” data set² containing 350K hours of cell tower connection data. Using this dataset, we perform comprehensive experiments to determine the thresholds for when to consider a location as an end-location versus an interim-location on a mobility path. We identify two types of end-locations, observable and hidden, and show that both of them are necessary for correct construction of mobility paths.
4. Finally, our analysis of the cellphone users’ mobility behaviors yields important lessons for networking researchers interested in testing large-scale ad-hoc routing protocols. As also identified in a recent study [8], we find that users spend approximately 85% of their time in 3 to 5 favorite locations, e.g., home, work, shopping. However, our analysis has exposed a more

interesting phenomena for the distribution of the remaining 15% of the users’ time. We identify a significant *long tail* in a user’s location-time distribution: **Approximately a total of 15% of a user’s time is spent in locations that each appear with less than 1% of time.**

Last but not least, the mobility profiles we generate for cellphone users include temporal information for patterns (which days of the week and which hours of the day) and time distribution data for all locations. These mobility profiles are useful for early warning systems and route prediction applications. By coupling the time-context with the mobility paths, these mobility profiles may be useful for the purposes of synthetic mobility scenario generation research.

Outline of the paper. The next section explains Reality Mining data set and mobility profiler architecture. Section 3 defines the mobility path concept, gives mobility path construction, mobility pattern discovery method, and construction of mobility profiles. The experimental results on the data set are presented in section 4, and conclusions in section 5.

2 Preliminaries

2.1 Reality Mining Data Set

The dataset for our work is collected by the Reality Mining project group from MIT Media Labs. The dataset comes from an experimental study involving 100 people for the duration of 9 months. Each person is given a Nokia 6600 cellphone with a software that continuously logs cell tower connectivity data. Due to the lack of GPS in the Nokia 6600, the location is recorded not in terms of an exact longitude-latitude pair, but rather in terms of the cell tower currently connected. In order to render the cell tower ids meaningful, the cellphone software prompts the user to provide a tag when it encounters a cell tower id for the first time. This way, some cell tower locations were able to be tagged semantically with a specific meaning for that user.

The logged data from all the cellphones total around 350K hours of monitoring time and fit into a database of 1GB size. The necessary data for our mobility profiler framework are stored in four tables. Figure 1 shows the database schema that presents the relation between these tables. The Cellspan table stores the connectivity information of a person to a cell tower. The Cellname table stores user-specific semantic tags for cell towers. cell tower and Person tables store all the cell tower and cellphone user information. The name field in the cell tower table denotes the cell tower’s broadcasted real name (a numerical id).

²<http://reality.media.mit.edu>

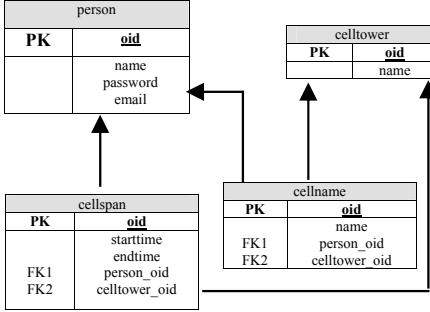


Figure 1: Reality Mining Database

In contrast to earlier work on the cellphone networks domain [8, 19, 22, 25] where the user location log data is provided by the service providers in a top down manner, in our study the location logs for the users are obtained from the cellphone side by a cellphone application. While the advantage of this approach is independence from service providers, the limited amount of information available thus poses several complications we need to address.

2.2 Overview of the Mobility Profiler Framework

Figure 2 illustrates the general architecture of our framework. We start with the “path construction” to construct ordered set of cell tower ids that correspond to a user’s travel from one end-location to another. Then, we apply “cell clustering” to gather the oscillating cell towers in the same group and replace the cell towers with their corresponding clusters so as to remove the oscillation problems on the paths. After the cell clustering, we apply the “topology construction” using the paths of cell clusters as input. The resultant topology information of clusters are employed for eliminating the majority of the candidate path sequences to expedite processing during the “pattern discovery” phase.

In the pattern discovery phase, we discover the frequent mobility patterns of each user separately. This task is executed efficiently by employing the topology information and a string matching support criteria (which we discuss later). In the “post processing” phase, we generate cellphone user profiles from their personal mobility patterns by adding the time-context information to the patterns and we generate time distribution data by using paths of cell clusters.

3 Mobility Profiler

In this section we present the five phases of the Mobility Profiler framework in detail.

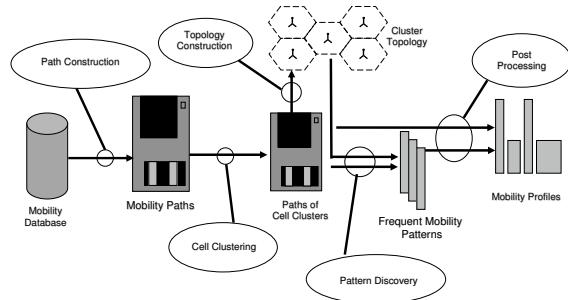


Figure 2: Mobility Profiler Framework

3.1 Path Construction Phase

Before we proceed to present the construction of the mobility paths for users, we give some basic definitions.

The connectivity information (of a person to a cell tower) stored in the Cellsapn table is gathered as follows. When a cell tower switching occurs, the end time for the previous cell tower is captured and a new record is created in the cellphone that contains the start and end time for that previous cell tower. Simultaneously, the start time for the new cell tower is recorded and is kept until the next cell tower switching occurs. There may also be an unaccounted time-gap between two cell tower switchings due to disconnection from all base stations or turning off the cellphone. To account for these, we define two time intervals:

Definition (Cell Duration Time): Cell duration time is the difference between end and start time for each cell span record L , that represents the connectivity information to a particular cell tower. The cell duration time for each cell span record is calculated as:

$$L_{dur}^k = L_{end}^k - L_{start}^k \quad (1)$$

Here L_{dur}^k is the cell duration time for k^{th} cell span record, L_{end}^k is the connection end time and L_{start}^k time is the connection start time for that entry.

Definition (Cell Transition Time): Cell transition time is the difference between the end and start time of two contiguous cell span record belonging to the same subject in the Reality Mining study (i -th user). The cell transition time is calculated as:

$$L_{(i)tra}^k = L_{(i)start}^{k+1} - L_{(i)end}^k \quad (2)$$

Here L_{tra}^k is the k^{th} cell transition time of the user, L_{end}^k is the connection end time for the $(k)^{th}$ cell-span record for that user and L_{start}^{k+1} time is the connection start time for $(k+1)^{th}$ cell-span record for the same user.

Definition (Observed End-Location): An observed end-location record corresponds to a cell tower location C_k

in the k^{th} cell-span record the duration time of which is greater than a predefined upper bound $\delta_{duration}$:

$$L_{dur}^k > \delta_{duration} \quad (3)$$

To illustrate consider a user arriving to her work place where she stays connected to a cell tower for 5 hours. When the user later leaves for home, a cell switching occurs. Since $L_{dur} = 5$ hours is larger than $\delta_{duration}$ time (of say 10 minutes) the cell location C_k is accepted as an end-location and the id of the corresponding cell tower is marked as an observed end location.

Definition (Hidden End-Location): A hidden end-location between two contiguous cell span record k^{th} and $(k+1)^{th}$ corresponds to a location H_k in which the user stayed longer than a predefined upper bound $\delta_{transition}$:

$$L_{(i)tra}^k > \delta_{transition} \quad (4)$$

This inequality states that a hidden location occurs when a significant amount of time is elapsed during cell transition. To illustrate, consider a user that switches off her cellphone at a movie theater and then switches on it at home after 3 hours. Since the transition time (3 hours) exceeds the threshold $\delta_{transition}$ (say 10mins), we say that the user has been in a hidden end-location H_k for these time intervals. The same case occurs when user is out of cellphone connectivity range for a significant amount of time.

Note that the Cellspan table does not store “related” cell-span records together. The main idea of the mobility path is to group cell span records together to correspond to users’ travel from one end-location to another. We define mobility path formally as follows:

Definition (Mobility path): A mobility path $C = [C_1, C_2, C_3, \dots, C_n]$ is an ordered sequence of cell tower ids corresponding to the cells that a user visited during her travel from one end-location to another. The mobility path must satisfy the following two rules:

End Location Rule:

- $\forall C_k \in C, L_{dur}^k > \delta_{duration} \Rightarrow k = 1 \text{ or } k = |C|$

Transition Time Rule:

- $\forall C_k, C_{k+1} \in C \Rightarrow L_{start}^{k+1} - L_{end}^k < \delta_{transition}$

The first rule states that the observed end-locations can only be the first or last locations of the mobility path. (Since the paths may also be terminated due to a hidden end-location, the dual of this rule is not true.) This rule also implies that for any location that is neither the first nor the last location, the duration time should be smaller than or equal to the predefined maximum cell duration threshold $\delta_{duration}$. The intuition behind this rule is that if a cellphone user stays for a significant amount of time in a cell

area C_k , then C_k should be taken as an end-location and the current path should be terminated.

The second rule states that the elapsed time for each cell tower transition within the path should not be greater than a predefined threshold $\delta_{transition}$. Thus, a cellphone user can not visit a hidden end-location within the path, otherwise the current path is terminated. The intuition behind the second rule is that if a user stays a significant amount of time outside cellphone connectivity, she may travel to locations that are not captured. In that case, merging hidden locations with previous locations increases the error and leads to noisy data in the paths.

The pseudocode for our path construction and example run are given in our technical report [4].

3.2 Cell Clustering

A major problem with the cellular network connectivity data is that a cellphone may dither between multiple cells even when the user is not mobile. A similar problem is also addressed in the Wi-Fi networks referred as the ping-pong effect [18].

The approach mentioned in [18] proactively asserts a simple hexagonal tiling of the cells to restrict the oscillation patterns between at most three immediate cell neighbour of a point. However, in cellular networks, the geometry can not constraint to be hexagonal tilings, and more significantly most oscillations are due to load balancing purposes and involves arbitrary number of neighbouring cell towers. Therefore, we have proposed two phased reactive approach (decide oscillation after mining real data) to solve this problem. Due to the space limitations we relegate the details of our algorithm for identifying the oscillating pairs in a given mobility path to our technical report [4].

3.3 Topology Construction

Topology construction is used to eliminate majority of candidate path sequences during the pattern discovery phase. In general, pattern discovery problem is solved by an exponential time algorithm, which may take a significant amount of time to execute. By employing the cell cluster neighborhood topology during pattern discovery, the candidate sequences which can not possibly correspond to a path on the cell cluster topology graph can be eliminated without calculating their supports.

Since we have user mobility paths as input, the cell cluster topology construction is an easy process by one scan through these paths. In this process, an edge between the cell cluster pairs C_k and C_{k+1} is created if both of them exist in any path in contiguous positions.

3.4 Pattern Discovery

In this phase, frequent mobility patterns are discovered from mobility paths. Although not the most recent or the most efficient one in the literature, we use a modified version of the AprioriAll [2] technique. This technique is suitable for our problem since we can make it very efficient by pruning most of the candidate sequences generated at each iteration step of the algorithm using the topological constraint mentioned above: for every subsequent pair of cell-clusters in a sequence, the former one must be neighbour to the latter one in the cell-cluster topology graph. We call this new version of AprioriAll as Sequential Apriori Algorithm. An important criteria in our domain is that a string matching constraint should be satisfied between two sequences in order to have support relation. For example, the sequence $< 1, 2, 3 >$ does not support $< 1, 3 >$ although 3 comes after 1 in both of them. However, sequence $< 1, 3, 2 >$ supports $< 1, 3 >$. A path S supports a pattern P if and only if P is a subsequence of S not violating the string matching constraint. We call all the paths supporting a pattern as its support set.

Sequential Apriori Algorithm (Algorithm 1): In the beginning, each cell cluster with sufficient support forms a length-1 supported pattern. Then, in the main step, for each k value greater than 1 and up to the maximum reconstructed path length, candidate patterns with length k+1 are constructed by using the supported patterns (frequency of which is greater than the threshold) with length k and length 1 as follows:

- If the last cell cluster of the length-(k) pattern is incident to the cell cluster of the length-1 pattern, then by appending length-(1) cell cluster, length-(k+1) candidate pattern is generated.
 - If the support of the length-(k+1) pattern is greater than the required support, it becomes a supported pattern. In addition, the new length-(k+1) pattern becomes maximal, and the extended length-(k) pattern and the appended length-(1) pattern become non-maximal.
 - If the length-(k) pattern obtained from the new length-(k+1) pattern by dropping its first element was marked as maximal in the previous iteration, it also becomes non-maximal.
- At some k value, if no new supported pattern is constructed the iteration halts.

Note that in the sequential Apriori algorithm, the patterns with length-k are joined with the patterns with length-1 by considering the topology rule. This step significantly eliminates many unnecessary candidate patterns before even calculating their supports, and thus increases the performance drastically.

Algorithm 1 Sequential Apriori

```

1: input: Minimum support frequency:  $\delta$ , Paths of clusters: S
2: Topology Matrix: Link, The Set of all Cell Clusters: C
3: output: Set of maximal frequent patterns: Max
4: procedure sequentialApriori ( $\delta$ , S, Link, C)
5:    $L_1 := \{\}$  // Set of frequent length-1 patterns
6:   for  $i:=1$  to  $|C|$  do
7:      $L_1 := L_1 \cup [C_i]$  if Support( $[C_i], S$ )  $> \delta$ 
8:   for  $k = 1$  to  $N - 1$  do
9:     if  $L_k = \{\}$  then
10:       Halt
11:     else
12:        $L_{k+1} := \{\}$ 
13:       for each  $I_i \in L_k$ 
14:         for each  $C_j \in C$ 
15:           if Link[LastCluster( $I_i$ ),  $C_j$ ] = true
16:              $T := I_i \bullet C_j$  // Append  $C_j$  to  $I_i$ 
17:             if Support( $T, S$ )  $> \delta$  then
18:                $T.maximal := TRUE$ 
19:                $I_i.maximal := FALSE$  // since extended
20:                $V := [T_2, T_3, \dots, T_{|T|-1}]$  // drop first element
21:               if  $V \in L_k$  then
22:                  $V.maximal := FALSE$ 
23:                  $L_{k+1} := L_{k+1} \cup \{V\}$ 
24:              $Max := \{\}$ 
25:             for  $k := 1$  to  $N - 1$  do
26:                $Max := Max \cup \{S \mid S \in L_k \text{ and } S.maximal = \text{true}\}$ 
27: end procedure

```

An auxiliary function $\text{Support}(I:\text{Pattern}, S)$ determines whether a given pattern has sufficient support from the given set of reconstructed user paths. Support of a pattern I is defined as a ratio between the numbers of reconstructed paths supporting the pattern I, the number of all paths.

$$\text{Support}(I, S) = \frac{|\{S_i \mid \forall i \text{ } I \text{ is substring of } S_i\}|}{|S|} \quad (5)$$

3.5 Representing Mobility Profiles

Frequent mobility patterns containing only location information and lacking any time-context information are inadequate for several applications, including route prediction, early warning systems, and user clustering. Therefore, we add time-context information to the frequent patterns in order to represent mobile user profiles.

Definition (Mobility Profile): A mobility profile for a cellphone user includes personal mobility patterns with contextual time data and distribution of spatiotemporal lo-

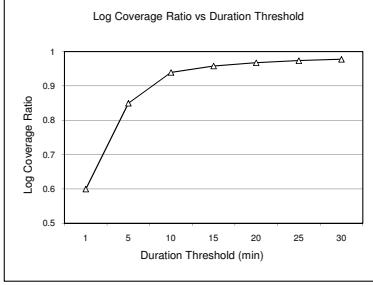


Figure 3: Duration Time Analysis

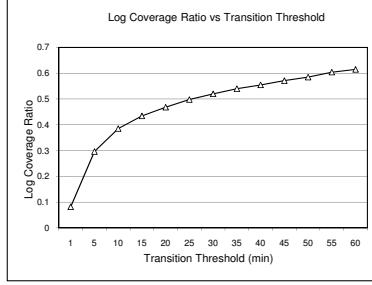


Figure 4: Transition Time Analysis

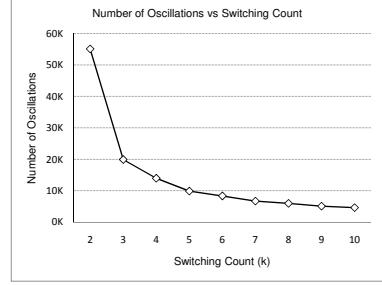


Figure 5: Switching Count Analysis

cations for that user. The time contextual data for mobility patterns are specified in two dimensions:

- **Days of Week:** Each frequent pattern stores its distribution over days of week. That means, the frequent pattern is tagged with the number of its instances observed on each day of the week.
- **Time Slices:** Each frequent pattern stores its distribution over each time slices given in the set {[12:00 a.m., 6:00 a.m.], [6:00 a.m., 12:00 p.m.], [12:00 p.m., 6:00 p.m.], [6:00 p.m., 12:00 a.m.]}. That means, the frequent pattern is tagged with the number of its instances started on each of these time slices.

Apart from the spatiotemporal mobility patterns, mobility profile of each user contains time distribution data of all locations visited by current user. The time distribution data is important since it identifies the importance of each location as proportional to the time spend on them.

4 Experimental Results

In this section, we will present our experimental results on MIT reality mining data set containing 350K hours of cellspan data. For analyzing MIT Reality Mining data, we have implemented Mobility Profiler Framework on Java Environment. The size of the source code for the whole framework is around 4KLOC. Our implementation contains a separate module for each of the phases discussed above.

In the rest of this section, first we give our results for determining duration and transition threshold, that are used for constructing mobility paths. For cell-clustering, we give our analysis for finding minimum switch count. For the pattern discovery phase, we present examples of interesting patterns discovered from Reality Mining data and give a case study for representing mobile user profile. We also provide interesting results related to the average time distribution of the locations for all users.

4.1 Determining End Location Thresholds

Suitable values for $\delta_{duration}$ and $\delta_{transition}$ need to be identified before executing the path construction phase. These two threshold values are determined by analyzing the ratio of cell span duration and cell span transitions that are smaller than predefined time values in experiment space. For determining $\delta_{duration}$ time, we have defined our experimental duration time space as a set {1, 5, 10, 15, 20, 25, 30} minutes and evaluated the ratio of cellspan records the duration time of which are smaller than these 7 discrete values in our set. This experiment is plotted in Figure 3. In this graph, the plot-line shows a very small tangent after duration time=10 min which has ratio value of 0.94. Analyzing the left part of the duration threshold=10 min, we see a significantly sharp drop of about 10%. Thus, we take the static time threshold as $\delta_{duration}=10$ min. One can argue that there may be non-end locations in which cellphone user stays more than 10 minutes. For example, a user may wait 15 minute in bus stop which can be intermediate location during trip from school to home. However, as it is shown from our graph, this type of behavior shows rarely since all of the locations the duration time of which is greater than 10 min $[10, infinity)$ lies between [0.94, 1.00) in terms of log ratio.

For determining $\delta_{transition}$ time, we define our experimental space as a set with 13 different time values from 1 minute to 60 minutes. We do not take higher values than 60 minutes since it is reasonable to accept the existence of hidden end locations if transition time is more than 60 minutes. In order to find acceptable value for $\delta_{transition}$ time, we use the ratio metric that is mentioned above for analyzing $\delta_{duration}$ time. Unlike the analysis of $\delta_{duration}$ time, there is still some visibility problem if we analyze this data without filtering the regular handoffs that take 0 seconds. In reality mining data set, nearly, 99.2% of contiguous cellspan records has regular handoff value that is 0 second that means the cellphone handles 99.2% of cell tower switches immediately. It is obvious that the user can not

Id	Pattern Name	Frequency
1	<Home, Media Lab>	0.279
2	<Media Lab, Home>	0.265
3	<XXX CommonWealth, Media Lab>	0.133
4	<Home, Charles Hotel, Media Lab>	0.060
5	<Media Lab, Charles Hotel, Home>	0.053

Figure 6: Mobility patterns for user X

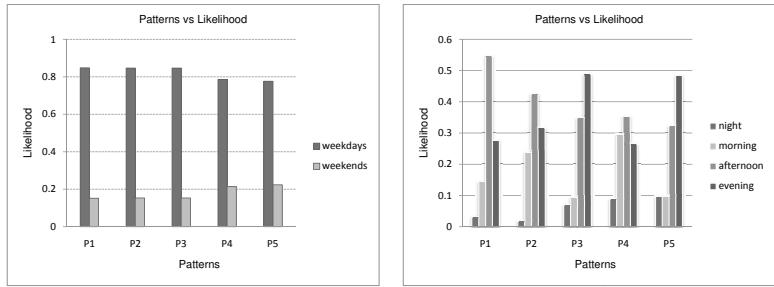


Figure 7: Days of Week Analysis

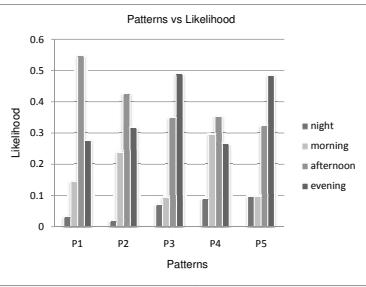


Figure 8: Time Slice Analysis

be in hidden end location in this time range. Therefore, we filter regular handoff times for analyzing $\delta_{transition}$. The result of the second experiment is given in Figure 4. In this graph, we notice that the tangent of line after threshold time 10 minutes is greater than one in the Figure 3 for $\delta_{duration}$ time. However, we notice that the tangent of the line is constant after 10 minutes threshold time until 60 minutes. In each neighbor point after 10 minutes, the increase in the log coverage ratio is around 2-3%. When we analyze the left part of transition threshold=10 min, we see a significantly sharp drop of about 10%. Thus, we accept 10 minutes as a reasonable threshold for $\delta_{transition}$ time. This is also a good choice as it relates to the duration time threshold for determining end-locations.

4.2 Cell Clustering

After determining $\delta_{duration}$ and $\delta_{transition}$ values as 10 minutes, we executed the path construction phase over 2.5M cell-span records resulting in approximately 120K mobility paths. However, these paths included a significant amount of noisy data due to cell tower oscillations not correlated with human mobility.

For solving the oscillation problem mentioned above, we cluster the cell towers by using their location tags. Each cluster is named by using majority voting over the locations names of its cell towers. For assigning untagged cell towers to the clusters, oscillating pairs of untagged cell towers are discovered. As mentioned in the clustering section we need minimum switching count to find the oscillating pairs. Therefore, we have performed an experiment on determining minimum switching count k . In this experiment, we count the number of oscillations with respect to different switching counts from $k = 2$ to $k = 10$. The results of this experiment is provided in Figure 5. As seen from Figure 5, the tangent of the plot-line decreases as k becomes larger. In fact, when moving on the x axis from infinity to zero. The biggest jump occurs when switching from point $k = 3$ to $k = 2$. We believe that the number of oscillations due to natural user mobility (which should be distinguished from

cell tower oscillations) significantly contributes for $k = 2$. Thus, in order to better distinguish between oscillations due to user mobility and cell tower oscillations, we take the minimum switching threshold $k = 3$.

After using $k = 3$ as the switching threshold, we find the oscillating pairs of untagged cell towers and assigned them to a cluster as discussed before.

4.3 Finding Maximal Mobility Patterns

We executed the pattern discovery phase for generating both global and personal frequent patterns. For the global pattern discovery, we have used a frequency support of $\delta = 0.001$ which means that each pattern should exist in at least 120 path over 120K total paths to be considered. Since the frequency of mobility paths is inversely correlated with the path-length, the length of the most frequent paths are usually two or three. However, the overall distribution of path length is more spread. In fact the average pattern length is around 4.8.

We performed the personal pattern discovery over paths of single user (user X) as our case study. The number of paths for user X is around 2K. We chose the frequency threshold as $\delta = 0.005$ which means that each pattern should exist in at least 10 mobility paths. The top five mobility patterns for user X are given in Figure 6.

4.4 Representing Cellphone User Profiles

Here we present our experimental results for mobility profiling on user X. The top five mobility patterns for X are plotted in Figure 7 and 8 on two different time domains (day of weeks and time slices). We also analyzed spatiotemporal distribution of visited locations for user X in Figure 9.

Figure 7 shows the distribution of all five patterns over weekdays and weekends. All of the top-5 patterns are active on weekdays with a balanced distribution over the 5 work days. The peak time for the first, second, and fourth patterns are afternoons whereas the peak time for the third and fifth patterns are evenings (Figure 8). As mentioned in section

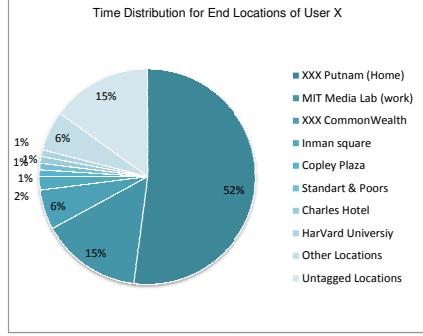


Figure 9: Time distribution for end locations for user X

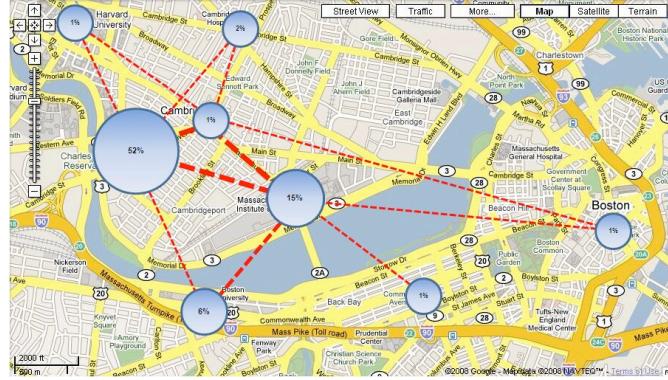


Figure 10: Time distribution for end locations on map for user X

IV, the user profiles give significant information about cell-phone user behaviors. For example, on a Tuesday afternoon if user X is at cell area tagged as "XXX CommonWealth," with high probability she will go to cell area tagged "Media Lab" next. These mobility profiles have the potential of yielding more accurate results for location prediction problem as they include an additional time dimension.

We have also analyzed the spatiotemporal distribution of locations for user X in Figure 9. Although it may first appear that there is no need to construct mobility paths and perform clustering to extract these spatiotemporal locations, mobility path construction is a very important step for generating an accurate and noise-free time distribution chart, and we have used the mobility paths for user X for constructing the time distribution chart. Mobility paths gather related cell span connectivity records together, and makes it possible to determine and analyze the oscillations and clustering among the cell towers. Replacing cell towers with corresponding clusters within these paths enables us to calculate the time elapsed on each cluster location accurately for the time distribution chart.

Figure 9 shows that user X spends 67% of her overall time at home or work. In fact, 79% of overall time elapsed at 8 different locations for user X. An even more interesting phenomenon is found when we consider the distribution of the remaining 6% (others) for user X in Figure 9. These remaining 6% of user X's time is spent in locations that each appear less than 1% of time: there are 69 different locations for user X in that portion. In other words the spatiotemporal distribution for user X shows a very heavy/long tail. We corroborated this finding in all users' spatio temporal distributions: **approximately 15% of the users time is spent in a large variety of locations that each appear less than 1% of total time**. We present a graph of the number of locations with respect to coverage ratios in Figure 11. In this figure a point (1%, 15%) means that on average 15% of total time elapsed on the locations in which the user spend

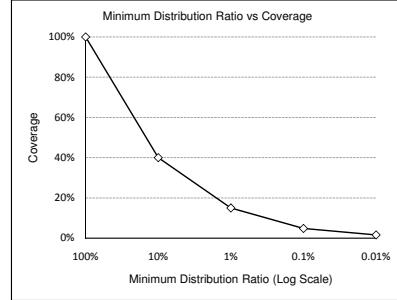


Figure 11: Minimum Distribution Ratio vs Coverage

less than 1% of total time. Since this graph is in logarithmic scale, it is possible to see clearly that there is a 15% heavy tail after 1% minimum distribution ratio. Indeed, the coverage ratio approaches zero only after two more logarithmic scales from that point. The average number of locations that remain in the 15% heavy tail area is more than 800, whereas it is around 12 for the remaining 85% portion.

One implication of this find is that, while simulating/testing large-scale mobile ad-hoc protocols, it is not sufficient to simply take the top-k popular locations. Doing so will discard about 15% of a user's visited locations.

5 Conclusion and Future Work

In this paper, we have proposed a complete framework for discovering mobile user profiles. We have defined the mobility path concept for cellular environments and introduced a novel path construction method. We have also proposed a cell clustering method that provides robustness against noises, such as cell tower oscillations and improper handoffs containing time delays. From the experimental results over 350K hours real data, we have shown that our framework is capable of producing user profiles that can be used for city wide sensing applications. Our analysis also

discovered a long tail for human mobility behavior: approximately 15% of a person's time is spent in a large variety of locations each of that takes less than 1% time.

As future work, we are going to work on a similar framework that uses GPS data to discover mobile user behaviors. We will also investigate the opportunities for using our mobility profiles in new applications, such as social networking, car insurance estimation and peer to peer file applications over smartphones.

References

- [1] T. F. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. J. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.
- [2] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.
- [3] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.
- [4] M. A. Bayir, M. Demirbas, and N. Eagle. Mobility profiler: A framework for discovering mobile user profiles. *Technical Report, Department of Computer Science and Engineering, University at Buffalo*, available at <http://www.cse.buffalo.edu/tech-reports/2008-17.pdf>, 2008.
- [5] J. Burke and et al. Participatory sensing. In *ACM Sensys World Sensor Web Workshop*, 2006.
- [6] M. Demirbas, C. Rudra, A. Rudra, and M. A. Bayir. Imap: Indirect measurement of air pollution with cell-phones. In *PerCom*, pages 537–542, 2008.
- [7] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 04-2:28–34, 2005.
- [8] M. Gonzalez, C. Hidalgo, and A. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.
- [9] A. Harrington and V. Cahill. Route profiling: putting context to work. In *SAC*, pages 1567–1573, 2004.
- [10] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *Sensys*, pages 125–138, 2006.
- [11] W. jen Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling time-variant user mobility in wireless mobile networks. In *INFOCOM*, pages 758–766, 2007.
- [12] A. Kansal, M. Goraczko, and F. Zhao. Building a sensor network of mobile phones. In *IPSN*, pages 547–548, 2007.
- [13] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *INFOCOM*, 2006.
- [14] D. Kirovski and et al. Health-os: A position paper. In *ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, 2007.
- [15] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *IPSN*, pages 481–492, 2008.
- [16] K. Laasonen. Clustering and prediction of mobile user routes from cellular data. In *PKDD*, pages 569–576, 2005.
- [17] K. Laasonen. Route prediction from cellular data. In *CAPS*, pages 147–158, 2005.
- [18] J.-K. Lee and J. C. Hou. Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application. In *MobiHoc*, pages 85–96, 2006.
- [19] J. Markoulidakis and et al. Mobility modeling in third-generation mobile telecommunication systems. *IEEE Personal Comm.*, pages 41–56, 1997.
- [20] N. Marmasse and C. Schmandt. A user-centered location model. *Personal and Ubiquitous Computing*, 6(5/6):318–321, 2002.
- [21] N. Oliver and F. Flores-Mangas. Mptrain: a mobile, music and physiology-based personal trainer. In *Mobile HCI*, pages 21–28, 2006.
- [22] C. Ratti, A. Sevtsuk, S. Huang, and R. Pailer. Mobile landscapes: Graz in real time <http://senseable.mit.edu/graz/>.
- [23] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *In Proc. of IEEE INFOCOM*, 2008.
- [24] C. Tude and T. Gross. A mobility model based on wlan traces and its validation. In *INFOCOM*, pages 664–674, 2005.
- [25] M. Zonoozi and P. Dassanayake. User mobility modeling and characterization of mobility pattern. *IEEE J. Selec. Areas Commun.*, 15 (7):1239–1252, 1997.