2012 IEEE 13th International Conference on Mobile Data Management

A Demonstration of Mining Significant Places from Cell ID Trajectories through A Geo-grid based Approach

Tengfei Bao[1] Huanhuan Cao[2] Qiang Yang[3] Enhong Chen[1] Jilei Tian[2]
[1]University of Science and Technology of China [2]Nokia Research Center
[3]Hong Kong University of Science and Technology
[1]{tfbao92, cheneh}@ustc.edu.cn [2]{happia.cao, jilei.tian}@nokia.com [3]qyang@cse.ust.hk

***Abstract*—Mining the frequently visited places of single mobile users, i.e., *significant places*, is crucial for supporting personalized location-based services. Most of existing works for significance place mining have a need to take advantage the GPS trajectories of users. However, it is difficult to encourage mobile users to contribute GPS trajectories because of the high power consumption of GPS. In this demonstration, we propose a geo-grid based approach for mining significant places from cell ID trajectories. In our approach, the mined significant places are represented as sets of geo-grids which are much smaller than the coverage areas of cell-sites. To be specific, we firstly extract the *stay areas* where the mobile user used to stay and map them to many geo-grids. Then we mine significant places from the geo-grids by considering their significance.**

I. Introduction

In recent years, location-based services such as Google Latitude (www.google.com/latitude) and Foursquare (foursquare.com) have been more and more popular with the rapid popularization of smart mobile devices. A type of interesting and promising location-based services is to provide personalized location-based services by not only considering the current locations of users but also the places which they frequently visit, i.e., significant places [1]. For example, the nearby deals can be recommended to mobile users by considering their significant places.

The GPS trajectories and cell ID trajectories of mobile users are two major source for mining significant places. Although several prior works have been done for mining significant places from the GPS trajectories of mobile users (e.g., [2], [6]), it is usually difficult to encourage mobile users to contribute GPS trajectories because the continuous GPS sensing is very power-consuming [3] and thus will dramatically hurt user experience. In contrast, cell ID trajectories are logs in mobile devices which record the IDs of serving cell-sites with a predefined time interval and are much easier to be collected since the power consumption of recording the IDs of serving cell-sites is trivial. Moreover, the approaches for mining significant places from cell ID trajectories can be applied to low end mobile devices without GPS sensors and thus make it possible to run significant place based services for a larger user base. However, although several prior works (e.g., [5], [4]) propose some approaches for mining significant places from cell ID trajectories by clustering cell IDs, the accuracy of the results are not acceptable for many practical applications because several cell-sites usually cover a too large area.

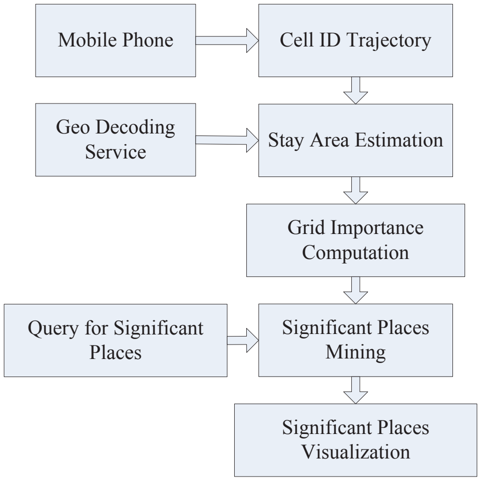
Mobile Phone
Cell ID Trajectory
Geo Decoding Service
Stay Area Estimation
Grid Importance Computation
Query for Significant Places
Significant Places Mining
Significant Places Visualization

Fig. 1. The Demonstration System Architecture.

To this end, in this demonstration we propose to mine the areas which consist of several geo-grids from cell ID trajectories as significant places, where a geo-grid is an area divided by particular longitudes and latitudes and usually much smaller than the coverage area of a cell-site. Figure 1 depicts the system architecture. Given the mobile phone generated cell ID data, our approach has two stages to mine the significant places as follows. In the first stage, we extract the stay areas where users used to stay from cell ID trajectories by leveraging the coverage areas of cell-sites and map the stay areas into geo-grids in a proper scale. Each geo-grid in the extracted stay areas are candidate significant places. In the second stage, we firstly calculate the significance of each geo-grid and then use a recursively pruning algorithm to separate the areas which consist of many geo-grids by removing the geo-grids with low significance. Finally, the maintained areas which are smaller than a predefined maximum area are taken as significant places and we show them on the web map interface. In this way, we can obtain significant places from cell ID trajectories with

978-0-7695-4713-8/12 $26.00 © 2012 IEEE
DOI 10.1109/MDM.2012.70

IEEE computer society

lower power consumption and much higher accuracy than the state-of-the-art works.

II. Extracting Stay Areas

Ideally, we can firstly detect the places where a given user used to stay but not only pass by as *stay points* and then mine significant places from them. However, the real stay points in the form of geographical coordinates cannot be directly inferred from cell ID trajectories because we can only roughly estimate the areas which the user used to visit through the coverage areas of recorded cell-sites, which can be estimated from the locations and serving radiuses of cell-sites provided by some public Web Services such as Google Geocoding API (http://code.google.com/apis/maps/documentation/geocoding/). To this end, we firstly try to find the stay areas where the user used to stay and then mine significant places from the discovered stay areas. Obviously, stay areas are estimations of stay points. The smaller the stay areas, the more precise they are for estimating real stay points. In this section, we present the details of our approach for discovering the stay areas of mobile users from their cell ID trajectories.

A. Stay Session Discovery

To extract stay areas, we firstly find the segments of cell IDs whose coverage areas may contain a stay point from the cell ID trajectory of a mobile user, which are referred as *stay sessions* for simplicity, and then take the overlapped coverage area of all cell-sites in a stay session as a stay area. The method of discovering stay sessions is motivated from the observation as follows.

Observation: if we take no account of the errors for the estimated coverage areas of cell-sites, we will have the following observation: suppose a user has stayed in a location for a while, the corresponding cell ID trajectory may consist of a) several duplicate occurrences of the same cell ID, or b) several different cell IDs whose coverage areas are mutually overlapped with each other. The first case is easy to understand. The second case usually occurs when the user is staying in the overlapped area of the coverage areas of several adjacent cell-sites. In such an area, the serving cell-site of the mobile user may be any of the group of adjacent cell-sites according to their signal quality. Consequently, the recorded cell IDs of serving cell-sites may change even though the user is not moving. Figure 2 shows an example of the second case that a group of cell IDs whose coverage areas mutually are overlapped implies a stay point. In the example the sampling rate of cell-sites in service is one minute. From this figure, we can see that when the user stays in the point P_1 for several minutes, the coverage areas of the corresponding cell IDs $\{c_1, c_2, c_3\}$ are mutually overlapped. When the user moves from point P_1 to point P_2, the coverage areas of the sampled cell IDs are not overlapped with all cell IDs in $\{c_1, c_2, c_3\}$. When the user arrives in point P_2 and stay for a while, the coverage areas of the recorded cell IDs $\{c_7, c_8, c_9\}$ are mutually overlapped again, which clearly implies the user is in a stay point.

Algorithm 1 Stay Session Detection

Input 1: a cell ID trajectory $C = c_1c_2...c_n$;
Input 2: a minimum staying time T_{min};
Output: a set of stay sessions S;

```
$S \leftarrow \emptyset$;
$s \leftarrow \{c_1\}$;
for $i = 1; i < n; i++$ do
    if $c_i \neq c_{i+1}$ then
        $Movment \leftarrow false$;
        for eachc $\in s$ do
            if $Distance(c, c_{i+1}) \geq c.Radius + c_{i+1}.Radius$ then
                $Movment \leftarrow true$;
                if $TimeRange(s) \geq T_{min}$ then
                    $S \leftarrow S \cup s$;
                $s \leftarrow \{c_{i+1}\}$;
        if $Movment = false$ then
            $s \leftarrow s + c_{i+1}$;//append $c_{i+1}$ to the tail of $s$
return $S$;
```

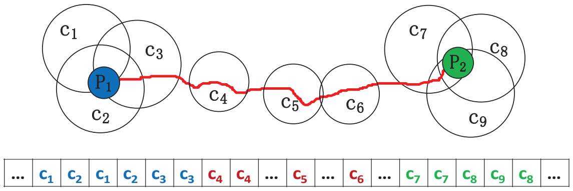

Fig. 2. An example of cell ID trajectory which implies that the user moves from a stay point to another stay point. Each circle denotes the coverage area of a corresponding cell-site.

Based on the above observation, we can easily detect the segments of cell IDs whose coverage areas may contain a stay point from the cell ID trajectories of mobile users, which are referred as *stay sessions* for simplicity. The notion of stay sessions are formally defined as follows.

Definition 1 (Closed Cell ID Segment): Given a cell ID trajectory $C = c_1c_2...c_n$, where $c_i(1 \leq i \leq n)$ denotes a cell ID, for a segment of C denoted as $s = c_jc_{j+1}...c_{j+k}(1 \leq j \leq n-k)$, s is called a closed cell ID segment of C iff $\forall_{j \leq a, b \leq (j+k)} c_a.A \cap c_b.A \neq \phi$, where $c.A$ denotes the coverage area of the cell-site with ID c.

Definition 2 (Stay Session): Given a predefined threshold of minimum time range T_{min}, for a closed cell segment $s = c_ic_{i+1}...c_{i+n}$, s is called a stay session iif (a) $(c_{i+n}.timestamp - c_{i+1}.timestamp) \geq T_{min}$ and (b) / $\exists_{s'}(s \subset s') \wedge (s'$ is a closed segment of $C)$.

According to notions we can detect stay sessions by scanning the cell ID trajectory and iteratively discover the closed cell-ID sequences and check whether they are stay sessions. Algorithm 1 illustrates the method of stay session extraction.

Herein the variable $Movement$ is used to record the recognition of a closed cell ID sequence and $c.Radius$ indicates the coverage radius of the cell-site c. The parameter of T_{min} is set to 30 minutes in our experiments. Moreover, the distance between the cell-site with ID c_i and another one with ID c_j is calculated as follows.

$$Distance(c_i, c_j) = R \times \arcsin \sqrt{\sin^2 (\frac{\Delta_{Lat}}{2}) + \cos (c_A.Lat) \cos (c_B.Lat) \sin^2 (\frac{\Delta_{Long}}{2})},$$

where $c.Lat$ and $c.Long$ indicates the latitude and longitude of the cell-site c respectively, R denotes the radius of equator[1], $\Delta_{Lat} = |c_A.Lat - c_B.Lat|$ and $\Delta_{Long} = |c_A.Long - c_B.Long|$.

B. Estimating Stay Areas by Geo-grids

Given a stay session $s = c_i c_{i+1}...c_{i+n}$, we can estimate the stay area of the user by $A_s = \bigcap_{c \in s} c.A$, where $c.A$ indicates the coverage area of the cell-site c. A stay area A_s indicates that the user's movement is limited in the area during the according time range, which implies it may contain a stay point of the user. The longer the time range and the smaller the stay area, the more likely the user is in a stay status and the stay point is covered by A_s.

Since the coverage areas of cell-sites are usually represented by areas of circles, stay areas are essentially irregular areas bounded by curves. However, it is inefficient to represent a stay area by a group of boundaries in the form of sphere curves. Moreover, too accurate estimations of stay areas are meaningless because the information of the coverage areas for cell-sites usually contain errors. Therefore, we use a simple and efficient geo-grid based method to estimate the stay area. The basic idea of the approach is as follows. Firstly, we partition the surface of the earth into many geo-grids by latitude and longitude. Then we can use a group of geo-grids to represent the coverage area of a cell-site c by enumerating the geo-grids whose centers are covered by $c.A$. Finally, we can quickly calculate the overlapped area among the coverage areas of several cell-sites by enumerating the joining geo-grids among their covered geo-grids as shown in Figure 3. Obviously, the smaller scale we use to partition the earth, the more accurate the estimation can be. But as mentioned above, we do not need too accurate estimations because of the inherent errors of the cell-site information. In practice, we partition the surface of the earth in the scale of 0.001 latitude × 0.001 longitude.

III. Mining Significant Place from Stay Areas

With the stay areas of a user, we can mine his (or her) significant places. Intuitively, we can count the visiting frequency of each geo-grid in the stay areas and take the top frequently visited geo-grids as significant places. To be specific, we can count a geo-grid to be visited once when it appears in one stay area. However, this naive approach does not take into account the different accuracy of estimating stay points for each stay area. Usually, the larger the stay area, the less accurate the estimation of a real stay point. Motivated by this observation, we should take into account the geo-grids occurring in small stay areas more than those occurring in big ones. Moreover, we observe that the longer the time range of a stay session, the more likely it contains a real stay point, which implies that we should pay more attention to the geo-grids occurring in stay areas extracted from long stay sessions. Along this line, for each geo-grid g occurring the extracted stay areas, the significance is calculated as follows.

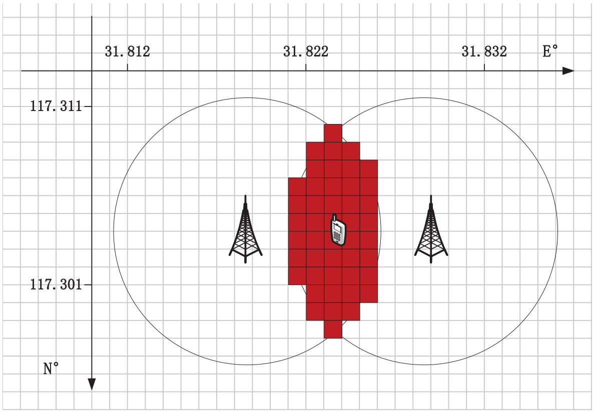

Fig. 3. An example of estimating stay areas by geo-grids. The red area estimates the overlapped area of two cell-sites' coverage areas.

Algorithm 2 Significant Places Extraction

```
Input 1: a set of areas $\Lambda = \{A\}$;
Input 2: a maximum area threshold $A_{max}$;
Output: a set of significant places $P$;
1:  $P \leftarrow \emptyset$;
2:  for each $A \in \Lambda$ do
3:      if $Area(A) > A_{max}$ then
4:          call $Separate(\Lambda, A_{max}, P)$;
5:      else
6:          $P \leftarrow P \cup A$;
7:  return $P$;

Method Separate($\Lambda', A'_{max}, P$)
1:  for each $A' \in \Lambda'$ do
2:      if $Area(A') > A_{max}$ then
3:          $g_{min} \leftarrow \mathrm{argmin}_g(Signicance(g))$, where $g \in A'$;
4:          for each $g \in A'$ do
5:              if $Signicance(g) \leq Signicance(g_{min})$ then
6:                  $A' \leftarrow A' - g$;
7:          if $A'$ is split to several areas $\Lambda^* = \{A^*\}$ then
8:              call $Separate(\Lambda^*, A_{max}, P)$;
9:          else
10:             go to 3;
11:     else
12:         $P \leftarrow P \cup A'$;
13: return;
```

[1] For simplicity, we assume the Earth is a perfect sphere.

$$Significance(g) = \sum_{s:g \in s.A} \frac{TimeRange(s)}{GridNum(s.A)}, \qquad (1)$$

where s denotes a stay session, $s.A$ denotes the corresponding stay area, $GridNum(s.A)$ indicates the number of geo-grids $s.A$ contains.

It is worth noting that a geo-grid with high significance may not correspond to one real significant place. On one hand, when the scale of the geo-grid is relatively big, a significant geo-grid may contain several significant places, which is called *false merging*. On the other hand, when the scale of the geo-grid is relatively small, several adjacent significant geo-grids may imply the same significant place, which is called *false splitting*. For example, the significant place may be a big plaza which covers several geo-grids. Another example of false splitting is that a significant place may be in the common boundary of adjacent geo-grids. For the false merging problem, we cannot split a geo-grid to discover the real significant places. Thus, we should select relatively small geo-grids in practice. For the false splitting problem, we can assume that two adjacent significant geo-grids may imply the same significant place. Based on the intuitive assumption, we propose a geo-grid pruning based algorithm for discovering the areas whose contained geo-grids have high average significance as significant places.

The basic idea of the algorithm is as follows. Initially, all geo-grids appearing in the extracted stay sessions are naturally split to several areas which consist of many geo-grids due to the connectivity among them. Firstly, we define a maximum area threshold as A_{max} to limit the areas of estimated significant places. Then for each area we recursively remove the geo-grids with the lowest significance in the area to split the original area by the connectivity among geo-grids. The pseudo code the algorithm is illustrated in Algorithm 2, where A denotes an area which consists of many geo-grids, the method $Seperate(\Lambda', A'_{max}, P)$ is a recursive function for separating areas in Λ' and inserting the areas which are small enough to the global set of significant places P.

IV. A Preview of the Demonstration

We implement the demonstration with a Python Web framework Django and use the MySQL database to store user Cell ID trajectories and mined significant places. We get the real location and cover areas of cell sites through Google Map service API. And the mined significant places are shown through the Google Map JavaScript API. With this demonstration, users can select their cell ID trajectory which had stored in the database to view the cell ID distribution or to query the significant places. All operations are performed in the web page interface and all the results will be shown in a web browser. Some screen shots of the demonstration are shown in Figure 4 and Figure 5. In Figure 4, the system shows the raw cell ID distribution in a selected cell ID trajectory spanning for one month. In Figure 5, the system shows the corresponding mined significant places.

In future extension, we plan to develop a Web service version of the demonstration which provide APIs to allow users to post their cell ID trajectories to our server and get the mined significant places in a predefined JSON format.

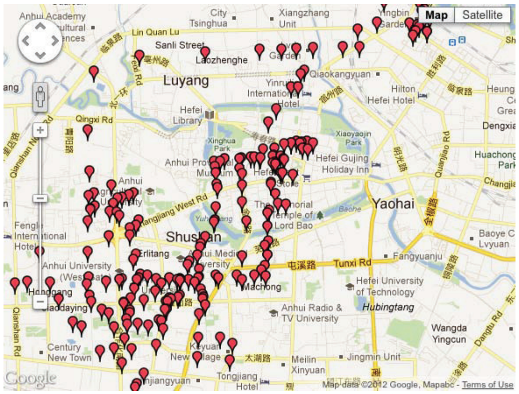

Fig. 4. The raw Cell ID distributions.

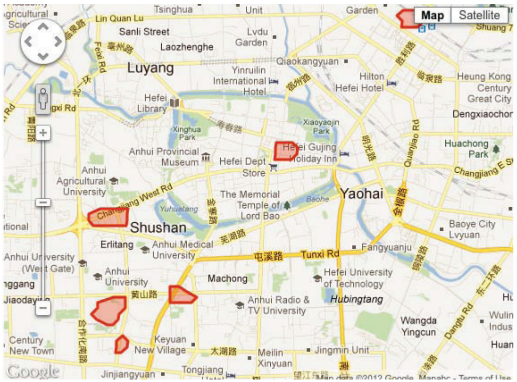

Fig. 5. The mined Significant Places.

V. Acknowledgements

This work was supported by grants from Natural Science Foundation of China (Grant No. 60775037), The National Major Special Science and Technology Projects (Grant No. 2011ZX04016-071), Research Fund for the Doctoral Program of Higher Education of China (20113402110024) and Nokia Research Center China.

References

[1] B. Adams, D. Phung, and S. Venkatesh. Extraction of Social Context and Application to Personal Multimeida Exploration. In *Proceedings of the ACM Conference on Multimedia(MM)*, pages 987–996, 2006.

[2] D. Ashbrook and T. Starner. Learning Significant Locations and Predicting User Movement with GPS. In *Proceedings of 6th IEEE International Symposium on Wearable Computers, Seattle, WA*, 2002.

[3] A. F. Ben, A. Phillips, and T. Henderson. Less is more: energy-efficient mobile sensing with senseless. In *MobiHeld '09: Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pages 61–62, 2009.

[4] N. Eagle, A. Clauset, and J. A. Quinn. Location Segmentation, Inference and Prediction for Anticipatory Computing. In *AAAI Spring Symposium On Technosocial Predictive Analytics, Standford*, 2009.

[5] K. Lassonen, and M. Raento, and H. Toivonen. Adaptive On-Device Location Recognition In *Proceedings of the Second International Conference on Pervasive Computing*, pages 287-304, 2004.

[6] N. Marmasse, and C. Schmandt, Location-Aware Information Delivery with ComMotion. In *Proceedings of HUC 2000, Bristol, England*, pages 157-171, 2000