Mining Correlation between Locations Using Human Location History

Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma
Microsoft Research Asia, 4F, Sigma Building, No.49 Zhichun Road, Haidian District, Beijing 100190, China
{yuzheng, v-lizzha, xingx, wyma}@microsoft.com

ABSTRACT

The advance of location-acquisition technologies enables people to record their location histories with spatio-temporal datasets, which imply the correlation between geographical regions. This correlation indicates the relationship between locations in the space of human behavior, and can enable many valuable services, such as sales promotion and location recommendation. In this paper, by taking into account a user's travel experience and the sequentiality locations have been visited, we propose an approach to mine the correlation between locations from a large number of users' location histories. We conducted a personalized location recommendation system using the location correlation, and evaluated this system with a large-scale real-world GPS dataset. As a result, our method outperforms the related work using the Pearson correlation.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - *data mining*. H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, retrieval model*.

General Terms

Algorithms, Measurement, Experimentation.

Keywords

Location Correlation, Location History, GPS trajectory.

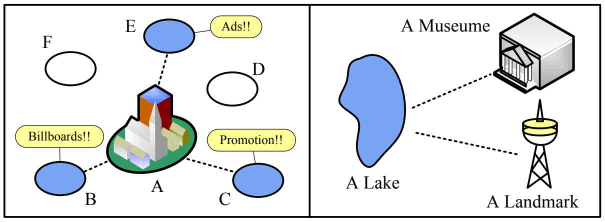
1. INTRODUCTION

The increasing popularity of location-acquisition technologies, such as GPS and GSM network, is leading to the collection of large spatio-temporal dataset of individuals [8][9][10]. The dataset cannot only represent people's location histories but also imply the correlation between geographical regions. Beyond the geo-distance relationship [2][6][7], this correlation denotes the relationship between locations from the perspective of human behavior, and might indicate the probability that two locations co-occurred in people's trips.

Typically, people might visit a few locations in a trip, e.g., access some malls when shopping, travel to a branch of landmarks in a sightseeing tour, or go to a cinema from a restaurant, etc. These locations might be similar or dissimilar, nearby or far away from each other; but they are correlated from the perspective of human behavior. For example, a cinema and a restaurant are not similar in terms of the business categories they pertain to. However, in a user's mind, these places would be correlated if most people have visited these places in one trip. In other cases, to buy something important like a wedding ring, an individual would access some similar shops selling jewelry in a trip. In short, these shops visited by this individual might be correlated. However, these similar shops could be far away from each other, i.e., they might not be co-located in geographical spaces.

The correlation between locations can enable many valuable services, such as location recommendation systems, mobile tour guides, sales promotion and bus routes design. For instance, as shown in Figure 1 A), a new shopping mall is built in location *A* recently. The mall operator is intending to set up some billboards or advertisements in other places to attract more people's attention; hence promote the sales of this mall. By mining a large number of users' location histories, we discover that, in contrast to locations *D* and *F*, locations *B*, *C* and *E* have a much higher correlation with location *A*. Hence, if putting the billboards or promotion information in locations *B*, *C* and *E*, the operator is more likely to maximize the promotion effect with minimal investment.

Another example can be demonstrated using Figure 1 B). If we discover a museum and a landmark is highly correlated to a lake by analyzing many people's location histories, the museum and landmark can be recommended to tourists when they travel to the lake. Otherwise, people would miss some fantastic places even if they are only two hundred meters away from these locations.



A) Put promotion information or ads. at correlated locations

B) Recommend places to tourists in terms of location correlation

Figure 1. Some application scenarios of the location correlation

However, when mining the correlation from people's location histories, we need to face the following challenges.

1) The correlation among locations does not only depend on the number of the users visiting these locations in a trip but also these users' travel experiences. For instance, some overseas tourists might randomly visit some places in Beijing as they are not familiar with this city. However, the local people of Beijing are more capable than them of arranging a more proper and reasonable way to visit some places in Beijing [10].

2) The correlation between two locations, A and B, also depends on the sequences, in which the two locations have been visited. First, this correlation between A and B, $Cor(A,B)$, is asymmetric; i.e., $Cor(A,B) \neq Cor(B,A)$. Second, people would choose different sequences to visit two locations. Third, the

correlation among locations occurring in a user's trip might not be identical.

In this paper, we report on an approach mining the correlation between locations from human location history. Beyond the geo-distance relationship and the business category similarity between locations, the location correlation describes the relationship between locations in the space of human behavior. The correlation among locations can be a fundamental and key knowledge of many applications and services. The contribution of this paper lies in the following five parts. 1) We propose a method to uniformly model each individual's location history. 2) We design a model inferring each user's travel experience in a given geo-region. 3) We propose an algorithm learning the correlation between locations. This algorithm considers users' travel experiences and the sequentiality of the locations in a user's trip. 4) We conduct a personalized location recommendation system using the correlation in a collaborative filtering (CF) algorithm. 5) We evaluate these two cases using a large-scale real-world GPS dataset collected by 112 users over 1.5 year. As a result, our system significantly outperforms the Pearson correlation-based CF model [1] and the Slope One algorithm [4].

2. PRELIMINARY

2.1 Problem Definition

Definition 1. Trajectory. A user's trajectory *Traj* is a sequence of time-stamped points, $Traj = \langle p_0, p_1, \dots, p_k \rangle$, where $p_i = (x_i, y_i, t_i)$ $(i = 0,1, \dots, k)$; t_i is a timestamp, $\forall 0 \leq i < k, t_i < t_{i+1}$ and (x_i, y_i) are two-dimension coordinates of points.

Definition 2. $Dist(p_i, p_j)$ denotes the geospatial distance between two points p_i and p_j , and $Int(p_i, p_j) = |p_i.t_i - p_j.t_j|$ is the time interval between two points.

***Definition 3*: *Stay Point*.** A stay point s is a geographical region where a user stayed over a time threshold T_r within a distance threshold of D_r [3]. In a user's trajectory, s is characterized by a set of consecutive points $P = \langle p_m, p_{m+1}, \dots, p_n \rangle$, where $\forall m < i \leq n$, $Dist(p_m, p_i) \leq D_r, Dist(p_m, p_{n+1}) > D_r$ and $Int(p_m, p_n) \geq T_r$. Therefore, $s = (x, y, t_a, t_l)$, where

$$s.x = \sum_{i=m}^{n} p_i.x / |P|, \qquad (1)$$

$$s.y = \sum_{i=m}^{n} p_i.y / |P|, \qquad (2)$$

respectively stands for the average x and y coordinates of the collection P; $s.t_a = p_m.t_m$ is the user's arriving time on s and $s.t_l = p_n.t_n$ represents the user's leaving time.

As shown in Figure 2, $\{p_1, p_2, \dots, p_8\}$ formulate a trajectory, and a stay point would be detected from $\{p_3, p_4, p_5, p_6\}$ if $d \leq D_r$ and $Int(p_3, p_6) \geq T_r$. Please refer to [5] for the detailed algorithm.

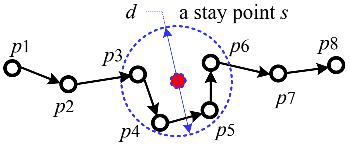


Figure 2. A trajectory and a stay point

***Definition 4: Location History*.** An individual's location history h is represented as a sequence of stay points they visited with corresponding arrival and leaving times,

$$h = \langle s_0 \xrightarrow{\Delta t_1} s_1 \xrightarrow{\Delta t_2}, \dots, \xrightarrow{\Delta t_{n-1}} s_n \rangle, \qquad (3)$$

where $\forall 0 \leq i < n, s_i$ is a stay point and $\Delta t_i = s_{i+1}.t_a - s_i.t_l$ is the time interval between two stay points.

However, so far, people's location histories are still inconsistent as the stay points detected from various individuals' trajectories are not identical. So, we put together the stay points detected from all users' trajectories into a dataset $\boldsymbol{S}$, and employ a clustering algorithm to partition this dataset into some clusters. Thus, the similar stay points will be assigned into the same cluster.

***Definition 5: Locations*.** $L = \{l_0, l_1, \dots, l_n\}$ is a collection of Locations, where $\forall 0 \leq i \leq n$, $l_i = \{s | s \in \boldsymbol{S}\}$ is a cluster of stay points detected from multiple users' trajectories; $i \neq j, l_i \cap l_j = \emptyset$.

After the clustering operation, we can substitute a stay point in a user's location history with the cluster ID the stay point pertains to. In short, a user's location history can be represented as a sequence of the locations. Supposing $s_0 \in l_i, s_1 \in l_j, s_n \in l_k$, Equation (3) can be replaced with

$$h = \langle l_i \xrightarrow{\Delta t_1} l_j \xrightarrow{\Delta t_2}, \dots, \xrightarrow{\Delta t_{n-1}} l_k \rangle. \qquad (4)$$

Later, we partition an individual's location history into some trips if the travel time spent between two consecutive locations exceeds a certain threshold T_p.

***Definition 6: Trip*:** A trip is a sequence of locations consecutively visited by a user, $Trip = \langle l_0 \xrightarrow{\Delta t_1} l_1 \xrightarrow{\Delta t_2}, \dots, \xrightarrow{\Delta t_{k-1}} l_k \rangle$, where $\forall 0 \leq i \leq k$, $\Delta t_k < T_p$ (a threshold) and $l_i \in L$ is a stay-point-cluster ID.

In short, a user's location history can be regarded as a collection of trips, $h = \{Trip\}$, and each $Trip = \langle l_i \rightarrow l_j \rightarrow \cdots \rangle$ is a sequence of locations represented by some clusters of stay points.

***Definition 7: Users*.** $U = \{u_0, u_1, \dots, u_m\}$ denotes the collection of users. $\forall 0 \leq k \leq m, u_k \in U$ is a user having a trajectory $Traj_k$, a location history h_k and certain travel experience e_k.

2.2 Framework

Figure 3 describes the framework for mining location correlation. First, as shown in Lines 2~4, we detect stay points from each user's trajectories, and formulate their own location histories into a sequence of stay points. Second, as depicted in Lines 5 and 6, we discover a set of locations L by clustering all users' stay points. Later, a user (u_k)'s location history (h_k) can be represented by a sequence of stay-point-clusters called locations here (refer to Lines 7 and 8). Third, we put all user's location history together, and learn each user's travel experience (e.g., e_k of u_k) using a iterative model (refer to Lines 9 and 10). Fourth, considering $\{(e_k, h_k), 0 \leq k < |U|\}$, we infer the correlation between locations, $Cor(l_i, l_j)$, where $l_i \in L$ and $l_j \in L, \forall 0 \leq i,j < |L|, i \neq j$.

```
MiningLocationCorrelation ($U, TRAJ, T_r, D_r, T_p$)
Input: A collection of users $U$ and their trajectories $TRAJ = \{Traj_k\}$,
         a time threshold $T_r$ and a distance threshold $D_r$ for stay point
         detection, and a $T_p$ for trip partition.
Output: A matrix Cor of correlation between each pair of locations.
1.  $S$=ϕ;   $H$ = ϕ;                          //temporal variables
2.  Foreach $u_k \in U$ do
3.        ST=StayPointDetection($Traj_k$, $T_r$, $D_r$); //refer to [8] for details
4.        $h_k$ = LocHistPresent(ST);          //a sequence of stay points
5.        $S$=$S$ ∪ ST;                    // a collection of all users' stay points
6.    $L$ = Clustering($S$);          //detect locations by clustering the stay points
7.    Foreach $u_k \in U$ do
8.        $h_k$ = LocHistRepresent($h_k$, $L$);     //a sequence of locations
9.        $H$= $H$ ∪ $h_k$;              //a collection of all users' location histories
10.  $E$=InferUserExperience($U, L, H$);              //refer to Section 3
11.  Cor=CalculateLocationCorrelation($L$,$E$, $H$,$T_p$); //refer to Section 4
12.  Return Cor.
```

Figure 3. The framework of our approach

3. INFERRING TRAVEL EXPERIENCE

As shown in Figure 4, we regard a user's stay on a location as an implicitly directed link from the user to that location, i.e., a user would point to many locations and a location would be pointed to by many users. Here, a green point stands for a stay point, and a gray-circle region denotes a location (a cluster of stay points).

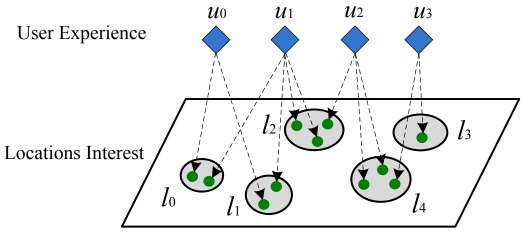


Figure 4. The model inferring user travel experience

User travel experience E and the location interest $\mathcal{T}$ have a mutual reinforcement relationship. The user with rich travel experiences in a region would visit many interesting places in that region, and a very interesting place in that region might be accessed by many users with rich travel experiences. More specifically, a user's travel experience can be represented by the sum of the interests of the locations they accessed; in turn, the interest of a location can be calculated by integrating the experiences of the users visiting it. Using a power iteration method, each user's experience and each location's interest can be calculated (refer to [10]).

Given a collection of users U's location histories H, we can build a adjacent matrix M between users and locations. In this matrix, an item r_{ij} stands for the times that u_i has stayed in location l_j , $0 \le i < |U|, 0 \le j < |L|$. For instance, the matrix specified by Figure 4 can be represented as follows.

$$M = \begin{array}{cc} & \begin{array}{ccccc} l_0 & l_1 & l_2 & l_3 & l_4 \end{array} \\ \begin{array}{c} u_0 \\ u_1 \\ u_2 \\ u_3 \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}; \qquad (5)$$

Then, the mutual reinforcement relationship of user travel experience $E = (e_0, e_1, \dots, e_m)$ and location interest $\mathcal{T} = (I_0, I_1, \dots, I_n)$ is represented as follows:

$$e_i = \sum_{l_j \in L} r_{ij} \times I_j; \qquad (6)$$

$$I_j = \sum_{u_i \in U} r_{ji} \times e_i; \qquad (7)$$

where e_i stands for u_i 's travel experience and I_j denotes the location interest of l_j . Writing them in the matrix form,

$$\boldsymbol{E} = \boldsymbol{M} \cdot \boldsymbol{\mathcal{T}}, \qquad (8)$$

$$\boldsymbol{\mathcal{T}} = \boldsymbol{M}^{\mathrm{T}} \cdot \boldsymbol{E}. \qquad (9)$$

If we use $\boldsymbol{\mathcal{T}}_n$ and $\boldsymbol{E}_n$ to denote location interests and travel experiences at the nth iteration, the iterative processes for generating the final results are

$$\boldsymbol{\mathcal{T}}_n = \boldsymbol{M}^{\mathrm{T}} \cdot \boldsymbol{M} \cdot \boldsymbol{\mathcal{T}}_{n-1} \qquad (10)$$

$$\boldsymbol{E}_n = \boldsymbol{M} \cdot \boldsymbol{M}^{\mathrm{T}} \cdot \boldsymbol{E}_{n-1} \qquad (11)$$

Starting with $\boldsymbol{\mathcal{T}}_0 = \boldsymbol{E}_0 = (1,1,\dots,1)$, we are able to calculate the final results using the power iteration method.

4. LOCATION CORRELATION

The correlation between two locations can be calculated by integrating the travel experiences of the users U' who have visited them in a trip in a weighted manner. Formally, the correlation between location A and B can be calculated as

$$Cor(A,B) = \sum_{u_k \in U'} \alpha \cdot e_k, \qquad (12)$$

where U' is the collection of users who have visited A and B in a trip, e_k is u_k 's travel experience, $u_k \in U'$, and $0 < \alpha \le 1$ is a dumping factor, which will decrease as the interval between these two locations' index in a trip increases. For example, in our experiment we set $\alpha = 2^{-(|j-i|-1)}$, where i and j are indices of A and B and in the trip they pertain to; i.e., the more discontinuously two locations being accessed by a user ($|i$-$j|$ would be big, thus α will become small), the less contribution the user can offer to the correlation between these two location.

As depicted in Figure 5, three users (u_1, u_2, u_3) respectively access locations (A, B, C) in different manners and create three trips $(Trip_1, Trip_2, Trip_3)$. The number shown below each node denotes the index of this node in the sequence.

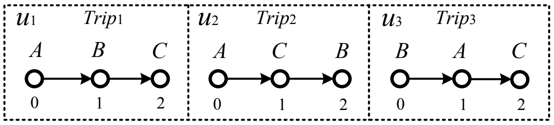


Figure 5. A case calculating the correlation between locations

According to Equation (12), from $Trip_1$ we can calculate $Cor(A,B) = e_1$ and $Cor(B,C) = e_1$, since these locations have been consecutively accessed by u_1 (i.e., $\alpha = 1$). However, $Cor(A,C) = \frac{1}{2} \cdot e_1$ (i.e., $\alpha = 2^{-(|2-0|-1)} = \frac{1}{2}$) as u_1 traveled to B before visiting C. In other words, the correlation between location A and C that we can sense from $Trip_1$ might not that strong as if they are consecutively visited by u_1. Likewise, we can learn $Cor(A,C) = e_2$, $Cor(C,B) = e_2$, $Cor(A,B) = \frac{1}{2} \cdot e_2$ from $Trip_2$, and infer $Cor(B,A) = e_3, Cor(A,C) = e_3, Cor(B,C) = \frac{1}{2} \cdot e_3$ from $Trip_3$. Later, we can integrate these correlation inferred from each user's trips and obtain the following results.

$$Cor(A,B) = e_1 + \frac{1}{2} \cdot e_2; \; Cor(A,C) = \frac{1}{2} \cdot e_1 + e_2 + e_3;$$

$$Cor(B,C) = e_1 + \frac{1}{2} \cdot e_3; \; Cor(C,B) = e_2; Cor(B,A) = e_3.$$

5. Personalized Location Recommendation

The personalized location recommendation systems aim to predict an individual's taste in some locations using their location history and those of multiple people. The location correlation is integrated into a collaborative filtering (CF) algorithm [1] to achieve a personalized location recommendation system.

Notation: The ratings from a user u_p, called an *evaluation*, is represented as an array $R_p = \langle r_{p0}, r_{p1}, \dots, r_{pn} \rangle$, where r_{pj} is u_p's implicit ratings (the occurrences) in location $l_j, 0 \le j < |L|$. $S(R_p)$ is the subset of the R_p, $\forall r_{pj} \in S(R_p), r_{pj} \neq 0$, i.e., the set of items (locations) which has been rated (visited) by u_p. The average of ratings in R_p is denoted as $\overline{R_p}$, and the number of elements in a set S is $|S|$. The collection of all *evaluations* in the training set is $\mathcal{X}$. $S_j(\mathcal{X})$ means the set of evaluations containing item j, $\forall R_p \in S_j(\mathcal{X}), j \in S(R_p)$. Likewise, $S_{i,j}(\mathcal{X})$ is the set of evaluations simultaneously containing item i and j.

Intuitively, to predict u_p's rating of location A given u_p's ratings of location B and C, if location B is more related to A beyond C, then u_p's ratings of location B is likely to be a far better predictor for location A than u_p's ratings of location C is. In contrast to the number of observed ratings (i.e., the number of people who have visited two locations) used by the weighted Slope One algorithm, the location correlation mined from multiple users' location histories carries more semantic meanings. Formally, our approach can be represented as

$$P\left(r_{pj}\right)=\frac{\sum_{i\in S(R_p)\wedge i\neq j}(dev_{j,i}+r_{pi})\cdot c_{ji}}{\sum_{i\in S(R_p)\wedge i\neq j}c_{ji}}, \tag{13}$$

$$dev_{j,i}=\sum_{R_p\in S_{j,i}(\mathcal{X})}\frac{r_{pj}-r_{pi}}{|S_{j,i}(\mathcal{X})|}, \tag{14}$$

where c_{ji} denotes the correlation between location l_i and l_j, and $dev_{j,i}$ is still calculated as Equation (14). Using Equation (13), we can predict an individual's ratings on the locations they have not accessed, and then rank these locations in terms of the predicted ratings. Later, the top *n* locations with relatively high ratings can be recommended to the individual.

6. EXPERIEMENT

6.1 Settings

Datasets: Carrying a GPS-enabled device, 112 users recorded their outdoor movements with GPS logs from May 2007 to Dec. 2008. The total distance of the GPS logs exceeded 254,030,449 kilometers, and the total number of GPS points reached 9,432,747. Most parts of this dataset were created in Beijing, China, and other parts covered 36 cities in China.

Stay point detection: In this experiment, we set T_r to 20 minutes and D_r to 250 meters for stay point detection. Using these parameters, 13,766 stay points were extracted.

Clustering: We use OPTICS (Ordering Points To Identify the Clustering Structure) to cluster stay points into some geospatial regions. We set the core-distance (d_c) to 100 meters and configure the minimum number of points ($minPt$) to 8.

6.2 Results

Effectiveness: Using the average *NDCG* and *MAP*, Table I compares the effectiveness of different methods in conducting the personalized location recommendation. Clearly, our approach (*Experience + Sequentiality*) outperforms the weighted Slope One algorithm (T-Test of *NDCG@5*, *p*=0.0053<0.01; T-Test of *MAP*, *p*=0.0049<0.01). Although our method is slightly weaker than the Pearson correlation-based CF model in terms of the average *NDCG* and *MAP*, the T-test result (*NDCG@5, p*=0.678>>0.01; *MAP, p*=0.741>>0.01) shows that the advantage of the Pearson correlation is not significant and not clear. In other words, some users thought the recommendation generated by our method is even better than that of the Pearson correlation-based scheme. Thus, we can claim that at least our method is as effective as the Pearson correlation-based one.

Table I. Effectiveness of the personalized location recommendation using different methods

	Ours	**The Pearson Correlation-Based CF model**	**The Weighted Slope One Algorithm**
NDCG@5	**0.840**	0.862	0.762
NDCG@10	**0.922**	0.938	0.891
MAP	**0.798**	0.804	0.665

Efficiency: Using the given GPS dataset, Figure 6 depicts the upper bound of computing complexity of different methods in calculating a prediction.

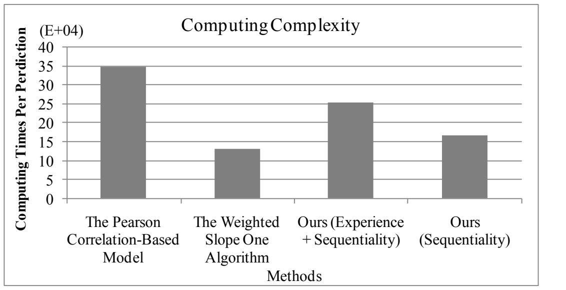


Figure 6. Average computing complexity in computing a prediction

7. CONCLUSION

In this paper, by considering the user travel experience and the sequentiality that locations have been visited, we designed an approach to mine the correlation between locations from a large amount of people's location histories. Beyond the geo-distance and the category relationship between locations, the correlation describes a more comprehensive relationship between locations in the space of human behavior and is a more nature way for human understanding. Using the location correlation, we conducted a personalized location recommendation system. We evaluated these two cases with a real-world large-scale GPS dataset. As a result, the personalized location recommendation is more effective than the weighted Slope one algorithm with a slightly additional computation. In addition, in contrast to the Pearson correlation-based CF model, our method is much more efficient while keeping the similar effectiveness.

8. REFERENCES

[1] Adomavicius, G. and Tuzhhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transaction on Knowledge and Data Engineering. 17, 6, 734-749.

[2] Huang, Y., Shekhar, S., and Xiong, H.. Discovering co-location patterns from spatial datasets: A general approach. TKDE, 16(12):1472-1485, 2004

[3] Hariharan, R. et al. Project Lachesis: Parsing and Modeling Location Histories, In Proceedings of GIScience, (Park Utah, October 2004), ACM Press: 106-124.

[4] Lemire D. and Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering. In Proceedings of SDM 2005.

[5] Li, Q. and Zheng, Y. et al. Mining user similarity based on location history. In Proc. of GIS'08 (Santa Ana, CA, Nov. 2008). ACM Press

[6] Morimoto, Y.. Mining frequent neighboring class sets in spatial databases. In Proceedings of SIGKDD, 2001, ACM Press: 353-358.

[7] Zhang, X., N. Mamoulis, D. W. Cheung, and Y. Shou. Fast mining of spatial collocations. In Proceedings of SIGKDD, ACM Press: 384-393, 2004.

[8] Zheng, Y., Li Q., Xie X., Ma, W. Y.. Understanding mobility based on GPS data. In Proceedings of Ubicomp'08, (Seoul Korea, Sept. 2008), ACM Press: 312-321

[9] Zheng, Y., Liu, L., Wang, L. Xie, X. Learning transportation modes from raw GPS data for geographic applications on the Web. In Proceedings of WWW 2008, (Beijing China, April 2008), ACM Press: 247-256.

[10] Zheng, Y., Zhang, L., Xie X., Ma, W. Y. Mining interesting locations and travel sequences from GPS trajectories for mobile users. In Proceeding of WWW2009, (Madrid, Spain. April 2009), ACM Press: 791-800.