Inferring Human Mobility from Sparse Low Accuracy Mobile Sensing Data

Andrea Cuttone  
DTU Compute  
Dept. of Applied Mathematics and Computer Science  
Richard Petersens Plads, Bldg. 324  
2800 Kgs Lyngby, Denmark  
ancu@dtu.dk

Sune Lehmann  
DTU Compute  
Dept. of Applied Mathematics and Computer Science  
Richard Petersens Plads, Bldg. 324  
2800 Kgs Lyngby, Denmark  
sljo@dtu.dk

Jakob Eg Larsen  
DTU Compute  
Dept. of Applied Mathematics and Computer Science  
Richard Petersens Plads, Bldg. 324  
2800 Kgs Lyngby, Denmark  
jaeg@dtu.dk

Abstract
Understanding both collective and personal human mobility is a central topic in Computational Social Science. Smartphone sensing data is emerging as a promising source for studying human mobility. However, most literature focuses on high-precision GPS positioning and high-frequency sampling, which is not always feasible in a longitudinal study or for everyday applications because location sensing has a high battery cost. In this paper we study the feasibility of inferring human mobility from sparse, low accuracy mobile sensing data. We validate our results using participants’ location diaries, and analyze the inferred geographical networks, the time spent at different places, and the number of unique places over time. Our results suggest that low resolution data allows accurate inference of human mobility patterns.

ACM Classification Keywords
H.2.8 [Database Management]: Database Applications – data mining

Author Keywords
Spatial data mining; mobile sensing; place discovery; location sensing; mobility;
Introduction
Understanding human mobility has a number of important applications, ranging from containment of infection diseases [4] to urban planning and traffic management [8]. Moreover mobility plays an important role in understanding face-to-face [1] and long distance social interactions [12].

Most studies on mobility have been using Call Detail Records (CDR) as proxy for location [15, 7, 3], inferring people’s positions depending on the cell towers that mobile phones are connected to. Such data sources can only produce very rough location estimates with space granularity of kilometers and time granularity of hours. Thanks to the explosive adoption of smartphones equipped with location sensors, mobile sensing data is becoming a promising alternative for inferring patterns of human mobility and social interaction [14].

An extensive literature on mining location data has been produced, but a large amount of this work focuses on high-precision GPS positioning and high-frequency sampling, with location collected every few minutes [20, 21] or even every few seconds [2, 19, 10, 18, 11, 5, 17]. Such collection methods are not realistic in a longitudinal study based on smartphone data, since high frequency GPS sampling would lead to rapid battery drain. For this reason, in the Copenhagen Network Study [16] we collect location data in an opportunistic fashion using Android smartphones and the Google Location API¹, which provides an improved battery life at the price of lower sampling frequency and more variable accuracy. Understanding data collected at a fixed rate is a necessary baseline and point-of-comparison for more advanced low energy approaches, such as adaptive sampling [9]. Thus, in this paper we focus on the specific problem of inferring human mobility from such sparse, low accuracy mobile sensing data. We perform a small experiment using diaries collected by 6 participants and one researcher, and we motivate that it is possible to infer mobility patterns with reasonable accuracy.

Related work
A number of recent studies on inferring mobility from mobile sensing data have been performed, and they differ for number of participants, frequency of sampling, and data mining techniques. Here we cite a few representative examples. Ashbrook et al. [2] collect data for 6 users at 1 sample per second; places are extracted using k-means and evaluated by user interviews. Zhou et al. [21] collect data for 28 users at 1 sample per minute, and extract places using DBSCAN clustering. Palma et al. [13] propose a speed-based DBSCAN clustering, but give no user evaluation of the results. Montoliu et al. [11] collect data for 8 participants using continuous sampling; locations are identified using grid clustering, and the results evaluated comparing with participants diaries. Cao et al. [6] analyze the GPS traces of 119 cars, and exploit GPS signal loss for detecting stops; the inferred network of places is analyzed. Yan et al. [17] determine stops using a dynamic speed thresholding and places are extracted using intersection with geometry; results are evaluated for 6 users.

Problem definition
We collect location data as a sequence of samples represented by tuples in the format \((\text{timestamp}, \text{latitude}, \text{longitude}, \text{accuracy})\) ordered by non-decreasing \text{timestamp}. A point of interest (POI) is a location of relevance for a person, such as his home, his workplace or a gym he frequents. A POI can be described as a tuple

¹http://developer.android.com/google/play-services/location.html
(POI-id, POI-latitude, POI-longitude), where POI-latitude and POI-longitude refer to the centroid of the POI. A stop at a POI is a specific occasion when the person has visited a POI in a given day and hour. A stop is thus described by a tuple \((\text{timestamp-arrival}, \text{timestamp-departure}, \text{POI-id})\). By definition, each stop is related exactly to one POI, but one POI can be related to many stops. Stops are non-overlapping, that is \(\text{timestamp-arrival}_{i+1} > \text{timestamp-departure}_i\) for every stop. A stops extraction algorithm accepts as input a sequence of location samples, and produces as output a set of stops related to POI.

Given the discovered stops and the ground truth stops, we need to compute how many actually match. In several related studies [21, 11, 17, 2] the participants were interviewed and asked to manually match the discovered locations with the personal diaries. In our case we set threshold parameters to define when a match occurs, trying to emulate human judgment. A person asked to decide whether two stops are matching would probably say yes if they are reasonably close to each other and the time interval was approximately the same. Therefore we set the following threshold parameters: we use a distance of 150 meters (roughly the average size of one city block), and we adopt a time threshold equal to 25% of the total stop duration, so that shorter stops must be identified more precisely than longer stops. Although it is possible to argue for different choices of these parameters, we found that different values (distance threshold = 50 and 100 meters, and time threshold = 10%) have only minor effect on the results. Let \(B_{\text{stops}}\) be the baseline stops (stops in the diary), \(D_{\text{stops}}\) the discovered stops (stops extracted by an algorithm), and \(B_{\text{stops}} \cap D_{\text{stops}}\) the baseline stops which are discovered and recognized as matching. Let:

\[
\text{recall}_{\text{stops}} = \frac{|B_{\text{stops}} \cap D_{\text{stops}}|}{|B_{\text{stops}}|}
\]

\[
\text{precision}_{\text{stops}} = \frac{|B_{\text{stops}} \cap D_{\text{stops}}|}{|D_{\text{stops}}|}
\]

\[
f1_{\text{stops}} = 2 \cdot \frac{\text{precision}_{\text{stops}} \cdot \text{recall}_{\text{stops}}}{\text{precision}_{\text{stops}} + \text{recall}_{\text{stops}}}
\]

In a similar fashion we evaluate the performance of identifying POI by calculating \(f1_{\text{POI}}\), considering a match if the centroids are at distance < 150 meters.

**Experimental settings**

We recruited a total of 6 participants, all of them students at our campus. Participants were provided with a Samsung Galaxy Nexus smartphone, with a collector app based on the Funf Open Sensing framework [1]. Participants were instructed to use the provided smartphone as their main phone. We asked participants to keep a diary of their daily movements, and we provided them an electronic spreadsheet where they could fill entries in the format: date, hour, place description. We instructed them to keep the diary updated as accurately as possible, and we sent weekly reminders via email. One author also carried a Samsung Galaxy Nexus smartphone for collecting his own data and kept his own location diary. Table 1 shows the summary statistics of diary entries for each participant (P1-P6) and for the researcher (R). The participants’ data was collected in October and November 2013, while the researcher’s data was collected in four periods between September and January 2014. At the end of the study, the participants were asked to create a list of POI with description, latitude and longitude obtained using Google Maps².

²http://maps.google.com
Table 1: Summary statistics of diary entries for each participant (P1-P6) and for the researcher (R).

<table>
<thead>
<tr>
<th>User</th>
<th>Stops</th>
<th>$\mu$ (Stops/day)</th>
<th>$\sigma$ (Stops/day)</th>
<th>Unique POI</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>120</td>
<td>2.9</td>
<td>1.3</td>
<td>33</td>
</tr>
<tr>
<td>P2</td>
<td>127</td>
<td>3.2</td>
<td>1.5</td>
<td>34</td>
</tr>
<tr>
<td>P3</td>
<td>187</td>
<td>4.0</td>
<td>1.9</td>
<td>37</td>
</tr>
<tr>
<td>P4</td>
<td>130</td>
<td>2.9</td>
<td>1.3</td>
<td>49</td>
</tr>
<tr>
<td>P5</td>
<td>61</td>
<td>2.3</td>
<td>0.9</td>
<td>18</td>
</tr>
<tr>
<td>P6</td>
<td>111</td>
<td>4.4</td>
<td>2.3</td>
<td>22</td>
</tr>
<tr>
<td>R</td>
<td>227</td>
<td>3.5</td>
<td>1.4</td>
<td>39</td>
</tr>
</tbody>
</table>

The usage of diaries as ground truth presents some challenges, as participants' compliance tends to decrease over time, and the process of filling a diary every day can be tedious, so participants may avoid or simply forget to do it. Moreover the concepts of stops and POI are complex and often ambiguous. Finally, any human task is error-prone, and by manually inspecting the diary entries we noticed self-evident errors such as sequence of stops in wrong temporal order and typos in descriptions, dates and times.

Data collection

Location samples are collected with a custom version of the Funf Open Sensing framework [1], which requests one sample every 15 minutes using the Android location API. We have indications that these settings do not significantly impact power consumption, since participants did not report reduced battery life using our collector app. The location is provided either by GPS positioning, Wi-Fi or cellular networks, depending on availability. Location is acquired in an opportunistic manner, so every time another app requests a location, this sample is also recorded in our system. The collected data is temporarily stored on the phone, and periodically uploaded to our servers.

This sampling method presents several challenges, including the variability of accuracy, the unpredictability of samples arrival, and the presence of outliers and duplicates. We firstly investigate the quality of the raw location data. We calculate the empirical cumulative distribution function (CDF) for the samples accuracy, and we find that the vast majority (>90%) of samples have accuracy better than 60 meters.

We then analyze the time distribution of samples. We calculate the CDF of the time in seconds between samples $\Delta t = \text{timestamp}_{i+1} - \text{timestamp}_i$. Figure 1 shows that around 60% of the intervals of time between samples are under 10 seconds, 80% are under 60 sec, and 90% are under 200 seconds.
Even though many samples are collected with high frequency, they are unevenly distributed in time. In fact the opportunistic sensing settings produce a pattern of burst collection, where a rapid sequence of samples is collected, and then none for a longer period of time. To investigate what is the probability of finding at least one sample in a given time interval, we divide the samples into time bins, and count what fraction of bins are empty for different bin sizes. Figure 2 shows that the fraction of empty bins decreases for larger bin sizes, with a large fraction of empty bins for smaller sizes. Starting from 900 seconds the fraction of empty bins becomes approximately constant.

This shows that in most cases the samples collected opportunistically tend to be highly redundant as they capture approximately the same location in a very short timespan.

**Stops detection algorithms**

In this section we describe three simple algorithms for extracting stops from location data, based on methods from the literature: distance grouping, speed thresholding and Gaussian Mixtures Model (GMM) clustering.

Each algorithm accepts as input a sequence of location samples and produces as output a sequence of stops. We discard stops with duration shorter than a minimum time, since very short stops may be inferred from the location data but they are not meaningful in human terms. Although the choice of what is the minimum time duration of a stop is quite subjective, we chose the minimum time as 15 minutes, since the very large majority of diaries entries have duration greater than 30 minutes.

Each algorithm has parameters that influence the stop extraction, and ultimately determine the $f_1$ performance. We are interested in how estimating the parameters on a set of participants would generalize to all others, therefore we perform cross-validation. We select the parameters that perform best for one participant and we calculate the scores for the remaining participants using these parameters. We repeat the operation for a each participant, and we calculate the average $f_1$. This procedure is used for each algorithm.

**Distance grouping**

The distance grouping algorithm is built on the idea that a stop corresponds to a temporal sequence of locations within a maximal distance $d_{\text{max}}$ from each other. Locations are examined sequentially by non-decreasing timestamp. Each stop initially contains only one location $\text{loc}_i$, and each subsequent location $\text{loc}_{i+k}$ is added to the
stop until \( \text{distance}(\text{loc}_{i+k}, \text{loc}_i) < d_{\text{max}} \). Then the process starts again from \( \text{loc}_{ci+k+1} \). The choice of the parameter \( d_{\text{max}} \) influences the detection of the stops. A large \( d_{\text{max}} \) tends to merge more location samples and produce less stops of longer duration, while a smaller \( d_{\text{max}} \) tends to merge less locations and produce more stops of shorter duration. An initial increase in \( d_{\text{max}} \) leads to better \( f_1 \) performance since smaller stops are correctly merged into more meaningful larger stops. For a range of values the \( f_1 \) score stabilizes around an optimal value, as increasing \( d_{\text{max}} \) does not result in merging more stops. Finally for larger values of \( d_{\text{max}} \), even very far stops are merged together, resulting in worse performance shown by the lower \( f_1 \) score.

**Speed thresholding**

Speed thresholding uses the calculated speed of movement to classify location samples as stops and moves. Given two successive geographical positions we calculate \( \text{speed}_{i+1} = \frac{\text{distance}(\text{pos}_{i+1}, \text{pos}_i)}{\text{timestamp}_{i+1} - \text{timestamp}_i} \). In our dataset however the sample-by-sample speed oscillates widely, due to the variability of the frequency and accuracy of samples. We are instead interested in a smoother speed estimate, in order to detect transitions between places. Therefore we create time bins of size \( T \), and for each bin we consider the position as the median of all the samples in the bin. We then calculate the speed between bins. Using the speed information, we can discard samples with \( \text{speed} > \text{speed}_{\text{max}} \), and then group consecutive static locations into stops. The choice of the bin size \( T \) and of the \( \text{speed}_{\text{max}} \) threshold influence the performance of the algorithm. A longer period produces smoother transitions but may fail to detect shorter stops. A lower \( \text{speed}_{\text{max}} \) threshold will result in discarding a larger number of location samples and consequently to more information loss.

**Gaussian Mixtures Model**

The distance grouping and the speed thresholding algorithms take a sequential approach to the stops detection, as they examine samples one by one in temporal order. A different approach is to look at the overall distribution of the samples independently of time, and identify POI as clusters of location samples with higher density. A Gaussian Mixtures Model (GMM) assigns samples to clusters modeled as a finite number of Gaussian distributions with unknown parameters. Once each sample is assigned to a cluster, we group temporally consecutive samples with the same cluster label into stops. The condition that determines the cluster assignment is the \( \text{min.covariance} \) parameter. A larger \( \text{min.covariance} \) tends to produce larger clusters, while a smaller \( \text{min.covariance} \) produces smaller clusters.

**Results and Discussion**

In this section we provide the results of the \( f_1\text{stops} \) and \( f_1\text{POI} \) scores. Since our work focuses on sparse opportunistic sampling, we cannot compare directly with results in the existing literature, which are based on data sets with high-frequency, high-precision sampling. Additionally, most studies provide either an informal performance assessment based on aggregated data [6, 13, 5, 17], or limit their evaluations to the overall identification of POI [20, 21, 19, 18, 11]. Here, we directly evaluate how well we can infer the full stop-by-stop sequence. This is a more challenging task, which shifts the focus to evaluating the feasibility of inferring mobility.

**Evaluation of stop extraction**

We apply the three algorithms for stops extraction for each participant (P1-P6) and for the researcher (R), and we obtain the \( f_1\text{stops} \) scores (Figure 3).
Figure 3: \( f_{1_{\text{stops}}} \) scores for each participant (P1-P6) and for the researcher (R).

For each participant, the three algorithm have a substantially similar performance. The scores suggest that it is possible to infer mobility patterns with reasonable accuracy, despite the very sparse nature of the collected data.

There is an evident difference between the researcher and participants, with the researcher’s score being much higher than any participants’ scores. To estimate an upper limit of how well diaries can be matched against the location data, we compare the sequence of locations inferred from the data against the sequence inferred from the diary. In order to generate a sequence of positions from the location data we create time bins of size 900 seconds and take the median position for each bin, while for each diary entry we generate one position at the corresponding POI for every 900 seconds. We compare the two sequences as follows: for each time bin we count a match if the distance between the diary position and the inferred position is < 100 meters. The we calculate an overlap score as:

\[
\text{overlap} = \frac{\text{matches}}{\text{bins}}
\]

We find a significant correlation between \( f_{1_{\text{stops}}} \) score and overlap (\( \rho_{\text{distance}} = 0.939 \), \( \rho_{\text{speed}} = 0.944 \) and \( \rho_{\text{GMM}} = 0.941 \)). We suggest that the higher \( f_{1_{\text{stops}}} \) score for the researcher is probably due to a better quality of the ground truth. The researcher’s data was curated to be as precise as possible and record every single stop, while it is quite likely that participants have not been as consistent in their diaries.

**Evaluation of POI extraction**

The stop extraction algorithms produce a sequence of stops, which can then be grouped into clusters corresponding to POI. In the case of GMM, the stops are automatically marked with a poi-id associated with the corresponding cluster. The distance grouping and speed thresholding algorithms instead do not provide any information about groups of stops. In this case we infer the POI membership for stops applying DBSCAN clustering to the stops extracted by distance grouping. We set \( \text{minpts} = 1 \) and we determine the optimal \( \epsilon_{\text{db}} \) by cross-validation, selecting the parameter value that performs best for one participant and calculating the \( f_{1_{\text{POI}}} \) scores for the remaining participants using this value.

Figure 4 shows the \( f_{1_{\text{POI}}} \) for the GMM and the DBSCAN algorithms. The results are consistent with our findings about \( f_{1_{\text{stops}}} \). The two algorithms have similar performance for the same participant, but there is a quite significant difference between participants’ and researcher’s scores. Also in this case there is a correlation with overlap, which again suggests that the lower participants score is due to inaccurate ground truth. Finally if we compare the \( f_{1_{\text{stops}}} \) and \( f_{1_{\text{POI}}} \) for each
participant, we see that the $f_{1\text{POI}}$ is better, which is probably because it is easier to infer POI than stops.

Figure 4: Comparison of the $f_{1\text{POI}}$ for the GMM and the DBSCAN algorithms.

Mobility analysis
In this section we further validate the discovery of stops and POI by performing an analysis of the mobility patterns. We consider the results obtained using the GMM algorithm, although similar results are obtained using the other algorithms.

We first look at the geographical network of the 20 POI where most time is spent. We draw a node to represent the geographical coordinates of each stop at a POI, and one arc for each transition between stops. We find that the two networks are quite similar, although some POI are notably missing from the discovered network. Figure 5 shows the comparison of the geographical network as inferred from the diary (left) versus the one from the discovered stops (right) for one participant.

We then look at the distribution of time versus POI. It is common knowledge that we spend the most of our lives at home and at work, and rest of our time is shared between all other locations, some of which are regularly visited. Several studies have confirmed this notion, as they found that the distribution of time spent at different locations is heavy-tailed [7]. For each participant, we compute the time spent at POI both from the diary stops and from the discovered stops. The two time distributions are remarkably close, and both show heavy-tailed distributions, with the 2 most visited POI (presumably home and work) corresponding to the majority of time. Figure 6 shows as example the comparison for one participant.
Finally, we calculate the number of unique POI over time. For each timestamp, we count the unique POI visited by the stops between timestamp0 and timestampi. We find that for all participants the two distributions follow a similar trend, although in some periods the discovered POI are fewer than the POI in the diary.

Conclusions
We have evaluated the feasibility of inferring human mobility from sparse low accuracy mobile sensing data. We have performed a comparison with ground truth in form of location diaries, and evaluated in details the inferred mobility in terms of geographical networks, time spent at different places, and the number of unique places over time. Our results indicate that it is possible to infer the mobility patterns accurately, despite the sparse nature of the data. As future work, different algorithms could be tested to see if the detection performance can be improved. We also find a significant difference in performance between the researcher’s and the participants’ scores, and we suggest that it may be caused by the inaccuracies in the participants’ diaries which seem to be a limiting factor for accurately studying human mobility.

Having indication of the feasibility of inferring mobility, we intend to apply these procedures to the dataset of nearly one thousand participants that we are actively collecting [16]. We believe that replacing raw location data with meaningful points of interest will allow us to understand human mobility at a level much closer to the human point of view.

Acknowledgments
This research was supported in part by the Villum Foundation through the High Resolution Networks grant.

References
have you been? using location clustering and context awareness to understand places of interest. In 


