Automatic Identification of Relevant Places from Cellular Network Data

Marco Mamei[†], Massimo Colonna[‡], Marco Galassi[†]

[†] *DISMI, University of Modena and Reggio Emilia, Italy*
[‡] *Engineering & Tilab, Telecom Italia, Italy*
marco.mamei@unimore.it, massimo.colonna@telecomitalia.it, marco.galassi@unimore.it

Abstract

We present a methodology to automatically identify users' relevant places from cellular network data[1]. In this work we used anonymized Call Detail Record (CDR) comprising information on where and when users access the cellular network. The key idea is to effectively cluster CDRs together and to weigh clusters to determine those associated to frequented places. The approach can identify users' home and work locations as well as other places (e.g., associated to leisure and night life).We evaluated our approach threefold: *(i)* on the basis of groundtruth information coming from a fraction of users whose relevant places were known, *(ii)* by comparing the resulting number of inhabitants of a given city with the number of inhabitants as extracted by the national census. *(iii)* Via stability analysis to verify the consistency of the extracted results across multiple time periods. Results show the effectiveness of our approach with an average 90% precision and recall.

Key words: Mobility patterns, CDR data, place identification.

1. Introduction

The widespread diffusion of mobile phones and cellular networks provides a practical way to collect location-based information from a large user pop-

Corresponding author: email: marco.mamei@unimore.it, phone: +390522522233, fax: +390522522312

[1] The approach described in this paper is the subject of Patent Application PCT/EP2014/058003 filed by Telecom Italia on April 2014

ulation. The analysis of such collected data is a fundamental asset in the development of several services and applications. In particular, we propose an innovative approach to automatically identify the places that people routinely frequent (e.g., home, work place, favourite nightlife locations) from the analysis of anonymized positioning data from a cellular network (i.e., CDR – Call Detail Record). The knowledge of such relevant places finds important applications in mobile services, marketing, traffic forecasting, urban planning and management services [1, 2, 3, 4] (see Section 2 for more details).

The basic approach to extract places from CDR data, adopted by almost all the state of the art, is illustrated in Figure 1. **(1)** CDRs of each users are collected. **(2)** CDRs are clustered in well specified spatial regions. **(3)** Clusters are then weighed on the basis some factors (e.g., number of days in which the user visits the cluster). **(4)** Clusters with a weight greater than a certain threshold are associated to relevant places. Our proposal improves the state of the art in all the three phases: clustering, weighing, thresholding.

1. **Clustering.** Some of the approaches at the state of the art try to identify relevant places on the basis of the network cell from where most traffic is generated [5, 6, 7] (i.e., they cluster CDRs by the cell in which they are generated). This approach however tends to be inefficient in that, in areas covered by multiple cells, a cell phone will split CDRs among different neighboring cells. Other approaches recognize this issue and focus on clusters of cells spanning a given area [1, 8]. Our clustering mechanism uses a spatio-temporal distance function to better identify areas that are visited by the user (see Section 4.3).
2. **Weighing.** While most of the approaches just count the number of CDRs or the number of days with CDRs to determine the "importance" of an area, we refine such a classification by combining different measures together. We use weights to take into account the time of CDRs, the number of days, and how evenly CDRs are spread across the week (see Section 4.4).
3. **Thresholding.** While most of the approaches assume the presence of a single home and a single work place for each user, our approach tolerates multiple homes, work and other kind of places, or even none of them. We developed a flexible thresholding scheme to identify relevant clusters (see Section 4.5).

In addition, the majority of the approaches at the state of the art focuses on home and work locations only. They try to identify home locations on

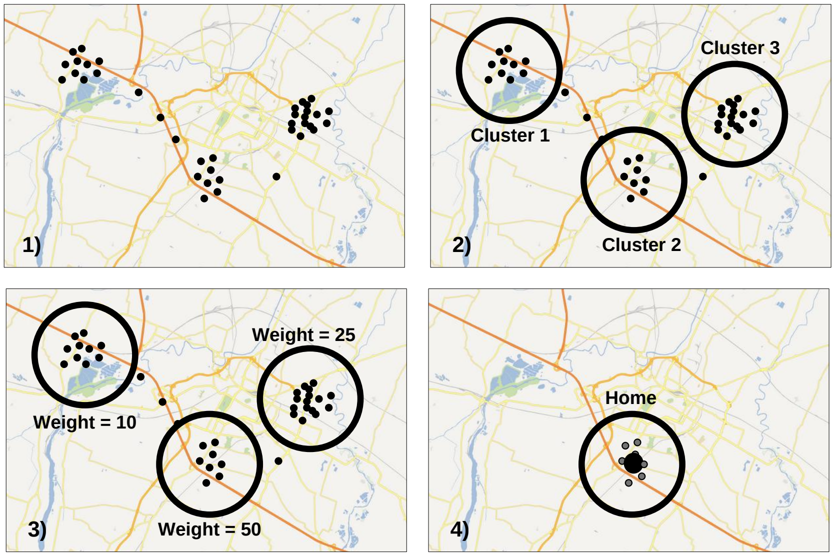

Figure 1: General methodology to identify frequented places. (*1*) CDRs for each user are selected. (*2*) A clustering algorithm groups CDRs into well-defined spatial regions. (*3*) A weighing mechanism gives each cluster a weight on the basis some aspects. (*4*) Clusters with a weight greater than a certain threshold are associated to relevant places.

the basis of CDRs generated at night, while work locations on the basis of CDRs generated during the day [5, 6, 1, 8, 7]. Such approaches use a fixed threshold to identify night and day time intervals. In our approach, we use a more flexible approach and try to extract also other kind of places.

In the following of this paper we first motivate the present work, describe applications scenarios and associated privacy issues (Section 2). Then we present related work in the area and show how our proposal improves over such existing approaches (Section 3). We describe in detail the proposed methodology (Section 4), and we present analysis and results' evaluation (Section 5). Finally we present some concluding remarks.

2. Motivations and Privacy Issues

The automatic identification of users' places is a relevant research topic in mobility data analysis and enables a number of application and services: *(i)* extracted places could be the basis to understand mobility patterns in the area. Specifically, from home and work places it is possible to automatically infer recurrent commuting trips. Then, aggregating those trips it is possible to obtain origin-destination matrices and traffic flow over an area [3, 1]. This naturally supports urban planning and traffic management. *(ii)* Knowledge of recurrent places can enable novel services targeting people visiting specific areas [4]. For example, it is possible to personalize outdoor advertising and information billboards on the basis of the places visited by the users in that area. Similarly, an application (like *TripAdvisor.com*) could rank places on the basis of the number of people routinely frequenting those places. More in general, such a knowledge could enable novel services in smart city scenarios [9]. *(iii)* The knowledge of places visited by people (i.e., the typical distribution of people in the city) could be also very important in emergency and disaster-response situations to prioritize interventions and manage mobility on the basis of the actual people distribution [10].

Telecom operators are uniquely positioned to extract such information from cellular network data. However, despite this potential for innovation, the analysis of CDRs to extract places frequented by people raises many concerns about individuals' privacy, rightfully constraining their use. In the European Union, the legal framework ruling on this kind of analysis (one of the stricter in preserving individuals' privacy) is described in the *EU Data Protection Directive (Dir. 95/46/EC)* and in the *Article 29 Data Protection Working Party (WP 29).* These directives are then implemented by member states' laws. In Italy (where our analysis is set) the implementing law is the *Italian Personal Protection Code (law 196/2003).*

Without mentioning a number or (important) subtleties, the key aspect of this set of laws can be summarized as (cf. *WP 29 Opinion 3/2013*): When the analysis aims at predicting the personal preferences, behavior and attitudes of individuals, *in order to inform measures or decisions that are taken toward them*, the relevant consent (opt-in) is necessary. Vice versa, analysis aiming at detecting trends, correlations and aggregate measures in the information, *without effects on single individuals*, can be performed provided that anonymization measures should be taken to ensure that the data are not available to support measures or decisions toward individuals.

Accordingly, to lawfully conduct this analysis, CDRs identifiers (i.e., user's id) must be pseudo-anonymized (i.e., masked via one-way hashing) and – most importantly – results can be provided only at an aggregate level so that no individual can be singled out. For example, while it is lawfully possible to create an application inferring the home locations of all the individuals to compute aggregated demographic distributions, it is not possible

Ref	Clustering	Weighing	Thresholding
[11]	Group by cell	max interval w/o cell changes	maximum
[5]	Group by cell	number of days	maximum
[6]	Group by cell	number of CDR	maximum
[7]	Group by cell	number of CDR	maximum
[1]	Grid	number of days	maximum
[8]	Cluster on distance	vector of temporal parameters	logistic regression

Figure 2: How related works address the three phases: clustering, weighing, thresholding

to create an application providing an individual with information on the basis of his/her locations (without his/her explicit consent).

Following these rules, a number of telecom operators started to provide innovative services on the basis of information extracted from CDR data (e.g., Telefonica SmartSteps – *dynamicinsights.telefonica.com* and Vodafone mAnalytics *tinyurl.com/pcbkds4*). Similarly, some companies were able to acquire CDRs form telecom operators to create novel services (e.g., Airsage – *www.airsage.com* and Positium – *www.positium.com*).

3. Related Work

Fueled by the recent availability of Telecoms' CDR data, a number of researchers recognized the application potential in automatically identifying users' important places (see Figure 2).

The work in [11] aims at identifying home places from the analysis of CDR data. For each user, the approach identifies the longest time period without antenna changes (which occurs mostly at night). Then, the home of the user is located in the area around that antenna.

The work in [5] proposes another approach to identify home and work places from CDR data. For each user, the cells where the user generates CDRs are selected. Then, for each cell, the average start time of calls is calculated. The cell with the highest number of days with calls and having the average start time of calls after 5pm is the home-cell. The cell with the highest number of days with calls and having the average start time of calls before 5pm is the work-cell. Home and work places are estimated to be in the areas around the home-cell and the work-cell respectively.

Similarly, the work in [6] classifies the place around the antenna in which the user produces most CDRs from 9pm to 7am as the user's home. While the place around the antenna in which the user produces most CDRs from 9am to 12am and from 2pm to 5pm is the user's work place.

The above three approaches are evaluated by comparing results with census, results show a good correlation between the two estimates.

The work in [7] proposes an approach for automatic residential localization from CDR data. This approach is based on a training phase to identify the time frame that best represents the "calling from home" behavior. In particular a genetic algorithm framework is used to find the best time frame. The result of this first phase is a set of days and time periods (e.g., Mon - Fri, 9pm - 3am) where CDRs from home locations are typically produced. The approach then marks the home location as the area around the cell tower that produces most CDRs in the specified time frame. The approach is able to identify user residential areas with 65% accuracy.

The main advantage of our approach with respect to these other approaches is that we do not focus the analysis on individual cells, but on clusters of cells. This is a notable improvement in that we avoid fragmenting the CDRs generated in close areas into separate bins. Accordingly, our approach tend to provide better and more stable results.

The work in [1] divides the area under investigation in a grid where the side of every cell is 500 meters long. For each cell of the grid, they count the number of nights the user connects to the network in the nighttime interval (6pm - 8am) while in that cell, and select as a home location the cell with the greatest value. Similarly, the work place is estimated as the most frequent cell on weekday mornings between 8am and 10am. The approach is evaluated by comparing estimated results with census statistics, result shows a good correlation between the two estimates.

The work in [8] proposes an approach to identify important places in people's lives and in particular home and work places. Similarly to our work, the approach clusters the antennas on the basis of their mutual distances. Then it applies logistic regression on the basis of when CDRs happened to determine whether a cluster represents an important place or not. Finally, home and work places are identified as those important clusters having the majority of CDRs between 7PM and 7AM (for home) and 1PM and 5pm (for work places). The approach is able to identify 75% of homes within 1.28 miles, and 75% of work places within 2.3 miles.

The innovations introduced in our work aim at improving over such a state of the art by means of better clustering, weighting and thresholding.

In addition to the above works specifically addressing CDR data, there is a large literature addressing the problem of identifying users' relevant locations on the basis of other location-based traces. Mainstream approaches are either based on segmenting and clustering GPS-traces to infer what are the places relevant to the user [12, 13], or on detecting places and mobility on the basis of nearby RF-beacons such as WiFi and GSM towers [14, 15, 16]. In general, it is very difficult to apply those mechanisms to CDR data, as the sampling frequency of CDR data is much lower.

4. Methodology

For each kind of place to be identified, our approach follows the general methodology described in Figure 1. In the following of this Section we present the key elements of the proposed approach: *(i)* we provide some details on the CDR dataset at the basis of our proposal (see Section 4.1). *(ii)* We describe the function we use to assign a weight to each CDR. (see Section 4.2). *(iii)* We discuss the approach we used to cluster CDRs in well-defined areas (see Section 4.3). *(iv)* We present our mechanism to assign an importance weight to clusters (see Section 4.4). *(v)* Finally, we describe an automatic thresholding approach to identify those clusters associated to relevant places (see Section 4.5).

4.1. CDR Data

We obtained a large set of mobility data from an Italian telecom operator. In particular, we analysed data from two regions of Italy (Piemonte and Lombardia inhabited by about 15 millions people), spanning 10 months (March – December 2012).

Mobility data is obtained from Call Detail Records (CDR) and Mobility Management (MM) procedure messages (i.e., IMSI attach/detach and Location Update). CDRs are routinely collected by cellular network providers for billing purposes. A CDR is generated every time a phone places or receives voice call or a text message. The IMSI attach/detach procedure marks the phone as attached/detached to the network on power up/power down of the phone or SIM inserted/removed. Location updates are messages exchanged for keeping the network informed of where the phone is roaming. CDR and MM messages are read on network interfaces through specific probes and also contain the identity of the phone, the identity of the cell through which the phone is communicating and the related timestamp. As MM messages, for

User	MMC	Time	Cell	Coord	Radius
3dd2b	222	7346286	123	(41.2,13.9)	450

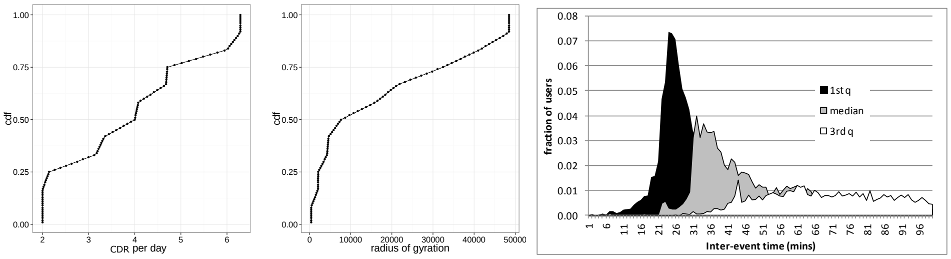

Figure 3: *(top)* Structure of our CDR dataset. Every time a user send or receive calls and text messages the network generates one CDR with information about the user (hashed) id, the MMC (Mobile Country Code), the timestamp of the CDR, the code of the cell tower and the coordinates and coverage radius of the cell tower. (*bottom-left*) Daily average number of CDR produced for a given percentile of users. (*bottom-center*) Radius of gyration for a given percentile of users. (*bottom-right*) Distributions of the media, first quartile, and third quartile of individual inter-CDR time

the purposes of our study, contain the same information as CDRs, we will refer to all these data as CDR data.

In the context of this work, all this information serves as sporadic samples of the approximate locations of the phone's owner. Specifically, the user's location is given in terms of the cell sector the user was connected with. The area covered by a given sector can be approximated by a circle with a given center and radius. In the following we use the term *cell* to actually refer to the cell sector. In Figure 3-top it is shown the structure of a CDR. Each record comprises a user (hashed) id , the MCC (Mobile Country Code) representing the country where the SIM card has been registered, the timestamp of the CDR, the code, and the coordinates and coverage radius of the cell. Thus, the spatial resolution of CDR localization is the cell radius.

It is worth noticing that differently from a number of other works we do not estimate the coverage of a network cell by using Voronoi tessellation. We stick to the simpler representation of a cell being represented by a circle with a given center and radius. In [17], it is shown that the approach do not change the user's location accuracy.

Figure 3-bottom-left illustrates the daily average number of CDR produced for a given percentile of users. We analyzed a sample of 1 month of data comprising more than 4 million users. It is possible to see that there is a fraction of the users producing a large number of CDRs per day, thus allowing tracking their mobility.

Figure 3-bottom-center illustrates the radius of gyration for a given percentile of users. The radius of gyration is a synthetic parameter describing the spatial extent of user traces. It is defined as the deviation of user positions from the corresponding centroid. It is given by: $r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - p_{centroid})^2}$ where p_i represents the i^{th} position recorded for the user and $p_{centriod}$ is the center of mass of the user's recorded displacements obtained by: $p_{centroid} = \frac{1}{n}\sum_{i=1}^{n}(p_i)$. It is possible to see that almost half of the user are urban dweller with r_g less than 10Km. Users in the (50^{th}-75^{th}) percentiles can be associated to urban commuters as the diameter of peri-urban areas of main cities in the region is about 25-30Km. Users beyond the 75^{th} percentile are associated to commuters travelling region-wide.

As a more detailed analysis, we tried to characterize the individual calling activity for a sample of 10000 users. For each user, we measured the inter-CDR time - i.e., the time interval between two consecutive network connections. The average inter-CDR time measured for the sample population is 241 minutes. This number is large because it considers the whole daily lives of that users, thus also spanning night gaps. We also measured the average inter-CDR time considering only CDR generated during the day. With that assumption the average inter-CDR time reduces to 52 minutes. Because the distribution of inter-CDR times for a user spans several temporal scales, we further characterized each calling activity distribution during the day by its first and third quartile and the median. Figure 3-bottom-right shows the distribution of the first and third quartile and the median for all the users. The arithmetic average of the medians is 64 minutes (the geometric average of the medians is 51 minutes) with results small enough to detect changes of location where the user stops for about 2 hours.

On the basis of these results, it seems possible to identify the places frequented by people by analyzing CDRs.

4.2. CDR Time-based Weight Function

The most important initialization parameter to identify a given kind of place is a function $w()$ associating for each day of the week and for each hour how likely the user is in that place at that day and time. For example, to identify the home place, the method will be initialized with a function having higher values associated to nights of the week. This function generalizes the fixed time intervals associated to home and work places adopted by the related works. The idea is that by avoiding sharp intervals results are more stable and robust. We will use this function to weigh CDRs of the user. CDRs produced on a weekday night are much more likely to come from the home place than CDRs produced in the early afternoon, when the user might be somewhere else. The function can be either set by a domain expert, can be configured for specific application purposes, or can be learnt from data (see appendix A).

4.3. Clustering

Clustering is a fundamental operation in our approach. As the CDRs produced in a given place can be associated to different antennas, it is fundamental to correctly cluster all such CDRs together. In fact, if the clustering is too fine-grained CDRs produced at a given place might be split among multiple clusters. Viceversa, if the clustering operation is too coarse, a given cluster will contain CDRs actually produced at a given place, but also CDRs produced at different nearby places. In both the cases, the estimated frequentation of a given place (i.e., cluster) is altered (as it will contain too few or too many CDRs) and results will be inaccurate.

One of the most important part of any clustering mechanism is the function measuring the distance between two CDRs. The distance between CDRs is in fact the key parameter to establish if the CDRs should be in the same cluster or not. To the best of our knowledge, all the approaches at the state of the art [1, 8] use the geographic distance between CDRs as the basis for clustering. The key limitation with this approach is that the geographic distance tends to be rather ineffective in our scenario. For example, let us assume that the user works in a given area, and also frequents a bar in the same area at night. Any clustering mechanism based on geographic distance will cluster together the CDRs produced from both the places. This “merging” could alter the estimated frequentation of the cluster affecting place recognition. As another example, if the user leaves home at about 8am by driving a car, CDRs produced around 8am will be scattered over a long distance. However, in a certain sense, they all belong to the “home” cluster, but a clustering mechanism based on geographic distances will be unable to put them together.

Distance Function. To improve clustering results, we propose to use a spatio-temporal distance function. The idea is that CDRs produced at completely different times of day should be far away even if they are physically close. Vice versa, CDRs that are produced at similar times should be closer than if they would be produced at completely different times.

More in detail, we define the following distance function: given two CDRs cdr_i and cdr_j generated at cells with location l_i and l_j and radius r_i and r_j, at times t_i and t_j respectively, we define a spatial distance function sd as: $sd(cdr_i, cdr_j) = max(0, geo(l_i, l_j) - (r_i + r_j))$

Where geo is the geographic distance between the two locations. It is worth noting that subtracting the sum of the radii is a “discount” factor so that large cells, typically found in the suburbs where distance among them is large, appear closer. It is also worth noting that two intersecting cells will have 0 spatial distance. Given the function $w()$ associating for each day of the week and for each hour the weight of the CDR (defined in Section 4.2), we define the following temporal distance function: $td(cdr_i, cdr_j) = |w(t_i) - w(t_j)|$

The idea of this function is that the more the CDRs are generated at times that are compatible with each other (with regard to the kind of place being analyzed) the more their w's values should be close, and so their temporal distance will be short. For example, in the process of discovering the home place, two CDRs generated on Monday at 5am and on Thursday at 11pm should be close in time, since both the two times are compatible with the home place. Viceversa, CDRs produced on Monday at 5am and on Monday at 11am would present very different values, as it is rather unlikely of being at home on Monday at 11am.

Finally, the actual distance function to be used in the clustering process is a linear combination between the two: $d(cdr_i, cdr_j) = sd(cdr_i, cdr_j) + k * td(cdr_i, cdr_j)$. Where k is a parameter to balance the influence of the spatial distance over the temporal distance.

Clustering Algorithm. After testing a number of algorithms, we decided to adopt a simple agglomerative algorithm [18]. The motivations at the basis of this choice are the following: *(i)* all the algorithms (e.g., K-Means and Spectral Clustering [18]) requiring as input the number of clusters to be found are rather inadequate. This is because the CDRs can be spread over a large area with a very large and unpredictable number of clusters. *(ii)* Density-/competitive- based algorithms (e.g., DBScan, SOM [18]) solve the issue, but do not easily constrain the size of the clusters to be identified. For example, in an early stage of this work, we adopted DBScan, but we had to rely on a rather complex mechanism to constrain the size of the clusters. Constraining the size of the clusters is important in the context CDR data. As the number of cells that can be associated with a certain position is limited to the cells "around" that position, the resulting cluster will have necessarily a limited scope. Since the basic idea of agglomerative clustering algorithms is to cluster the elements that fall within a given radius from a given place, this algorithm is a good starting point.

In a first step we sorted the all CDRs by the their weight (according to the weight function of the kind of place under analysis) with heaviest CDRs first. Then, the algorithm starts with the first CDR in the sorted list and makes this CDR the centroid of the first cluster. Then, for each subsequent CDR, it checks to see whether the CDR falls within a threshold radius of the centroid of any existing cluster. If it does fall within the threshold radius of an existing cluster, and if the spatial extend of the cluster plus the new CDR is still within the threshold, the algorithm adds the CDR to the cluster and moves the centroid of the cluster to be the weighed average of the locations of all the CDRs in the cluster. If it does not, the CDR becomes the centroid of a new cluster. The algorithm completes once every CDRs has been assigned to a cluster (see Algorithm 1).

In the context of the agglomerative clustering algorithm, our spatio-temporal distance function is applied as follows:

1. The spatial distance function sd is given by considering the geographic distance between the centroid of the cluster c and the location l_j associated with the new CDR cdr_j, corrected by the average radius of the cells in the cluster r and the radius of the cell associated with the new CDR r_j. $sd(c, cdr_j) = max(0, geo(c, l_j) - (r + r_j))$
2. The temporal distance considers the average weight $w(t)$ of the CDRs in the cluster and the weight of the new CDR cdr_j produced at time t_j $= w(t_j)$. It is worth noting that the idea of using the function $w()$ in the temporal distance function is fundamental to average correctly the CDRs' times in the cluster. $td(c, cdr_j) = |w(t) - w(t_j)|$
3. In out implementation, given k the parameter to balance the influence of the spatial distance over the temporal distance, we set also k as the threshold of the clustering algorithm: a CDR is in the cluster if: $d(c, cdr_j) = sd(c, cdr_j) + k * td(c, cdr_j) < k$

4.4. Weighing

The identified clusters reflect places visited by the user and from where (s)he generates a number of CDRs. The next step of our approach is to

```
Data: $cdr[\,]$, $k$, $w()$
Result: $c[\,]$
$c[\,] = \emptyset$;
sort $cdr[\,]$ by $w()$;
forall the $cdr_i \in cdr[\,]$ do
    $assigned = false$;
    forall the $c \in c[\,]$ do
        $d = d(c, cdr_i) = sd(c, cdr_i) + k * td(c, cdr_i)$;
        if $d < k$ then
            add $cdr_i$ to c;
            $assigned = true$;
        end
    end
    if $not\ assigned$ then
        create new cluster $c$;
        add $cdr_i$ to c;
        add $c$ to $c[\,]$;
    end
end
```

Algorithm 1: Clustering

assign a weight to clusters so as to reflect their importance. We definded three weighing mechanisms: *(i)* Weight on Time, *(ii)* Weight on Days, *(iii)* Weight on Diversity. The final weight of a cluster C is a linear combination of the 3 weights: $W(C) = W_{time} + \alpha \cdot W_{days} + \beta \cdot W_{diversity}$

α and β are parameters to balance the contribution of the different aspects to the final weight. In our experiments we empirically set $\alpha = \beta = 0.25$ (see analysis in the next Section).

Weight on Time (W_{time}). This weight mechanism gives a weight to each cluster according to the time in which the cluster's CDRs have been generated. More in detail, as a first step multiple CDRs generated on a given day on the same hour are removed (only 1 CDR per day per hour is retained – this is important to avoid that peculiar situations where the user might call a lot biasing the results). Then, we considered the weight function $w()$ associated to the place under analysis and computed W_{time} of the cluster C containing n CDRs as: $W_{time}(C) = \sum_{i=1}^{n} w(cdr_i)$

Weight on Days (W_{days}). This weight mechanism simply counts the number of days in which the cluster has CDRs. This will favor clusters that are regularly visited over the whole period rather than cluster that are visited only during few days (even if those visits happen at highly compatible hours).

Weight on Diversity ($W_{diversity}$). This weight mechanism gives a weight to each cluster according to the distribution of days in which the cluster is frequented. In particular, we create a random variable associating to each day of the week the fraction of CDRs happening on that day (i.e., probability of having that day in the cluster). Then we compute the information entropy of that variable: $H(X) = -\sum_{i \in \{days\ of\ week\}} p(x_i) log_2 p(x_i)$

It is rather easy to see that the information entropy is a measure of how evenly the cluster is visited across the week, thus the more the entropy the more likely the cluster resembles a place regularly visited in all the days. The weight associated to a cluster C having CDRs in n days is given by $W_{diversity}(C) = H(X) * n$

We multiply the entropy by the number of days to make this weight comparable with the other ones. Algorithm 2 illustrates the weighing phase.

```
Data: $c[\,]$, $w()$, $\alpha$; $\beta$
Result: $W[\,]$
forall the $c \in c[\,]$ do
    $cdr[\,]$ = events in $c$, only one event per hour;
    $W_{time} = \sum_{i=1}^{n} w(cdr_i)$;
    $n$ = num days in $c$;
    $W_{days} = n$;
    forall the $d$ $in$ $\{mon,tue,wed,thu,fri,sat,sun\}$ do
        $p[d] = \frac{num\ d\ in\ c}{n}$;
    end
    $W_{diversity} = -n * \sum_{d} p(d) log_2 p(d)$;
    add $W_{time} + \alpha * W_{days} + \beta * W_{diversity}$ to $W[\,]$;
end
```

Algorithm 2: Weighing

4.5. Thresholding

Once all the clusters are given a weight, we have to set a threshold to identify those clusters that represent a place that is actually regularly frequented. In virtually all the approaches at the state of the art, this step is performed by just taking the cluster with the maximum weight. Although this simple approach often works well for home and work places (in that a typical user has just 1 home and 1 work place), it cannot cope with other kind of places. A notable improvement in this direction is described in [8] in which a logistic regression based on clusters' weights is used to classify relevant from non-relevant clusters. However, from our perspective this approach has two main drawbacks: on the one hand, since groundtruth data about users' places is scarce – due to privacy limitations – it is hard to properly train a classifier. On the other hand, since users' behaviors have much variance (e.g., some users produce just few CDRs, while others produce a lot), it is difficult to derive classification thresholds fitting all the cases.

In some other works (and also in an early stage of our work), this thresholding step is performed by running an outlier detection algorithm (based on Z-scores) to identify significant clusters. However, this approach is unsatisfactory for two main reasons: *(i)* The approach selects a cluster to be a relevant place because of its relative weight compared with other clusters.

Instead, the fact that a cluster weighs enough to be relevant it is something that should be independent of the weight of the other clusters. *(ii)* It is rather difficult to parameterize the algorithm so as to specify how often a place should be visited to be considered as relevant.

To solve the above issues, we adopted a different approach to set the threshold value. Our approach is based on estimating what would be the weight of the hypothetical cluster (hyp) that would result if all the CDRs generated by the user in the time window compatible with the place under investigation would be generated from the *same* place.

For example, let us assume that the algorithm is used to detect places visited at night. The weight function for night places sets a positive weight to CDRs generated in the time window 19:00 - 24:00 for all the days of the week. We consider all the CDRs generated by the user in that time window and hypothetically assume they are all generated from the same place. We then apply the weighing mechanism described above and give a weight to the resulting place (i.e., cluster). This weight w_{ref} is an upper bound for the weight of night places in that it represents the extreme case in which the user spends *all* his/her time that single place. It is worth noting that w_{ref} is highly user dependent. Users generating lots of CDRs will have high w_{ref}. Viceversa, users generating few CDRs will have low w_{ref}. Thus w_{ref} well adapts to different users' behaviors.

To convert w_{ref} in our threshold we apply the following procedure.

As a first step we try to understand if there is enough data to identify places at all. If the w_{ref} is too low, it means that just few CDRs are generated in the time window of the place under investigation and so it is impossible to identify regularly visited places.

Calling $days_{ref}$ the number of days in which we have CDRs within the time window of the place under investigation, and $maxdays$ the number of days compatible with the place under investigation in the whole observation period, $days_{ref}/maxdays$ is the fraction of relevant days in which the user generated CDR. For example, considering a dataset of 1 month (30 days) of data. If we are looking for home CDRs then $maxdays = 30$, as in the weight function all the days are compatible with the home place. If we are looking for Sunday CDRs then $maxdays = 4$, as in the weight function only 1 day of the week is compatible with Sunday's places. Supposing that the user does not generate CDRs during the weekend, then we will have $days_{ref}/maxdays = 26/30$ for the home place, $days_{ref}/maxdays = 0/4$ for Sunday place. if $days_{ref}/maxdays < \gamma$ we abort place identification as there

```
Data: $cdr[\,], c[\,], W[\,], \gamma, \delta, f, max\_days$
Result: $p[\,]$
hyp;
forall the $cdr_i$ in $cdr[\,]$ do
    if $w(cdr_i) > 0$ then
        add $cdr_i$ to $hyp$;
    end
end
$W_{ref}$ = Weighing(hc);
$days_{ref}$ = num days in $hc$;
if $(days_{ref}/max\_days) < \gamma$ then
    return $\emptyset$;
end
$th = \delta * f * W_{ref}$;
$p[\,]$;
forall the $c_i$ in $c[\,]$ do
    if $W_i > th$ then
        add barycenter($c_i$) to $p$;
    end
end
```

Algorithm 3: Thresholding

is not enough data. In our experiments we use $\gamma = 0.2$.

If the above condition does not apply, we are in the case in which the algorithm can actually proceed trying to identify relevant places. To explain how the procedure works, let us focus on an example. Let us assume that the algorithm is trying to identify the user's night places. Of course a place can be relevant even if its weight is well below w_{ref} in that it is not required nor realistic that the user spends all his/her time in that place. If w_{ref} is the weight of the user being in the palace for the whole time, $w_{ref}/7$ can be roughly associated to the weight of the user being in the palace once a week.

Accordingly, a reasonable threshold for night-places' clusters could be $th = 1/7 * w_{ref}$ if we want to identify places that are visited once a week.

Similarly, a reasonable threshold for places visited on Sunday could be $th = 1/4 * w_{ref}$ if we want to identify places that are roughly visited once a month. We call f the fraction of time required to mark a places as relevant (e.g., 1/7 or 1/4 as in the example above). Finally, we added a smoothing factor δ that provides further customization, the final threshold is given by: $th = \delta * f * w_{ref}$. f is set on the basis of the place we are looking for. δ is set via optimization (i.e., via a training mechanism). In most of our experiments we used $\delta = 0.5$ (see analysis in the next Section).

All the clusters with a weight greater than the threshold th are selected as relevant. Once a cluster is selected, we set the corresponding place to the average of the coordinates of the clusters' CDRs (i.e., its barycenter).

This final step concludes our approach: a number of places p representing the users' habits are retrieved. Algorithm 3 describe this phase.

5. Experiments

In this section we report experiments conducted to test the behavior of the proposed methodology. The main part of the experiments tests home and work place identification as the available groundtruth information allows sound evaluation. Then some experiments on the recognition of other places is reported. To comply with privacy limitations described in Section 2, it is generally not possible to retrieve the actual home/work locations of individuals (e.g., from billing contracts) to be used as groundtruth. Accordingly, most of the experiments have been performed over a set of 11 volunteer users, who explicitly consent for the experiment. However, experiments in Section 5.2 involve a larger population sample of 10000 individuals whose aggregated inferred places (for whom explicitly consent is not required) were compared with census information.

5.1. Home and Work Places

To evaluate obtained results with regard to home and work places, we perform the following procedure. For each volunteer user, we collect the number of places that our algorithm produces as result. Then we select the place (if any) that is closest to the groundtruth. If the distance between the place and the groundtruth is below a certain threshold T, we have found the place correctly and score 1 True Positive (TP) result. If there are no results, or if all the places are farther away than T from the ground truth, we missed the place and score 1 False Negative (FN) result. All the identified places that are not the true positive are False Positives (FP): places that the algorithm selects as relevant but they are not.

To take into account the fact that distances in the city center are more "significant" than in the suburbs (in the city centers cell radii are smaller, so the algorithm is expected to identify the place more precisely), the threshold T is adjusted to the average radius of cells in the area. In our experiments we considered T = k + *average radius*. Where k is the parameter used in the clustering mechanism. The idea is that k is the radius of the cluster. setting T lower that k makes little sense as the clustering will create clusters already larger that the threshold.

We evaluate our algorithm in terms of *recall*, *precision* and the average distance between the result and groundtruth.

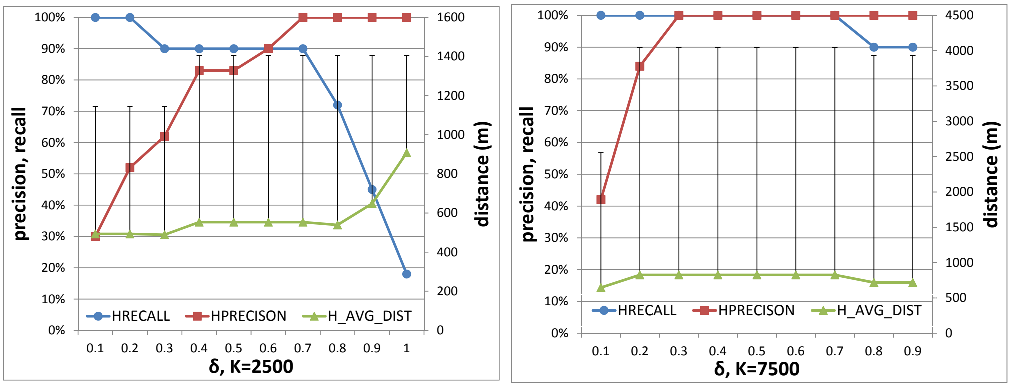

Figure 4: Results obtained for different values of k and δ using leave-one-out data as input. Identification of the home place. We report precision, recall and the average distance at which location can be identified.

5.1.1. Results Varying Parameters

The proposed approach is based on a set of parameters to tune the behavior of the algorithm. In a first set of experiments we assess the algorithm's performance on the basis of the different parameters' values.

Clustering and Thresholding Parameters. Apart from the weight function itself, the two most important parameters of our algorithm are the k parameter of the clustering and the δ parameter used for thresholding.

A large value of k creates large clusters. In general, a large value of k tends to improve the recall of the algorithm. Large clusters weight more so it is likely that important place will surpass the threshold, identifying the place. For the same reason, however, large value of k tends to create false positives and it reduces the precision of the algorithm: the place is identified, but because the cluster is large, its actual location cannot be pinned down.

The δ parameter is also fundamental in that it basically scales the threshold value. A low δ implies that a lot of clusters are promoted as relevant places. This naturally increases recall, but lowers precision (i.e., the number of false positives rises).

We run the algorithm for different values of k and δ. Figure 4-left shows results for $k = 2500m$ and δ varying from 0.1 to 1. Figure 4-right shows results with $k = 7500m$ and δ varying from 0.1 to 1.

The y-axis on the left is the recall and precision percentage (HRECALL and HPRECISION – H for home, since the graph refers to home place identification). Since *delta* scales the threshold, it is easy to understand that we have high-recall and low-precision for low values of *delta* (low threshold), while we have low-recall and high-precision for high values of *delta* (high threshold). The graph shows also the average distance between real and estimated home (H_AVG_DIST) and it refers to the y-axis on the right expressing the distance in meters. We also report error bars with the maximum experienced distance between real and estimated home. Accordingly, the parameters' choice $k = 2500m$ and $delta = 0.6$ in home place recognition leads to: recall = 91% (10/11 TP), precision = 90% (1 FP), average spatial error = 553m, maximum spatial error = 1405 m. Vice Versa, parameters' choice $k = 7500m$ and $delta = 0.6$ leads to: recall = 100% (11/11 TP), precision = 100% (0 FP), an average spatial error of 902 m, and maximum error of 3184 m. It is possible to see that large values of k tend to improve recall and precision, but also increase the average spatial error. Of course the same kind of analysis could be conducted for all the other kind of places.

In general, the optimal parameter choice depends on the application. If the application requires a high precision with regard to the exact spatial location of the place, then low k values are to be preferred. If the application tolerates spatial errors, then high k values are to be preferred so ad to improve recall and precision. Similarly, if the application can tolerate false positives, then it is best to select a low δ.

Weighing Parameters. In Figure 5 we present experiments showing the impact of coefficients used in the linear combination α and β in the results. It is possible to see that the choice of α and β does not influence significantly the results with regard to home identification, while excessively high values for α and β significantly reduces the precision of work place identification. In general, in all the experiments we adopted $\alpha = 0.25$ and $\beta = 0.25$ as it is the value optimizing home recall, without undetermining work precision.

5.1.2. Results Varying Data Size

In this section we try to measure the performance of the algorithm with different amounts of data available. How much CDR data is needed to obtain accurate results? This is of course fundamental to configure a system based on this algorithm. We run the algorithms with an increasing amount of input data. Figure 6 illustrates the results. We provide results both at a week granularity, and at the month granularity. The results are rather interesting. While work recall and home precision remain high also for a small dataset, work precision and home recall are notably reduced. With a 1-week dataset, we obtain around 50% for both home recall and work precision.

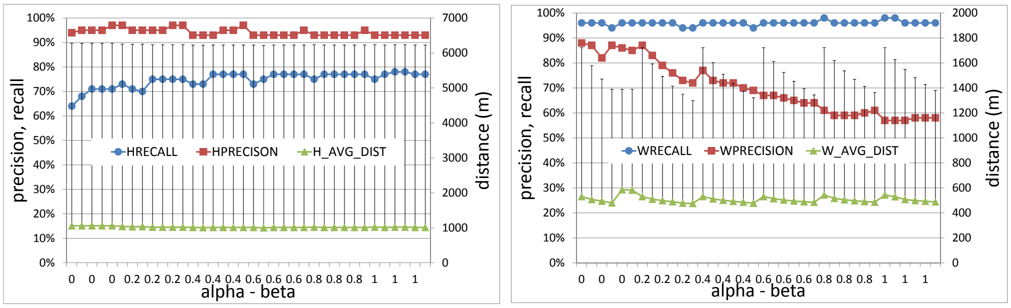

Figure 5: Results obtained for different values of α and β. *(left)* Identification of the home place. *(right)* Identification of the work place.

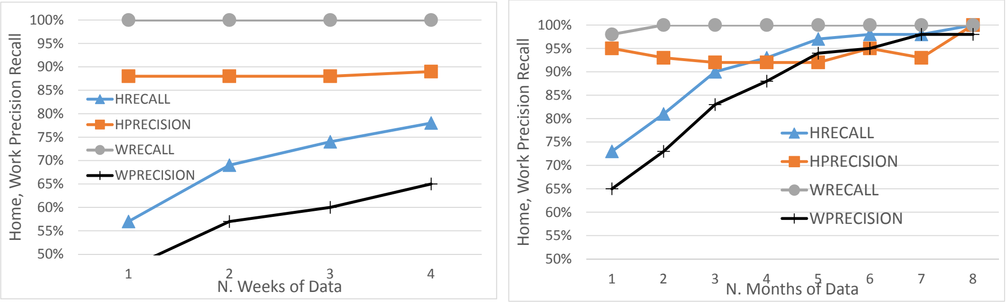

Figure 6: Recall and precision results for an increasing amount of input data. *(left)* Results varying the number of *weeks* provided to the algorithm. *(right)* Results varying the number of *months* provided to the algorithm

The reasons at the basis of these results are the following: home place identification deals with the scarce number of CDRs produced at night. With just few weeks of data, it might happen that there are too few CDRs to identify the home place, hence low recall results. Vice versa, as the number of CDRs is low, it is very difficult to create false positives, thus the precision remains high. Work place identification deals with selecting the work location out of the many CDRs produced during the day. As there are a lot of CDRs produced also in a single week, work recall tends to remain high. However, the approach also produces false positives (threshold values for a single week is rather low) and thus the precision is low. Table (see Figure 7) summarizes obtained results. The top two rows use all the data for the analysis. The bottom two rows use only data from one month.

Dataset	k	δ	α	β	H Recall	H Precision	H Avg Dist	W Recall	W Precision	W Avg Dist
all	7500m	0.7	0.25	0.25	100%	100%	826m	100%	80%	677m
all	2500m	0.6	0.25	0.25	90%	90%	553m	100%	80%	396m
1 month	6000m	0.5	0.25	0.25	80%	93%	1020m	96%	70%	507m
1 month	2500m	0.5	0.25	0.25	59%	79%	774m	96%	76%	404m

Figure 7: Table summarizing main results obtained. We report precision, recall and the average distance at which location can be identified.

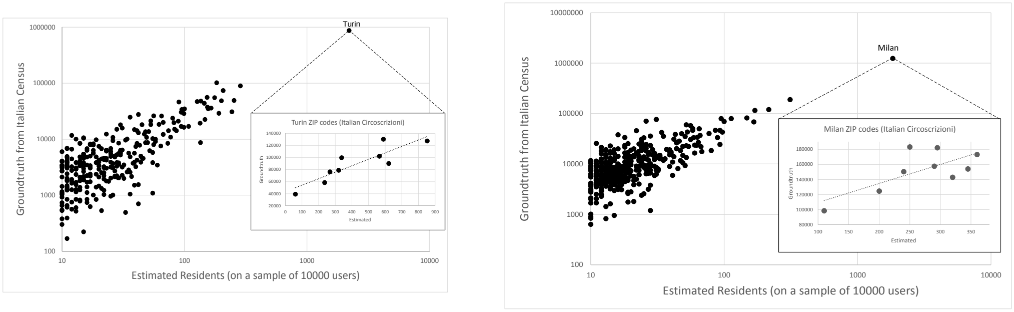

Figure 8: Correlation between the number of people living in a city as measured by our approach and by the Italian census. *(left)* Piemonte, *(right)* Lombardia. Plot details for the city of Turin and Milan.

5.2. Census Comparison

In another set of experiments we apply our algorithm to a larger sample of the users "living" in the two regions under analysis. For each region, we considered a random sample of 10000 individuals and run our place identification algorithm for each of them. We count the number of people living in each city and compare this number with data coming from the Italian census.

Figure 8 illustrates our results that well conform to the linear correlation. The graph is in a log-log scale to compress the variation of sizes among cities. Figure 8(left) illustrates results for Piemonte. Computed correlation coefficient is 0.91. We also conduced the same analysis in the city of Turin, considering regions associate to Italian *circoscrizioni* (similar to ZIP codes). Results provide correlation coefficient 0.88. Figure 8(right) illustrates results for Lombardia. Computed correlation coefficient 0.90. We also conduced the same analysis in the city of Milan, obtaining correlation coefficient 0.72.

5.3. Other kind of Places

In this set of experiments, we try to assess the results of our algorithm with regard to places other than home and work. The main issue is that we do not have accurate groundtruth information for these other kinds of places. Volunteer users also reported visited places other than home and work, but such reports are rather anecdotal and it is difficult to extract sound statistics.

To overcome this limitation we adopt a cross-validation approach. We consider data coming from a given time period (e.g., a month) and extract the resulting places. Then we consider data coming from another time period (e.g., another month) and test if the extracted places remain the same or change. The more the places remain the same, the more the algorithm provides consistent results cross-validating them. In the experiment we cross validated results obtained by running the algorithm on individual months of data. Results over a given month serve as groundtruth information to evaluate results over a different month in terms of recall and precision.

Figure 9 illustrates the results. It is possible to see that our algorithm produces consistent results for different categories. Although it is true that the algorithm could be consistent also in reporting false positives over consecutive months, we think this is rather unlikely as the identified places are visited many times in all the months. On the contrary, we speculate that some inconsistencies are instead actually due to new places discovered by the users that were not visited before. To further validate this consistency results, we compared the extracted places with the anecdotal information provided by volunteer users obtaining a good correspondence in several cases.

Kind of Place	**Precision**	**Recall**
Friday Night	88%	93%
Sunday	73%	100%
Night	87%	53%
Saturday Night	93%	100%
Overall Avg.	85%	73%

Figure 9: Recall and precision results obtained by cross validating the places extracted by the algorithm over individual moths for different kind of places.

6. Concluding Remarks

In this work we propose an innovative method to identify frequented places from cellular network data. Our approach improves the state of the art in many directions and provides good identification accuracy.

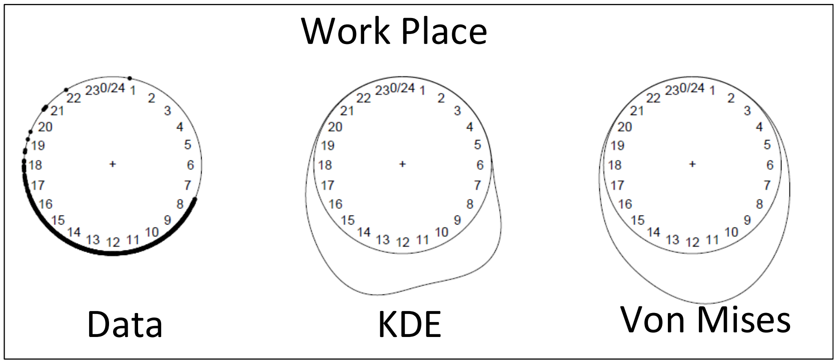

Figure 10: Examples of matrices learnt via maximum likelihood. *(left)* data points. *(center)* KDE approach. *(right)* Von Mises approach.

A partial limitation in the evaluation of our algorithm is the small sample size of users providing groundtruth information. A more extensive evaluation of data with groundtruth information would be much valuable. A possible alternative, in the lack of further groundtruth information, would be to run a consistency experiment, following the approach presented in Section 5.3.

A. Weight on Time Function

The time-based weight function $w()$ described in Section 4.2 can be either set by a domain expert or automatically learnt via training examples. In the latter case, we tested two approaches ultimately producing similar results. On the one hand, we just modeled the temporal visits by using a Kernel Density Estimation (KDE) algorithm to smooth-out irregularities due to sample biases and create the distribution. On the other hand, we modeled the temporal visits by using a Von Mises distribution (i.e., circular Gaussian) associated to the days of the week under analysis. More in detail, we considered a set of users for which groundtruth information (e.g., reported relevant places) were known. For each kind of places we considered the times in which users produced CDRs in the proximity of the reported places (we considered a fixed threshold of 1Km). The list of times is used to learn the distributions either by placing a small Gaussian kernel on each point (KDE), or by fitting the Von Mises via maximum likelihood. Figure 10 reports some results obtained with this approach. The figure illustrates: **(left)** timing data (i.e., the times of day in which users were at the place under analysis) for the “training” users, **(center)** the density functions obtained using the KDE approach and **(right)** the density function obtained using the Von Mises approach. It is easy to see that the Von Mises smooths the data further by creating a unimodal distribution. The resulting functions (that can vary for different days of the week) are used to weigh CDRs.

References

[1] F. Calabrese, G. D. Lorenzo, L. Liu, C. Ratti, Estimating origin-destination flows using mobile phone location data, IEEE Pervasive Computing 10 (4) (2011) 36–44.

[2] L. Ferrari, M. Mamei, M. Colonna, Discovering events in the city via mobile network analysis, Journal of Ambient Intelligence and Humanized Computing 5 (3) (2014) 265–277.

[3] M. Iqbal, C. Choudhury, P. Wang, M. Gonzalez, Development of origin-destination matrices using mobile phone call data, Transportation Research Part C: Emerging Technologies 40, 63-74.

[4] D. Quercia, G. D. Lorenzo, F. Calabrese, C. Ratti, Mobile phones and outdoor advertising: Measurable advertising, IEEE Pervasive Computing 10 (2) (2011) 28–36.

[5] A. Rein, S. Siiri, J. Olle, S. Erki, T. Margus, Using mobile positioning data to model locations meaningful to users of mobile phones, Journal of Urban Technology 17 (1) (2010) 3–27.

[6] Z. Duan, L. Liu, S. Wang, Mobilepulse: Dynamic profiling of land use pattern and od matrix estimation from 10 million individual cell phone records in shanghai, in: International Conference on Geoinformatics, Shanghai, China, 2011.

[7] V. Frias-Martinez, J. Virseda, A. Rubio, E. Frias-Martinez, Towards large scale technology impact analyses: Automatic residential localization from mobile phone-call data, in: International ICTD Conference, London, UK, 2010.

[8] S. Isaacman, R. Becker, R. Caceres, S. Kobourov, M. Martonosi, J. Rowland, A. Varshavsky, Identifying important places in peoples lives from

cellular network data, in: International Conference on Pervasive Computing, San Francisco (CA), USA, 2011.

[9] F. Zambonelli, Toward sociotechnical urban superorganisms, IEEE Computer 45 (8) (2008) 76–78.

[10] L. Bengtsson, X. Lu, A. Thorson, R. Garfield, J. von Schreeb, Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: A post-earthquake geospatial study in haiti, PLoS Med, 8(8).

[11] S. Bekhor, Y. Cohen, C. Solomon, Evaluating long-distance travel patterns in israel by tracking cellular phone positions, Journal of Advanced Transportation February (2011) 11–24.

[12] J. Kang, W. Welbourne, B. Stewart, G. Borriello, Extracting places from traces of locations, in: Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, Philadelphia (PA), USA, 2004.

[13] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from gps trajectories for mobile users, in: International World Wide Web Conference, Madrid, Spain, 2009.

[14] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, J. Hughes, Learning and recognizing the places we go, in: International Conference on Ubiquitous Computing, Tokyo, Japan, 2005.

[15] D. Kim, J. Hightower, R. Govindan, D. Estrin, Discovering semantically meaningful places from pervasive rf-beacons, in: International Conference on Ubiquitous Computing, Orlando (FL) ,USA, 2009.

[16] T. Sohn, A. Varshavsky, A. LaMarca, M. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. Griswold, E. de Lara, Mobility detection using everyday gsm traces, in: International Conference on Ubiquitous Computing, Orange County (CA), USA, 2006.

[17] M. Ulm, P. Widhalm, Properties of the positioning error of cell phone trajectories, in: NetMob, Boston (MA), USA, 2013.

[18] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, Springer, 2006, pp. 25–71.