Recommending Friends and Locations Based on Individual Location History

YU ZHENG, Microsoft Research Asia
LIZHU ZHANG and ZHENGXIN MA, Tsinghua University
XING XIE and WEI-YING MA, Microsoft Research Asia

The increasing availability of location-acquisition technologies (GPS, GSM networks, etc.) enables people to log the location histories with spatio-temporal data. Such real-world location histories imply, to some extent, users' interests in places, and bring us opportunities to understand the correlation between users and locations. In this article, we move towards this direction and report on a personalized friend and location recommender for the geographical information systems (GIS) on the Web. First, in this recommender system, a particular individual's visits to a geospatial region in the real world are used as their implicit ratings on that region. Second, we measure the similarity between users in terms of their location histories and recommend to each user a group of potential friends in a GIS community. Third, we estimate an individual's interests in a set of unvisited regions by involving his/her location history and those of other users. Some unvisited locations that might match their tastes can be recommended to the individual. A framework, referred to as a hierarchical-graph-based similarity measurement (HGSM), is proposed to uniformly model each individual's location history, and effectively measure the similarity among users. In this framework, we take into account three factors: 1) the sequence property of people's outdoor movements, 2) the visited popularity of a geospatial region, and 3) the hierarchical property of geographic spaces. Further, we incorporated a content-based method into a user-based collaborative filtering algorithm, which uses HGSM as the user similarity measure, to estimate the rating of a user on an item. We evaluated this recommender system based on the GPS data collected by 75 subjects over a period of 1 year in the real world. As a result, HGSM outperforms related similarity measures, namely similarity-by-count, cosine similarity, and Pearson similarity measures. Moreover, beyond the item-based CF method and random recommendations, our system provides users with more attractive locations and better user experiences of recommendation.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5 [**Computing Methodologies**]: Pattern Recognition; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering, retrieval model*; H.2.8 [**Database Applications**]: Spatial Databases and GIS

General Terms: Algorithms, Measurement, Experimentation

Additional Key Words and Phrases: Recommender system, spatio-temporal data mining, user similarity, GPS trajectories, location history, collaborative filtering, GeoLife

ACM Reference Format:
Zheng, Y., Zhang, L., Ma, Z., Xie, X., and Ma, W.-Y. 2011. Recommending friends and locations based on individual location history. ACM Trans. Web 5, 1, Article 5 (February 2011), 44 pages.
DOI = 10.1145/1921591.1921596 http://doi.acm.org/10.1145/1921591.1921596

5

This article is an expanded version of Li et al. [2008], which appeared in *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographical Information Systems*, 247–256.
Authors' addresses: Y. Zheng, X. Xie, and W.-Y. Ma, Microsoft Research Asia, Beijing 100190, China; email: {yuzheng, xingx, wyma}@microsoft.com. L. Zhang and Z. Ma, Department of Electronic Engineering, University of Tsinghua, Beijing 100184, China; email: {zlz02, mazx}@tsinghuaedu.cn.

DOI 10.1145/1921591.1921596 http://doi.acm.org/10.1145/1921591.1921596

1. INTRODUCTION

Recommender systems are changing the way people interact with the Web by providing a more personalized information access experience than searching. Typically, these systems estimate a particular user's interests from the data implicitly or explicitly generated by the user. In addition, the social environment of a user is usually involved in inferring their taste. Hence, digital results matching an individual's preferences are more likely to be retrieved for him/her.

In the past years, companies like Amazon [Linden et al. 2003] have shown the effectiveness of recommender systems in improving the sales of a retailer. However, so far, most of the products and researches related to recommendation are based on online user behavior in Web communities, such as news recommenders [Das et al. 2007] and music recommenders [Li et al. 2007; Tiemann et al. 2007].

Recently, the increasing pervasiveness of location-acquisition technologies, like GPS and GSM networks, are leading to the collection of large spatio-temporal datasets, which bring the opportunity of discovering valuable knowledge about users' movements. A branch of geographic applications based on user-generated GPS data have appeared on the Web, and received considerable attention. In such applications [Bikely; GPS Sharing; SportsDo; Counts and Smith 2007; Zheng et al. 2008c, 2009a, 2010d], using a GPS-enabled device, individuals can record their outdoor movements with GPS trajectories when traveling in the real world. Later, these individuals are able to upload these logs to a Web community where they can visualize and browse their own travel/sports experiences on a Web map. These systems tell the users' basic information, such as distance, duration, and velocity, of a particular route; tags and photos can also be shown for the route. Further, users are able to exchange life experiences among each other by sharing their GPS trajectories in the Web community.

GPS-log-sharing provides people with a more explicit and fancy approach than the text-based description to express their life experiences. For example, rich information, such as velocity/acceleration/bearing/altitude of each point, slope/curvature of a segment and the names of locations a user passed by, can be mined out from a bike-riding trajectory. In this manner, users are facilitated to absorb knowledge from others' past experiences. Meanwhile, by browsing other people's GPS trajectories on a Web map, an individual is likely to discover a travel route that interests him/her. Hence, the individual can get references when making a decision for travel planning. Unfortunately, so far, these applications still use raw GPS data directly without much understanding. Facing a large dataset of GPS trajectories, users have to spend a lot of manual effort to discover locations matching their tastes by themselves.

In contrast to users' online activities, people's outdoor movements in the real world would imply more information about their interests and preferences. For instance, if a person usually goes to stadiums and gyms, it denotes that the person might like sports. Likewise, if a user frequently travels to some mountains, it might imply that the user is interested in hiking. According to the first law of geography [Tobler 1970], "everything is related to everything else, but near things are more related than distant things", people who have similar location histories might share similar interests and preferences. The more location histories they share, the more correlated these two users would be. It is not difficult to understand that people who visit the same restaurants and shopping malls might share some similar entertainment interests. Also, users traveling to the same lakes and valleys might pertain to the similar style of tourists. In turn, the geographical regions visited by similar users might imply a similar profile. As a consequence, people's location histories can not only help us understand the similarity between individuals but also reveal the correlations among geographic locations.

In this article, we report on a personalized friend and location recommender system. This system 1) uses a particular individual's visits on a geospatial location in the real world as his/her implicit ratings on the location, 2) estimates the similarity between users in terms of their location histories, and 3) infers an individual's interests in an unvisited place by involving his/her location history and those of other users. In this system, each user will be recommended two categories of objects, similar users (potential friends) who might share similar places preferences and geospatial regions that could match a user's tastes although have not having been found by themselves. Therefore, an individual is first able to organize with minimal effort some social activities, such as hiking and cycling. In short, with such a friend list in the community, a user is more capable of delivering invitations to the right candidates who might also have a passion related to that invitation. Second, given the recommended places from such potential friends' location histories, users can easily expand their travel knowledge and discover the locations that interest them.

The work reported in this article is a location-history-based recommender system, which estimates the similarity between users in terms of their movements in geographical spaces. This is a step toward estimating a user's tastes on items (locations) they have not considered (visited) using the user's implicit ratings and social environment. From the algorithm's perspective, this system moves toward incorporating the content-based method into a user-based collaborative filtering algorithm. This is also a step toward associating recommender systems with geographical information systems on the Web. The main contributions of this work lie in the following three aspects.

(1) We propose a framework, referred to as hierarchical-graph-based similarity measurement (HGSM), which uniformly models people's location histories and effectively estimates the similarity between users. In this framework, we consider the following three factors.
   —Sequence property of users' movements. We take into account not only the geographic regions they accessed, but also the sequence of these regions being visited. The longer similar sequences matched between two users' location histories, the more related these two users might be.
   —Hierarchy property of geographic spaces. We mine user similarity by exploring people's movements on different scales of geographic spaces. Users who share similar location histories on geographical spaces of finer granularities might be more correlated.
   —Popularity of different locations. Analogous to inverse document frequency (IDF), we consider the visited popularity of a geographical region when measuring the similarity between users. Two users who accessed a location visited by a few people might be more correlated than others who share a location history accessed by many people. For instance, lots of people have visited the Great Wall, a well-known landmark in Beijing. However, it might not mean all these people are similar to one another. However, if two users visited a restaurant which is not that famous, they might indeed share some similar preferences.
(2) Using HGSM to estimate the similarity between users, a collaborative filtering-based method is employed in our recommender system to infer an individual's interests in unvisited geospatial regions. Meanwhile, we understand the profile of a geospatial region by exploring the categories of points of interest (POIs) within the region. Therefore, we are able to find geospatial regions with similar profiles which enable us to integrate the content-based method into collaborative filtering. This approach can reduce to some extent the cold start problem of our systems. Also, such profiles endow us with the ability to filter some boring locations, such as people's abodes. Moreover, the approach allows us to recommend various types

of locations based on users' requests on different occasions. For instance, we can recommend a region covering some restaurants to an individual searching for a place for dinner. Likewise, we are able to suggest a geospatial region containing some malls to a person when he/she prefers to go shopping.

(3) We evaluate our approach using a large-scale GPS dataset collected by 75 people over a period of one year in the real world. The total number of GPS points almost reached 7 million (6,963,824), and its total distance exceeded 135 thousand (135,940) kilometers. As a result, our HGSM outperforms the baseline methods, such as the Cosine similarity and Pearson similarity, in measuring the similarity between users based on location history. Moreover, beyond the item-based collaborative filtering, our approach provides users with more attractive places and more personalized user experiences.

The rest of this article is organized as follows. In Section 2, we first present the user interface of the system. Later, the architecture of our recommender system, which consists of three parts, location history representation, user similarity mining, and CF-based location recommendation, is introduced. In Section 3, we detail the processes of mining the similarity between users based on their location histories. Section 4 describes the CF-based location recommender, and Section 5 reports major experimental results. After giving a survey on the related works in Section 6, we draw our conclusions in Section 7, and propose the future work we attempt to conduct in Section 8.

2. OVERVIEW OF OUR RECOMMENDER SYSTEM

In this section, we first demonstrate the user interface of this recommender system using a few cases. Then, we define some terms used in this article and briefly introduce the architecture of our system.

2.1. User Interface of the System

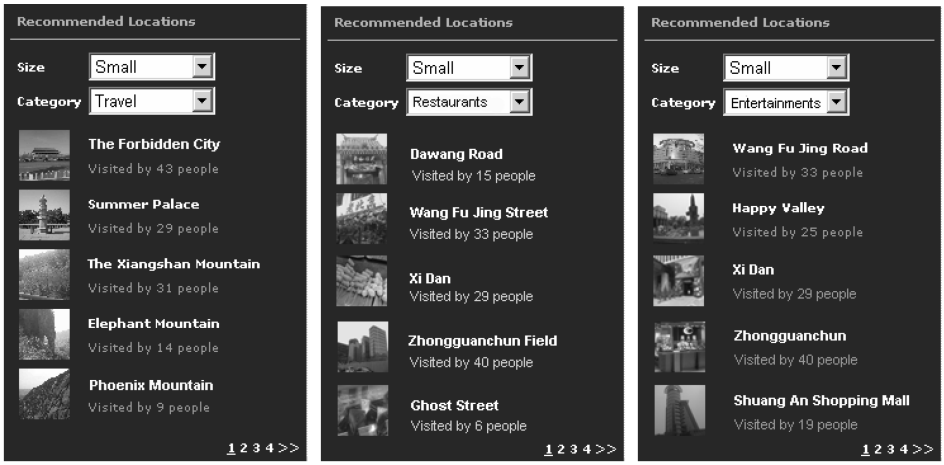
The recommender system reported in this article is an important component of our project GeoLife [Zheng et al. 2009a and 2010d], which is a GPS-log-driven application on Web maps. GeoLife focuses on lively visualization [Zheng et al. 2008c and 2008d], fast retrieval [Wang et al. 2008; Chen et al. 2010] and a deep understanding of GPS track logs [Li et al. 2008; Zheng et al. 2008a, 2008b, 2009b, 2010a, 2010b, 2010c, 2010e, 2010f] for both personal and public use. This recommender system has been deployed in the prototype of GeoLife as a part of research result.

Figure 1 presents the user interface of our recommender system. A particular user, John, can sign in GeoLife using his Live ID. In GeoLife, we help John maintain a personal Web site, called MyGeoLife, where John can upload and manage his own trajectory data. By default, the data is private for John's personal use; however, John can pick out some GPS trajectories to share with others if he desires. Once he makes public some trajectories, we are able to provide him with a more personalized location recommendation.

After logging onto MyGeoLife, in the right box of the window, John can discover a group of potential friends and a set of geospatial locations recommended to him. These potential friends are more likely to share similar tastes (in terms of location history) with John as compared to other users in this community. Below the recommended friends, the top five geospatial regions that might match John's interests are also listed with corresponding thumbnails. These regions are mined from those potential friends' past experiences; John has not found them by himself. With a database of POI, we are able to identify a proper name for a given region using an inverse geo-coding technology.

Further, we can understand the properties of a georegion based on the categories of the POIs located in this region. Here, we differentiate four kinds of categories,

Fig. 1. User interface of the recommender system.



(a) Travel recommendation (b) Restaurant recommendation (c) Entertainment recommendation

Fig. 2. Render recommendation results according to a user's preferences on different categories.

"restaurant," "sports," "entertainments" and "travel." (Refer to Section 4.2 for details.) Therefore, as illustrated in Figure 2(a), John can select "Travel" in the category combo box when he intends to find some interesting landscapes like the Summer Palace. Or, as demonstrated in Figure 2(b), he can select the category of "restaurant" if he prefers to look for a place for dinner, e.g., Sanlitun. Of course, if John does not specify any categories, for example, using "All," in Figure 1, locations of various types would be recommended together. All these results mentioned above are ranked based on their ratings estimated by our algorithm.

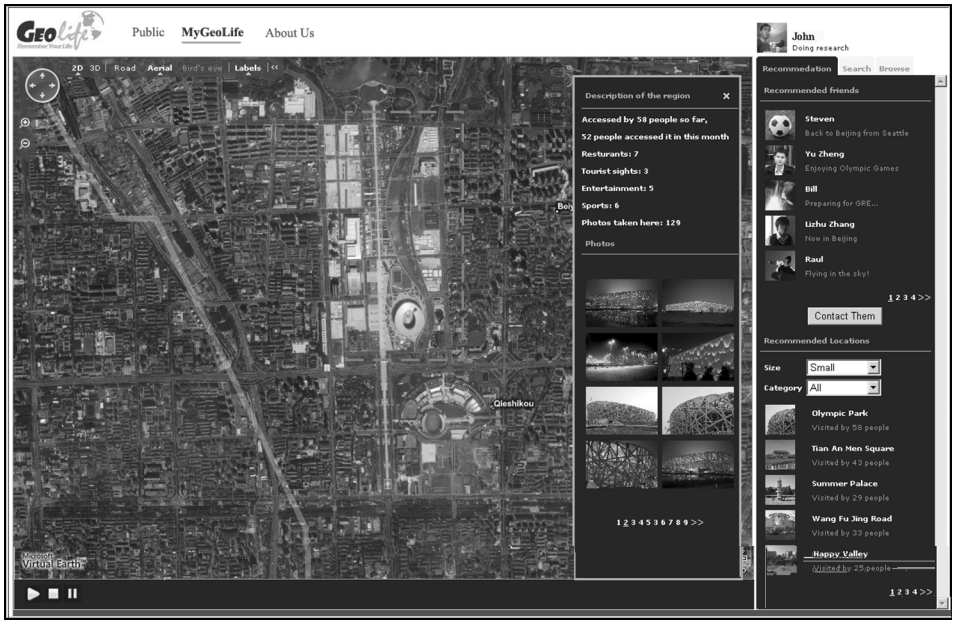


Fig. 3. View a geospatial region in our recommender system.

As depicted in Figure 3, John can take a closer look at a recommended location by clicking the icon of this location in the results list. In a pop-up information box, John can obtain summarized information of this region and browse a set of photos taken by other users visiting the region. Meanwhile, he can view the POIs and businesses located in the region on the map (if switching the map to a road view). Thus, he is able to make a decision whether this place deserve, his arrival before really accessing it.

If John is attracted by the location shown in Figure 3, he can invite a group of people from the community to visit there together. As demonstrated in Figure 4, by clicking the "contact them" button, John will be provided with an interface where he can send his proposal with a suggested destination to the potential friends in this community. After receiving the invitation message from John, these potential friends can view what the proposed region looks like by browsing the Web map and photos taken within this region. Later, they are able to make their own decision on whether to join this activity.

2.2. Difference between This Article and Our Previous Publication

In the previous publication of GIS 2008, we proposed only a preliminary measurement that estimates the similarity between users in terms of their location histories. In this paper, we first improve the similarity measurement and then conduct a friend and location recommendation system, employing the improved similarity measurement. More specifically, the differences lie in the following three aspects.

(1) *Conduct a personalized friend and location recommendation system.* In this article we integrate the user similarity into a collaborative filtering (CF) model to conduct a personalized friend and location recommendation system. This is a totally new research we performed after the GIS publication. This work includes the following.
   (a) Using a collaborative filtering (CF) based method, our system involves the location histories of a user's potential friends to estimate the user's interests on a set of unvisited georegions. Refer to Section 4.

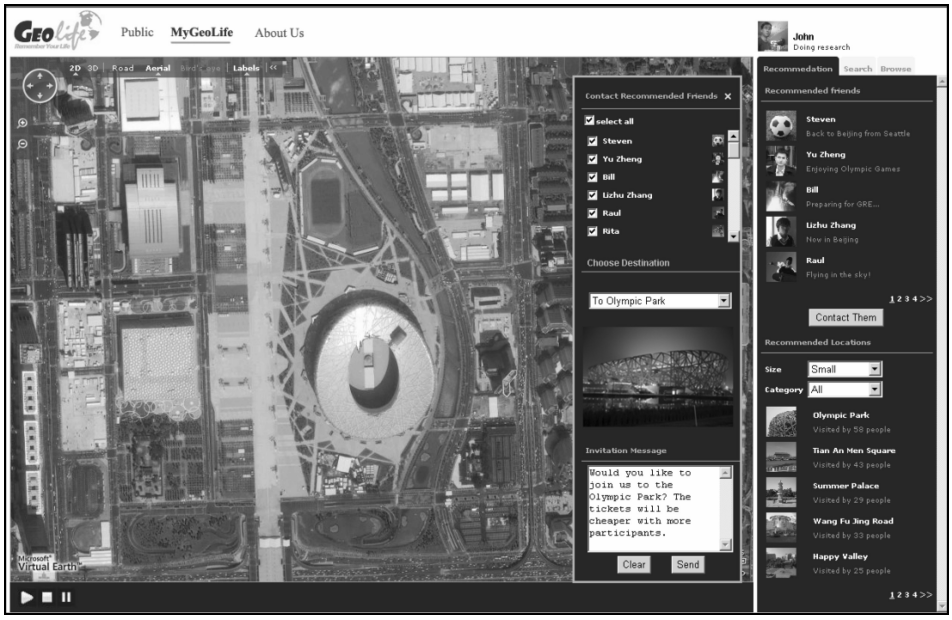


Fig. 4. Propose a social activity by sending a proposal to potential friends in the community.

(b) By understanding the profile of a geospatial region so that a content-based method is integrated into the location recommender to reduce the cold start problem. Refer to Section 4.3.
(c) We building a prototype of this recommender system and showcasing its user interfaces in Section 2.1.
(d) We evaluate this recommender system (not the similarity measurement) based on the GPS data collected by 75 subjects over a period of 1 year in the real world. A study investigating users' feedback on the recommended locations is reported in Section 5.3.2.

(2) *We improve our similarity meas as follows.*
(a) We propose a new sequence matching strategy. By splitting the long sequence into several short sequences, we enhanced the efficiency of the matching process while keeping its effectiveness. Refer to Section 3.2.2 for details.
(b) We take into account the popularity of a location, which improves the performance of the measurement. Analog to IDF, we consider the visited popularity of a geographical region when measuring the similarity between users. Refer to Section 3.3.

(3) *More evaluation and discussion.*
(a) In this article, we evaluated the performance of the improved measurement using a real-world GPS dataset. Also, we studied the effectiveness of an IDF feature when integrating with different similarity measurements, such as the Cosine similarity and Pearson similarity.
(a) More experiments and discussion have been conducted in this research. For example, the new user problem and the new location problem of recommender system have been discussed and considered. Meanwhile, based on the newly performed experimental results, we give more justifications in choosing the parameters for the algorithm. Refer to 5.1.4

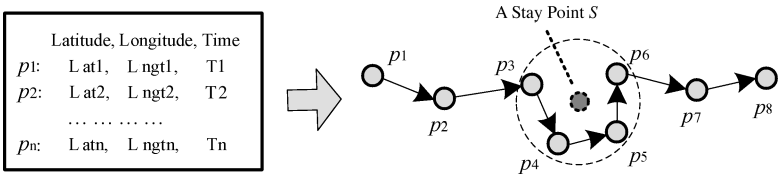


Fig. 5. A GPS trajectory and a stay point.

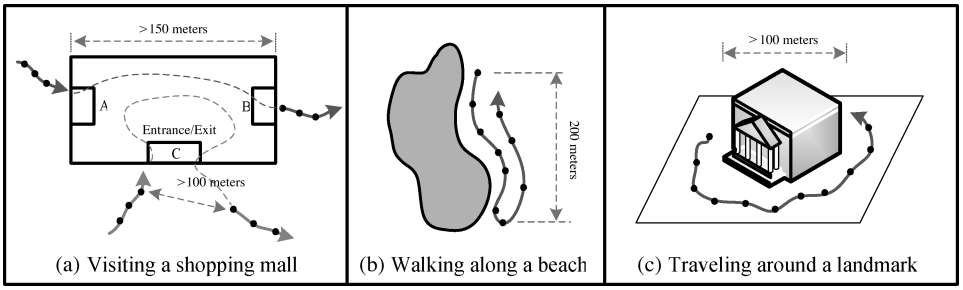


Fig. 6. Some examples of stay points.

2.3. Preliminary

In this section, we will define some terms: GPS trajectory (*Traj*), stay point(s), location history (*LocH*), and hierarchical graph (*HG*).

Definition 1 (*GPS Trajectory*). A GPS trajectory (*Traj*) is a sequence of GPS points, each of which contains a latitude ($p_i.Lat$), longitude ($p_i.Lngt$) and timestamp ($p_i.\ T$). Thus, $Traj = p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_n$, where $p_i.T < p_{i+1}.T$.

Definition 2 (*Stay Point*). Generally speaking, a stay point(s) stands for a geographic region where a user stayed over a certain time interval. The extraction of a stay point depends on two scale parameters, a time threshold (θ_t) and a distance threshold (θ_d). Formally, a single stay point(s) can be regarded as a virtual location characterized by a group of consecutive GPS points $P = \{p_m, p_{m+1}, \ldots, p_n\}$, where $\forall m < i \leq n, Distance(p_m, p_i) \leq \theta_d$ and $|p_n.T - p_m.T| \geq \theta_t$. Conditioned by P, θ_d and θ_t, a stay point $s = (Lat, Lngt, arvT, levT)$, where

$$s.Lat = \sum_{i=m}^{n} p_i.Lat/|P|, \tag{1}$$

$$s.Lngt = \sum_{i=m}^{n} p_i.Lngt/|P|, \tag{2}$$

respectively, stands for the average latitude and longitude of the collection P, and $s.arvT = p_m.T$ and $s.levT = p_n.T$ represents a user's arrival and leaving times on s.

As demonstrated in Figure 5, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_8$ formulates a GPS trajectory and a stay point can be constructed by points $\{p_3, p_4, p_5, p_6\}$.

Typically, these stay points occur in the following two situations. One is when people enter a building and lose a satellite signal over a time interval, until coming back outdoors. Figure 6(a) shows an example using an individual's GPS trajectory while visiting a shopping mall. The other situation is when a user exceeds a time limit at a certain geospatial area. For instance, people strolling along a nice beach (refer to

Algorithm StayPointDetection(*Traj*, θ_d , θ_t)

Input: A GPS trajectory *Traj*, a distance threshold θ_d and a time span threshold θ_t
Output: A set of stay points $S=\{s\}$

```
1. i=0, pointNum = |Traj|; //the number of GPS points in Traj
2. while i < pointNum do,
3.       j:=i+1;
4.       while j < pointNum do,
5.             dist=Distance(pi , pj ); //calculate the distance between two points
6.             if dist > θd then
7.                   ΔT=pj .T-pi .T;  //calculate the time span between two points
8.                   if ΔT> θt then
9.                         s.Lat=ComputMean({pk .Lat | i<=k<=j})   //equation (1)
10.                        s.lngt=ComputMean({pk .lngt | i<=k<=j}) //equation (2)
11.                        s.arvT= pi .T;  S.levT=pj .T ;
12.                        S.insert(s);
13.                  i:=j; break;
14.            j:=j+1;
15.  return S.
```

Fig. 7. The algorithm for stay point detection.

Figure 6(b)), or being attracted by a landmark building (See Figure 6(c)) could generate a stay point.

Here, we hope to represent each stay of a user as precisely as possible. Unfortunately, we have to use a proper georegion to specify an individual's stay for to the following reasons.

First, a strict region size, such as 20×20 meters, might be more capable of accurately identifying a business like a Starbucks visited by a user; however, it would cause many stays to remain undetected. As demonstrated in Figure 6(a), a user could enter a shopping mall from Gate A while leaving the mall from Gate B (see the blue line). Given a shopping mall could cover a 150×150 meter georegion, the distance between the last GPS point before entering the mall and the first point after coming out from the mall could be larger than 150 meters (i.e., the user's stay at this shopping mall cannot be detected using a very small region constraint like 20 meters). Moreover, even if a user leaves the shopping mall from the same gate they entered, like Gate C, in most cases, the distance between the last GPS point before entering and the first point after coming out could be larger than 100 meters. Typically, the GPS devices need some time to relocate themselves after coming back outdoors, while people do not have patience to wait.

Second, sometimes a very small region constraint could cause the stays of people to be overdetected. As shown in Figure 6(b) and (c), multiple stay points could be detected when people stroll along a beach or wander around a landmark. This is not aligned with people's intuitiveness, as in their minds they only access one place (the beach or the landmark).

Using the algorithm shown in Figure 7, these stay points can be detected automatically from a user's GPS trajectory. For instance, in our experiment, if an individual spent more than 30 minutes within a distance of 200 meters, the region is detected

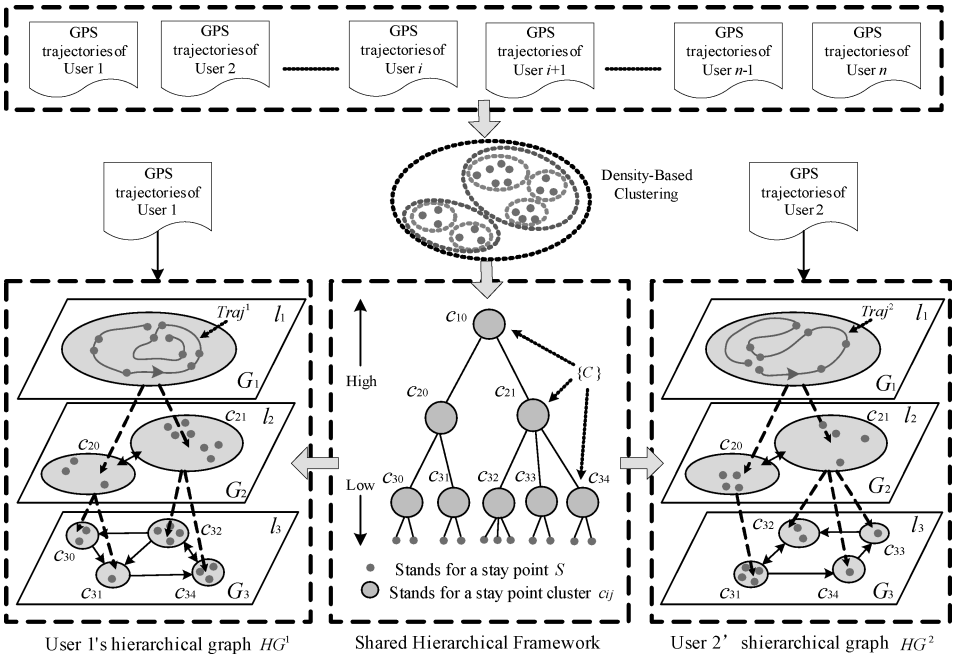


Fig. 8. Hierarchical graph modeling individual location history.

as a stay point. As compared to a raw GPS point, each stay point carries a particular semantic meaning, such as the shopping mall we accessed and the restaurants we visited. The reasons why we detect stay points using the algorithm shown in Figure 7, rather than directly clustering raw GPS points, lie in two aspects. (1) Most significant places like shopping malls and restaurants cannot be detected if we directly cluster raw GPS points, as GPS devices lose satellite signal indoors (i.e., few GPS points will be generated on such places). (2) Using an interpolation operation, the computation of clustering such a big dataset will be extremely heavy with increasing users. Refer to Section 5.4.1 for details.

Definition 3 (*Location History*). Generally, location history is a record of locations that an entity visited in geographical spaces over an interval of time. In this article, an individual's location history (*LocH*) is represented as a sequence of stay points (s) he/she visited with corresponding arrival and leaving times.

$$LocH = \left(s_1 \overset{\Delta t_1}{\rightarrow} s_2 \overset{\Delta t_2}{\rightarrow}, \ldots, \overset{\Delta t_{n-1}}{\rightarrow} s_n\right), \quad \text{where} \quad s_i \in S \quad \text{and} \quad \Delta t_1 = s_{i+1}.arvT - s_i.levT.$$

However, the location histories of various people are inconsistent and incomparable, as the stay points pertaining to different individuals are not identical. Also, it is subjective to directly measure how similar two stay points are based on the geodistance between them. Moreover, user similarity is not a binary value (That is, it is not reasonable to judge whether two users are similar or not). What we aim to do is to identify how relevant two individuals are as compared to others, and then, for each user, rank a group of people according to the similarity between them.

To address the preceding issue, we propose a framework, called hierarchical graph (*HG*), to uniformly model each individual's location history. As illustrated in Figure 8, to build such a graph for every user, two steps need to be performed.

(1) Formulate a shared framework F. We put all users' stay points together into a dataset. Using a density-based clustering algorithm, we hierarchically cluster this dataset into several geospatial regions (clusters C) in a divisive manner. Thus, the similar stay points from various users are assigned to the same clusters on different layers. This structure of clusters, referred to as hierarchical framework (F), provides various users with a uniform framework to formulate their own graphs.
(2) Construct a personal HG. Based on the shared hierarchical framework F and individual location history ($LocH$), each user can build a personal directed-graph in which a graph node is the cluster containing the user's stay points and a graph edge stands for the sequence of the clusters (geographic regions) being visited by this user. Here, we do not differentiate the diverse paths that a user created between two places (clusters).

Definition 4 (*Hierarchical Framework F*). F is a collection of stay point-based clusters C with a hierarchy structure L. $F = (C, L)$, where $L = \{l_1, l_2, \ldots, l_n\}$ denotes the collection of layers of the hierarchy. $C = \{c_{ij} | 1 \leq i \leq |L|,\ 0 \leq j \leq |C_i|\}$, where c_{ij} represents the jth cluster of stay points on layer $l_i \in L$, and C_i is the collection of clusters on layer l_i.

As illustrated in Figure 8, from the top to the bottom of the hierarchy, the geospatial scale of clusters decreases while the granularity of geographic regions increases from being coarse to being fine. Thus, the hierarchical feature of this framework is useful and essential to differentiate people with different degrees of similarity. The users who share the same location histories on a lower layer might be more correlated than those who share location histories on a higher layer.

Definition 5 (*Hierarchical Graph*). Given a user's location history ($LocH$) and the shared framework (F), the user's hierarchical graph (HG) can be formulated as a set of graphs $HG = \{G_i = (C_i, E_i), 1 < i \leq |L|\}$. On each layer $l_i \in L$, $G_i \in HG$ includes a set of vertexes C_i and the edges E_i connecting $c_{ij} \in C_i$.

Notations. In the rest of this article, we use the following notations to simplify the descriptions. $U = \{u_1, u_2, \ldots, u_n\}$ represents the collection of users in a community, $u_k \in U, 1 \leq k \leq |U|$ denotes the k-th user. $Traj^k$, S^k, $LocH^k$ and HG^k respectively stand for the u_k's GPS trajectories, stay points, location history, and hierarchical graph. s_j^k means the j-th stay point of u_k; G_j^k is u_k's personal graph on layer l_i; seq_j^k denotes u_k's sequence extracted from layer l_i.

2.4. Architecture of the System

Figure 9 gives an overview of the architecture of our recommender system, which consists of three parts: location history representation, user similarity exploration and location recommendation.

First, based on individual GPS trajectories, we build a hierarchical graph for each user using the method we proposed in Section 2.2. This hierarchical graph is capable of modeling the user's location histories on different geospatial scales.

Second, given two users' hierarchical graphs, we are able to match the similar sequences shared by them on each layer of the hierarchy and calculate a similarity score for them. Later, a group of people, called potential friends, with relatively high scores will be retrieved for a particular individual.

Third, using a POI database, we understand the profile of a geospatial region by exploring the categories of POIs located in the region. Such profiles enable us to detect the similarity between geospatial regions and recommend locations based on users' diverse requests. At the same time, with the similarity between locations, we are able

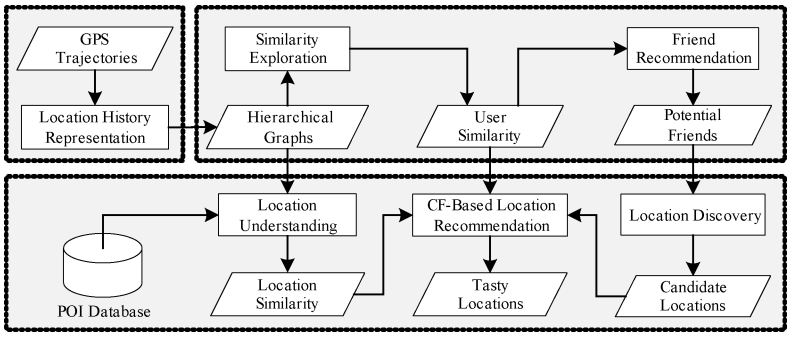


Fig. 9. Architecture of our recommender system.

Algorithm LocHisRepresent($\varphi, \theta_d, \theta_t$)

Input: The collection of users' GPS trajectories: $\varphi = \{Traj^k, 1 \le k \le |U|\}$.
Output: The collection of each user's hierarchical graph: $\varepsilon = \{HG^k, 1 \le k \le |U|\}$
1. SP=$\emptyset$;
2. **Foreach** $u_k \in U$ **do**
3. S^k= StayPointDetection($Traj^k, \theta_d, \theta_t$);
4. SP. Add(S^k);
5. F= HierarchicalClustering (SP);
6. **Foreach** $u_k \in U$ **do**
7. HG^k = GraphBuilding($F, S^k, Traj^k$);
8. ε. Add(HG^k);
9. **Return** ε;

Fig. 10. The procedures of location history representation.

to conduct a hybrid recommender based on a collaborative filtering and the content-based method. By employing the hybrid recommender, we estimate a particular user's interests in the geographic regions which appeared in his/her potential friends' location histories but have not been found by the user.

Finally, the top_N geographic regions, which are most likely to attract the user, are recommended and shown on a Web map.

2.4.1. Location History Representation. Figure 10 describes the major processes of modeling a user's location history based on his/her GPS trajectories. First, given the collection of all users' GPS, trajectories, $\varphi = \{Traj^k, 1 \le k \le |U|\}$ for each user, we extract stay points from each individual's trajectories using the algorithm described in Figure 7 and then put these stay points together into a dataset $SP = \{S^k, 1 \le k \le |U|\}$. Later, using a density-based clustering algorithm, this dataset SP will be hierarchically clustered into several geospatial regions C in a divisive manner. Thus, the similar stay points from various users will be assigned to the same clusters on different layers. Third, based on the shared hierarchical framework (F) and individual trajectory, a personal hierarchical graph is built for each user to model his/her location history on different geospatial scales.

2.4.2. User Similarity Explorations. Figure 11 briefly shows the procedures of the user similarity exploration. First, given the hierarchical graphs HG^p and HG^q of two users (u_p and u_q), we search for the same graph nodes shared by these two users on each layer of the hierarchy. Later, from each layer $l_i \in L$, two sequences (seq_i^p and seq_i^q)

Algorithm SimilarityExplorer (HG^p, HG^q)

Input: hierarchical graphs HG^p and HG^q of two users, u_p and u_q
Output: the similarity score $ss^{p,q}$ between u_p and u_q
1. $ss_{pq} = 0$, $Q^{p,q} = \emptyset$, i=1;
2. **While** $i \leq |L|$ **do**
3. (seq_i^p, seq_i^q)=SequenceExtraction($HG^p.G_i, HG^p.G_i$);
4. $sseq_i^{p,q}$=SequenceMatching(seq_i^p, seq_i^q);
5. $Q^{p,q} = Q^{p,q} \cup sseq_i^{p,q}$; //the collection of similar sequences
6. $ss^{p,q}$ = SimilarityMeasure($Q^{p,q}$); //calculate similarity score
7. **Return** $ss^{p,q}$;

Fig. 11. Explore similarity between users based on individual hierarchical graphs.

containing such graph nodes will be respectively retrieved for u_p and u_q. Second, we can find a set of similar subsequences ($sseq_i^{p,q}$) from the given sequence pairs (seq_i^p and seq_i^q). Here, a similar sequence stands for two individuals sharing the property of visiting the same sequence of places with similar time intervals. Third, based on the retrieved similar sequences, we calculate for the pair of users a similarity score ($ss^{p,q}$) considering the following three common sense knowledge.

—*Sequence*. The longer the similar sequence of visitation shared by two users, the more similar the two users might be.
—*Hierarchy*. The finer the granularity of geographic regions shared by two individuals, the more similar these two individuals might be.
—*IDF*. Two users visiting a geospatial region accessed by a few people might be more correlated than others sharing a location history accessed by many people.

Consequently, we endow location sequences of different lengths with different significances. The longer a similar sequence is, the higher score this sequence can obtain. At the same time, the lower the layer a similar sequence was found, the higher similarity score the sequence obtains. (Refer to Section 3 for more details.)

2.4.3. Location Recommendation. Figure 12 depicts the major procedures of the location recommendation. First, given a user (u_k) as a query, we can rank other people in a community ($u_j \in U$) according to their similarity score ($ss^{k,j}$) to u_k. Then, a group of people (U') with relatively high similarity scores can be retrieved as potential friends for u_k. Second, from the location histories (LH) of u_k's potential friends, we are more likely to discover some geospatial regions (Loc) that might interest u_k but have not been found by u_k. Third, using our HGSM, a CF-based method is employed to infer the individual's interest in the geospatial regions that u_k has not visited previously. Consequently, the top-N geospatial regions with relatively high predicted ratings are recommended to u_k.

In addition, we understand the profile of a geographical region by exploring the categories of POIs within it. So far, four categories, consisting of "restaurants," "entertainments," "sports," and "travel," are exploited in our system. With this, we are able to find geospatial regions with similar profiles which enable us to integrate the content-based method into collaborative filtering. This approach can reduce the cold start problem of our systems. Also, such profiles endow us with the ability of filtering some unwanted locations like people's homes. Moreover, it allows us to recommend various types of locations based on users' requests on different occasions. In other words, we can recommend the regions containing some landscapes to a user if they wants

Algorithm Recommender (u_k, U, LH, SM)

Input: a user u_k, user collection U, each user's location history $LH = \{LocH^j, 1 \le j \le |U|\}$, Similarity matrix $SM = \{ss^{k,j}, 1 \le k \le |U|, 1 \le j \le |U|, j \ne k\}$.

Output: $Loc[L][N]$ Top n Locations matching u_k's tastes on each layer of hierarchy

1. $Loc[L][N] = \emptyset$;
2. v =GetSimilarityVector(u_k, SM); // $v = \left(ss^{k,1}, ss^{k,2}, \ldots, ss^{k,j}\right), 1 \le j \le |U|, j \ne k$
3. U'=FriendRecommender(u_k, v); // $U' \subset U, \forall u_j \in U', u_p \in \overline{U'}, ss^{k,j} > ss^{k,p}$.
4. **Foreach** layer $l_i \in L$ **do**
5. $tempLoc = \emptyset$;
6. **Foreach** user $u_j \in U'$ **do**
7. $locals$=LocationDiscovery(G_i^k, G_i^j); // G_i^k is u_k's graph on layer l_i
8. // $locals = \{c_{im} | c_{im} \in G_i^j.C_i \wedge c_{im} \notin G_i^k.C_i\}$;
9. $tempLoc$= $tempLoc \cup locals$; // insert $locals$ to the collection
10. CF-BasedInference($tempLoc$, v, LH);
11. //maintain top-N geospatial regions with high ratings
12. $tempLoc$ =Top_N($tempLoc$);
13. $x = 0$;
14. **Foreach** region $c_{im} \in tempLoc$ **do**
15 $Loc[i][x] = c_{im}$; $x++$; //c_{im} is the m-th cluster on layer l_i
16. **Return** Loc;

Fig. 12. Major procedures of location recommendation.

to get some suggestions for journey planning. Likewise, the regions containing some shopping malls or cinemas could be recommended when a user prefers to find a place for entertainment (refer to Section 4 for details).

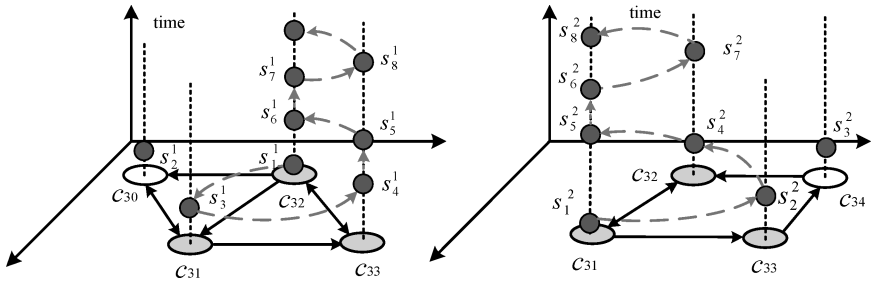
3. USER SIMILARITY EXPLORATION

In this section, we detail the processes of user similarity exploration: location history extraction, sequence matching, and similarity measurement.

3.1. Location History Extraction

The hierarchical graph (HG) offers an effective representation of a user's location history ($LocH$) which implies a sequence property of the user's movements on geographic spaces of different scales. Given HG^1 and HG^2 of two individuals (u_1 and u_2), we first find the same graph vertexes ($V_i^{1,2}$) shared by the two individuals on each layer ($l_i \in L$) where $V_i^{1,2} = \{c_{ij} | c_{ij} \in HG^1.C_i \cap HG^2.C_i)\}, 1 \le i \le |L|$. Then, on each $l_i \in L$, a sequence is respectively formulated for u_1 and u_2 based on $V_i^{1,2}$. Later, measuring the similarity between two users can be transformed into a problem of sequences matching.

Following the example shown in Figure 8, Figure 13 demonstrates how a sequence of places is extracted from each layer of an individual's HG. As we can see, u_1 and u_2 share the same graph vertexes, $V_3^{1,2} = \{c_{31}, c_{32}, c_{33}\}$, on the third layer of HG^1 and HG^2. Over these vertices, a list of green nodes linked by a dash line denotes the stay points that the users generated in the corresponding clusters. This list can be obtained by ranking the user's stay points in each cluster by timestamps. Using a brown curve, we can sequentially connect these green nodes over the shared graph vertices in terms of time serials. Therefore, on the 3rd layer of the hierarchy, a sequence



(a) G_3^1 : u_1's graph on the 3rd layer (b) G_3^2 : u_2's graph on the 3rd layer

Fig. 13. Sequence representation based on users' hierarchical graph.

$seq_3^1 = c_{32} \rightarrow c_{31} \rightarrow c_{33} \rightarrow c_{33} \rightarrow c_{32} \rightarrow c_{32} \rightarrow c_{33} \rightarrow c_{32}$ is generated for u_1, and a sequence $seq_3^2 = c_{31} \rightarrow c_{33} \rightarrow c_{32} \rightarrow c_{31} \rightarrow c_{31} \rightarrow c_{32} \rightarrow c_{31}$ is created for u_2. Here, seq_i^k denotes the sequence of u_k on the ith layer of HG^k. For simplicity we represent these sequences as follows:

$$seq_3^1 = c_{32}(1) \rightarrow c_{31}(1) \rightarrow c_{33}(2) \rightarrow c_{32}(2) \rightarrow c_{33}(1) \rightarrow c_{32}(1),$$
$$seq_3^2 = c_{31}(1) \rightarrow c_{33}(1) \rightarrow c_{32}(1) \rightarrow c_{31}(2) \rightarrow c_{32}(1) \rightarrow c_{31}(1),$$

where the number following a graph vertex stands for the occurrences that the user successively travels in the corresponding cluster. Given the information of each stay point, we are able to calculate the time interval (Δt_i) between consecutive items of these sequences. For instance, in seq_3^1, $\Delta t_1 = s_3^1.arvT - s_1^1.levT$ and $\Delta t_3 = s_6^1.arvT - s_5^1.levT$. Thus, the two sequences can be represented as follows:

$$seq_3^1 = c_{32}(1) \overset{\Delta t_1}{\rightarrow} c_{31}(1) \overset{\Delta t_2}{\rightarrow} c_{33}(2) \overset{\Delta t_3}{\rightarrow} c_{32}(2) \overset{\Delta t_4}{\rightarrow} c_{33}(1) \overset{\Delta t_5}{\rightarrow} c_{32}(1)$$
$$seq_3^2 = c_{31}(1) \overset{\Delta t_{1'}}{\rightarrow} c_{33}(1) \overset{\Delta t_{2'}}{\rightarrow} c_{32}(1) \overset{\Delta t_{3'}}{\rightarrow} c_{31}(2) \overset{\Delta t_{4'}}{\rightarrow} c_{32}(1) \overset{\Delta t_{5'}}{\rightarrow} c_{31}(1).$$

Here, two users' location histories become comparable because we use cluster ID rather than stay point ID to represent the items of a sequence. These clusters are derived from all users' datasets and shared by all the users.

3.2. Sequence Matching

3.2.1. Concepts of Similar Sequences

Definition 6 (*Similar Sequence*). Generally, a similar sequence stands for two users, u_p and u_q, sharing the property of visiting the same sequence of places with similar time intervals. Formally, a pair of sequences, seq_i^p and seq_i^q,

$$seq_i^p = \langle a_1(m_1) \overset{\Delta t_1}{\rightarrow} a_2(m_2) \overset{\Delta t_2}{\rightarrow} \ldots \overset{\Delta t_{j-1}}{\rightarrow} a_j(m_j) \overset{\Delta t_j}{\rightarrow} \ldots \overset{\Delta t_{n-1}}{\rightarrow} a_n(m_n) \rangle,$$
$$seq_i^q = \langle b_1(m'_1) \overset{\Delta t'_1}{\rightarrow} b_2(m'_2) \overset{\Delta t'_2}{\rightarrow} \ldots \overset{\Delta t'_{j-1}}{\rightarrow} b_j(m'_j) \overset{\Delta t_j{}'}{\rightarrow} \ldots \overset{\Delta t'_{n-1}}{\rightarrow} b_n(m'_n) \rangle,$$

where $a_j \in V_i^{pq}$ is a cluster ID and V_i^{pq} are the graph vertices shared by u_p and u_q on layer l_i. m_i represents the times the user successively visits cluster a_j, and Δt_j stands for the transition time the user traveled from cluster a_j to a_{j+1}.

seq_i^p and seq_i^q are similar if and only if they satisfy the following conditions.

(1) $\forall 1 \leq j \leq n$, $a_j = b_j$, (i.e., the nodes at the same position of the two sequences share the same cluster ID);

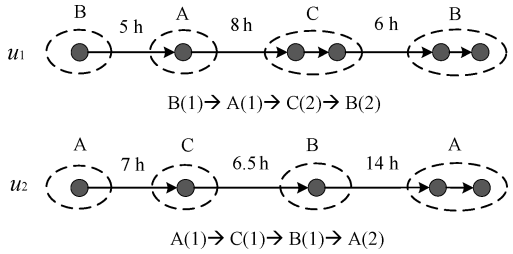


Fig. 14. An example of similar sequences.

(2) $\forall 1 \leq j < n$, $\frac{|\Delta t_j - \Delta t'_j|}{\max(\Delta t_j, Dt'_j)} \leq p$, where p is a predefined ratio threshold, called *temporal constraint*. It denotes that the two users have similar transition times between the same regions.

If both conditions hold, a similar sequence, $sseq_i^{p,q}$, contained in seq_i^p and seq_i^p is retrieved as follows.

$$sseq_i^{p,q} = \left\langle a_1(min(m_1, m'_1)) \rightarrow a_2(min(m_2, m'_2)) \rightarrow \cdots a_n(min)(m_n, m'_n) \right\rangle,$$

where $\min(m_1, m'_1)$ denotes the minimum value between m_1 and m'_1.

Definition 7 (*n-Length Similar Sequence*). If the number of nodes in similar sequences is n, we call this sequence the *n-length* similar sequence.

Definition 8 (*The Maximal-Length Similar Sequence*). The *maximal-length* similar sequence stands for the sequence that is not contained in any other similar sequences.

Using part of the two sequences (seq_3^1 and seq_3^2) depicted in Figure 13, Figure 14 further illustrates the definitions mentioned in this Section. For simplicity's sake, we use A, B and C to respectively represent the cluster c_{31}, c_{32} and c_{33}. As a result, the first four items of seq_3^1 and seq_3^2 can be represented as

$$\left\langle B(1) \overset{5h}{\rightarrow} \text{A}(2) \overset{8h}{\rightarrow} \text{C}(2) \overset{6h}{\rightarrow} \text{B}(2) \right\rangle \quad \text{and} \quad \left\langle A(1) \overset{7h}{\rightarrow} \text{C}(2) \overset{6.5h}{\rightarrow} \text{B}(2) \overset{10h}{\rightarrow} \text{A}(2) \right\rangle.$$

The values of transition time between adjacent clusters are assumed for demonstration.

Clearly, the time interval between the two users' transition time from A to C is 1 hour (8–7 = 1 h). If the temporal constraint is configured as 0.2, a 2-length similar sequence, A(1) → C(1), is detected ($1/8 = 0.125 < 0.2$) from u_1 and u_2's location histories. Likewise, other two 2-length similar sequences C(1) → B(1) and A(1) → B(1) can be retrieved, as they also hold the conditions mentioned in definition 6. However, these 2-length similar sequences are not maximal-length similar sequences as they are contained in the 3-length similar sequence A(1) → C(1) → B(1). Therefore, what we aim to detect from two users' location histories are the maximal-length similar sequences like A(1) → C(1) → B(1). Although u_1 and u_2 share the same order of visiting B and A, the temporal constraint cannot be satisfied. This phenomenon usually implies that u_2 visits too many other places before reaching region A. Hence, the sequence property of the user's movements would be reduced tremendously.

3.2.2. Algorithm for Similar Sequences Detecting. Although there are already some sequence matching algorithms, it is not proper to employ them directly due to the following two reasons.

First, it is well-known that the computation of existing sequence matching algorithms increases quite fast with the extending length of the sequences to be matched. However,

Algorithm SequenceMatching (seq_i^j, seq_i^k, p)

Input: A sequence pair seq_i^j and seq_i^k from the i-th layer of two users (u_j and u_k)'s hierarchical graph, a temporal constraint threshold p.

Output: A set of similar sequences $sseq_i^{j,k}$ on the i-th layer.

1. $sseq_i^{j,k} = \emptyset$;
2. Seq^j=Split(seq_i^j); //partition a long sequence into several short sequences
3. Seq^k= Split(seq_i^k);
4. **Foreach** sequence $sq^j \in Seq^j$ **do**
5. **Foreach** sequence $sq^k \in Seq^k$ **do**
6. $sseq$ =SearchSimilarSeq(sq^j, sq^k, p);
7. $sseq_i^{j,k} = sseq_i^{j,k} \cup sseq$;
8. **return** $sseq_i^{j,k}$;

Fig. 15. Major procedures of sequence matching.

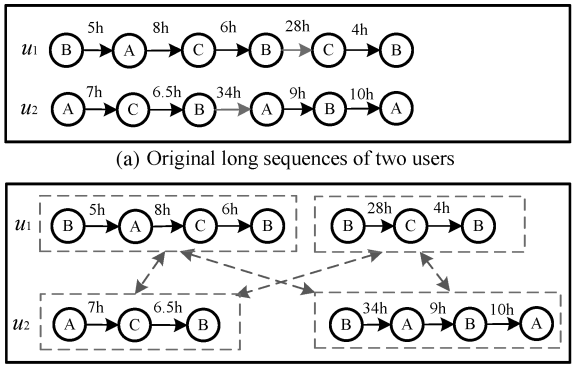
a user could reach over 200 locations in a quarter (including duplicated places like home and company). The computation will be extremely heavy, therefore, we cannot measure the similarity between users very quickly. (Imaginge we have 1 million users in the system.)

Second, we do not expect (and it is not necessary) to retrieve all the similar sequences from a given sequence pair. What we aim to do is to search users' location histories for enough similar sequences which can differentiate these users with different degrees of similarity. Initially, we observe that the number of similar sequences with a relatively long length is extremely small. In short, the probability of sharing a long sequence of movements between two individuals is very small. Then we observe that the excessively long time interval between two nodes of a sequence implies that the user has visited several other places before reaching the next node (i.e., the sequential property of the user's movements between these two regions has been reduced greatly). Thus, we are able (and need) to split long sequences into several short subsets for better efficiency, while maintaining the effectiveness of our method.

As a result, we employ a split policy to partition each user's sequences into several subsets, each of which has a relatively short length. Then, we match these subsequences very efficiently against one another and merge the search results into a collection. Using a sequence pair, seq_i^j and seq_i^k, of two users, u_j and u_k, Figure 15 describes the major procedures of sequence matching.

Following the case shown in Figure 13, Figure 16 demonstrates the sequence spliting and match process. Like Figure 16, we still use A, B, and C to represent the clusters c_{31}, c_{32} and c_{33}. As depicted in Figure 16(A), if the transition time between consecutive nodes of a sequence exceeds a certain threshold (t_p), 24 hour in this case, we split the sequence into two parts. Later, we match each subsequence of u_1 against that of u_2.

Figure 17 shows the algorithm we implemented to detect similar sequences from given sequence pairs. Two operations, sequence extension and sequence pruning, are involved in this process. In the extension operation, we aim to extend each *m-length* similar sequence to one of $(m+1)$-*length*. This operation starts with finding a 1-length similar sequence. Subsequently, in the pruning operation, we pick out the *maximum-length* similar sequence from the candidates generated by the extension operation and remove the rest. Basically, the extension and pruning operations would be implemented alternatively and iteratively until each node in the sequence is scanned.



(a) Original long sequences of two users

(b) Split long sequences into subsets and match each pair, respectively

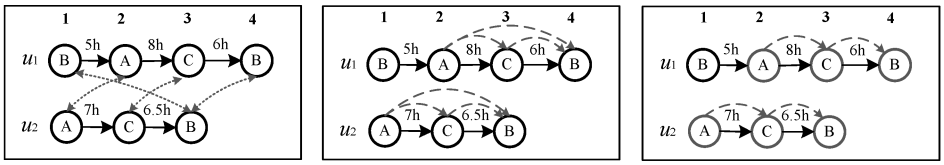
Fig. 16. Split a long sequence into subsequences by the time interval between consecutive nodes.

Algorithm SearchSimilarSeq (seq_1, seq_2, p)

Input: A sequence pair seq_1 and seq_2 of two users, and a temporal constraint threshold p.
Output: A set of *maximum-length* similar sequences $sseq$.
Local Variable: *step*

1. Add 1-*length* sequence into $sseq$, *step* := 1
2. **Foreach** *step-length* sequence $seq \in sseq$ **do**
   // Extend a *step-length* sequence to a (*step*+1)-*length one*
4. G = ExtendSequence(seq, p);
5. $sseq = sseq \cup G$;
6. PruneSequence($sseq$); //Prune non-maximum sequences
7. *step* := *step* +1;
8. **return** $sseq$;

Fig. 17. Detecting similar sequences from a given sequence pair.



(a) Finding 1-*length* similar sequences (b) Finding 2-*length* similar sequences (c) Finding *3-length* similar sequences

Fig. 18. Demonstration of sequence matching.

Using a subsequence pair illustrated in Figure 16(b), Figure 18 demonstrates the algorithm presented in Figure 17. Here, the figures on the top of each box stand for the index of each node in a sequence. First, as shown in Figure 18(a), we detect the 1-*length* similar sequences as follows. $\langle B_{13} \rangle$, $\langle A_{21} \rangle$, $\langle C_{32} \rangle$ and $\langle B_{43} \rangle$, where the subscript of each character represents the index of the matched node in each sequence. For instance, $\langle B_{13} \rangle$ denotes the the first node of sequence 1 sharing the same node B with the third node of sequence 2. Such indexes can help us differentiate nodes of the same cluster ID being visited by users at different times. Second, Figure 18(b) depicts the process of the extension operation based on the results of the first step. If we

Algorithm SimilarityMeasure ($Q^{p,q}$)

Input: The collection of similar sequences shared by two users u_p and u_q
Output: A similarity score $ss^{p,q}$ denoting the correlation between u_p and u_q.

1. $ss^{p,q}$=0; //overall similarity score
2. **Foreach** layer $l_i \in L$ **do**
3. $ss_l = 0$; //similarity score on a layer
4. $\alpha = f(i)$; // α is a i-dependent factor
5. **Foreach** sequence $seq \in sseq_i^{p,q}$ **do**
6. $ss_q = 0$; //similarity score of a sequence
7. *len*=GetSequenceLength(*seq*); //the number of nodes in *seq*
8. $\beta = f'(len)$; //β is a *len*-dependent factor
9. **Foreach** cluster c_{ij} in *seq* **do**
10. $IDF_{ij} = log\frac{|U|}{n_{ij}}$; //IDF of cluster c_{ij}, n_{ij} is number of users visited c_{ij}
11. $ss_q = ss_q + IDF_{ij} \times \min_{p,q}(m_p, m_q)$;
12. //m_p is the times u_p successively accessed c_{ij}(refer to Definition 6)
13. $ss_l = ss_l + \beta \times ss_q$; // sum up the score of each similar sequence on a layer
14. $ss^{p,q} = ss^{p,q} + \alpha \times ss_l$; //sum up the score of each layer
15. $ss^{p,q} = \frac{ss^{p,q}}{|S^p| \times |S^q|}$; //normalization. $|S^p|$ denotes the number of stay points of u_p
16. **Return** *sseq*;

Fig. 19. Similarity measurement based on retrieved similar sequences.

set the *temporal constraint* p to 0.2, three *2-length* similar sequences $\langle A_{21} \rightarrow C_{32}\rangle$, $\langle A_{21} \rightarrow B_{43}\rangle$ and $\langle C_{32} \rightarrow B_{43}\rangle$ can be retrieved. Then, in the pruning operation, all the 1-*length* sequences will be removed from the similar sequence set (*sseq*) because they are contained in the 2-*lenght* sequences. Third, based on the 2-*length* sequences, one 3-*length* similar sequence, $\langle A_{21} \rightarrow C_{32} \rightarrow B_{43}\rangle$, can be detected. Subsequently, in the pruning operation, all *2-length* similar sequences will be removed from *sseq* as they are subsets of the retrieved 3-*length* similar sequence.

3.3. Similarity Measurement

Figure 19 describes the process of calculating a similarity score between two users. Using the sequence matching method we introduced previously, a collection of similar sequences, $Q^{p,q} = \{sseq_i^{p,q}, 1 \leq i \leq |L|\}$, can be retrieved from u_p and u_q's location histories. When calculating the score, we take into account three factors: (1) the visited popularity of a place contained in a similar sequence, (2) the length of a similar sequence, and (3) the layer on which the sequence was found.

First, we calculate the score (ss_l) that two users (u_p and u_q) obtain on a certain layer by adding up the score (ss_q) of each similar sequence found on this layer. Then, the score (ss_l) of each layer will be weighted and summed up to a final score ($ss^{p,q}$).

3.3.1. Inverse Document Frequency. When measuring the similarity score of a given similar sequence, we involve inverse document frequency (IDF) to differentiate the visited popularity of each geospatial region (cluster of stay points) contained in the sequence. Here, a cluster can be regarded as a document, while the users who have visited this cluster can be deemed as terms. If the number of users (n_{ij}) that visited a region (c_{ij}), is very large, the $IDF_{ij} = log\frac{|U|}{n_{ij}}$ of this region would become very small. Therefore, this region will not offer many contributions to the similarity score of these two users. Intuitively, the phenomenon that lots of people visited a geospatial region like the Great Wall might not mean all these people are similar to one another. The real reason might

be that this region has a famous reputation which attracts a variety of users. However, if two individuals share a location history which is not that well known (and hence, is not accessed by so many people), the individuals might share some similar interests indeed.

3.3.2. Length of a Similar Sequence. Further, we add up the score of each node in a sequence to calculate the similarity score (ss_q) of the sequence. The score of a node is a multiplication of two parts ($IDF_{ij} \times \min(m_p, m_q)$): the IDF of this region, as mentioned previously, and the times ($\min(m_p, m_q)$) the two users successively accessed this region in this sequence (refer to Definition 6). This paradigm looks like the TF (term frequency)-IDF policy in document retrieval. In addition, a length-dependent factor (β) is involved to distinguish the importantce of similar sequences with various lengths (len). For instance, we use $\beta = 2^{len-1}$ in our experiment. In other words, the longer the sequence matched between two users' location histories, the more related these two users might be; hence, a high score should be awarded to this sequence.

3.3.3. The Hierarchy of the Geospatial Scale. By summing up the similarity score (ss_q) of each sequence on a layer, we can calculate the similarity score (ss_l) of the two users on the layer. A layer-dependent factor (a) is involved to weight the significance of sequences found on different layers. For instance, we use $a = 2^{i-1}$ in our experiment. In other words, people that share a sequence of places on a lower layer (with finer granularity) might be more related than others who share a sequence of places on a higher layer (with coarse granularity). Later, we sum up the ss_l of each layer to achieve the overall similarity score ($ss^{p,q}$) of two users.

Finally, to provide a fair result to the users with various scales of GPS trajectories, we divide the overall similarity score ($ss^{p,q}$) by the multiplication of the scales of their dataset ($|S^p| \times |S^p|$). Intuitively, people joining in a Web community earlier are more likely to accumulate more GPS trajectories than new users. If we do not consider the scale of data, more similar sequences would be retrieved from these people's relatively large datasets. Therefore, these senior users might always be recommended to others, althought they are not the most perfect candicates.

4. LOCATION RECOMMENDATION

In this section, we first introduce how we discover a collection of geospatial regions for a particular user. Second, a collaborative filtering-based method is employed to infer the user's interests on these discovered regions. Third, using a POI database, we understand the properties of a region. Hence, we can recommend different types of locations matching the user's preferences at various occasions. In addition, by understanding the properties of a region, we can find the similar regions based on their profiles (contents). Therefore, we are able to reduce the well-known cold start problem of a recommender system by combining the content-based approach with collaborative filtering.

4.1. Location Discovering

Using the approach proposed in Section 3, we are able to measure the similarity ($ss^{k,j}$) between two users (u_k and u_j), and formulate a similarity matrix SM, where $SM = \{ss^{k,j}, 1 \le k \le |U|, 1 \le j \le |U|, j \ne k\}$. Given u_k as a query, we can retrieve from the SM the vector v^k containing the similarity scores between u_k and others, where $v^k = \langle ss^{k,j}, 1 \le j \le |U|, j \ne k \rangle$. Then, $ss^{k,j}$ will be normalized to a value falling into [0, 1] by Equation (3)

$$ss^{k,j} = \frac{ss^{k,j} - \min(v^k)}{Max(v^k) - \min(v^k)}. \tag{3}$$

Later, the top_N users with relatively high $ss^{k,j}$ will be retrieved as u_k's potential friends (U'). On each layer $l_i \in L$, we retrieve for u_k a set of regions R_i^k that are accessed by u_k's potential friends but not visited by u_k. Here, $R_i^k = \{c \in C_i | r_c^k = \emptyset \wedge \exists u_j \in U', r_c^j \neq \emptyset\}$, $1 \leq i \leq |L|$, r_c^k represents u_k's accesses (ratings) on geospatial region c. Here, a particular user's occurrences in a geospatial region are used as implicit ratings of this user on the region.

4.2. CF-Based Inference

Given the geospatial regions discovered for a particular user, a collaborative filtering based method is employed to infer the user's interests on these regions. Equation (4), (5), and (6) describe the process predicting u_k's rating (r_c^k) on a location c.

As shown in Equation (4), the similarity between users u_k and u_j, $sim(u_k, u_j)$, is essentially a distance measure and is used as a weight, i.e., the more similar u_k and u_j are, the more weight r_c^j will carry in the prediction of r_c^k. Here, $sim(u_k, u_j)$ is calculated using HGSM. However, different people may visit places with various times (e.g., a user would visit a park twice while another person may access the same park four times, although both of them are strongly interested in this park), i.e., they use the rating scale differently. Therefore, an adjusted weighted sum is used here.

First, instead of using the absolute values of ratings, we use their deviations from the average rating of the corresponding user. That is $r_c^j - \overline{r^j}$, where $\overline{r^j}$ denotes the average rating of u_j. Second, a normalizing factor d is involved. Here, d can be calculated as Equation (5), where U' is the collection of the users who are the most similar to u_k. Third, we consider u_k' rating scale by calculating the average rating ($\overline{r^k}$) of u_k as Equation (6), where C' represents the collection of locations accessed by u_k.

Actually, the equations shown here illustrate a well-known method [Adomavicius and Tvzhhilin 2006; Nakamura and Abe 1998], which has been used widely in many recommender systems. Hence, we do not explain them in more detail.

$$r_c^k = \overline{r^k} + d \sum_{u_j \in U'} sim(u_k, u_j) \times \left(r_c^j - \overline{r^j}\right); \tag{4}$$

$$d = \frac{1}{|U'|} \sum_{u_j \in U'} sim(u_k, u_j); \tag{5}$$

$$\overline{r^k} = \frac{1}{|c'|} \sum_{c \in C'} r_c^k, C' = \left\{c \in C_i | r_c^k \neq \emptyset\right\}; \tag{6}$$

4.3. Location Understanding

Beside using collaborative filtering to infer a particular user's interests on locations, offline we understand the profiles of a geospatial region by exploring the categories of the POIs within the region. Actually, what attracts people is not the region itself but the POIs (contents), located in the region, such as shopping malls, restaurants and cinemas. The motivations of this step lie in the following three parts.

(1) We aim to differentiate locations with different profiles, which enable the location recommendation based on users' requests on different occasions. Sometimes, an individual prefers to get suggestions related to restaurants when he/she is looking for a place for dinner, while on other occasions the individual might want to view some recommendations about entertainments before going shopping.
(2) We can filter some regions, which might not be useful or attractive to individuals. For instance, a region only covering people's homes should not be recommended to

users. However, it might appear in the discovered locations when many users from a community live closely in the real world.

(3) By understanding the profile of a geospatial region, we are able to combine the content-based method with collaborative filtering to reduce the cold start problem of recommender systems.

In our system, we investigate four categories of POIs: restaurants (R), entertainments (E), sports (S) and travels (T). Here, the entertainments include POIs of shopping malls, cinemas, cafés and bars, etc. Then, a vector, $Z = \langle R, E, S, T \rangle$, is formulated for each region we discovered from users' datasets. Each item of the vector denotes the number of POIs pertaining to the corresponding category. For instance, $Z = \langle 2, 5, 0, 0 \rangle$ stands for a region that contains two restaurants and five entertainment entities. When a region does not contain any POIs, we regard it as a travel place, that is, $Z = \langle 0, 0, 0, 1 \rangle$. Intuitively, this situation occurs when users exploit new tourist spots in the real world. In fact, a geospatial region usually covers various categories of POIs. Hence, we allow a region to simultaneously hold multiple properties, such as restaurants and entertainments.

With such a vector, Z, we are able to achieve the three objectives mentioned previously. For instance, if a user prefers to get recommendations related to sports, a region, with $Z = \langle 2, 5, 0, 0 \rangle$, should be filtered on the results page (See Figure 2 for a case). Meanwhile, as shown in Equation (7), given two vectors, Z_j and Z_k, of two regions, c_j and c_k, we are able to infer their similarity using the cosine similarity measure.

$$Sim(c_j, c_k) = \frac{(z_j . z_k)}{\|z_j\|_2 \cdot \|z_k\|_2}. \tag{7}$$

Later, the ratings on the similar locations can enable the content-based recommendation which will reduce the new item problem of collaborative filtering. In other words, the users' ratings (accesses) on a geospatial region can be used as estimated tastes of these users on other locations which share similar profiles with the region. Therefore, when a new location is discovered, we are able to obtain enough ratings from multiple users and, hence accurately predict other users' interests in it.

The process of understanding geospatial regions would not take many computations due to the following two reasons. First, this process can be conducted offline; second, the number of locations is constrained and increases very slowly. In short, we can perform this process offline infrequently.

5. EXPERIMENTS

In this section, we first present the experimental settings, including the GPS devices, volunteers, GPS data and some parameters we selected in the experiment. Then, using the GPS trajectories collected by 75 users over a period of one year, we evaluate two aspects of our recommender system, friend recommendation and location recommendation. With regard to the friend recommendation, we compare our HGSM with other measures, including the cosine similarity and Pearson similarity measurements. Regarding the location recommendation, by performing a user study, we compare our approach with the pure item-based collaborative filtering and random recommendation.

5.1. Settings

5.1.1. GPS Devices. Figure 22 shows the GPS devices we chose to collect data. They include stand-alone GPS receivers (Magellan Explorist 210/300, G-Rays 2 and QSTARZ BTQ-1000P) and GPS phones. Except for the Magellan 210/300, these devices are set to receive GPS coordinates every two seconds. Regarding the Magellan devices, we

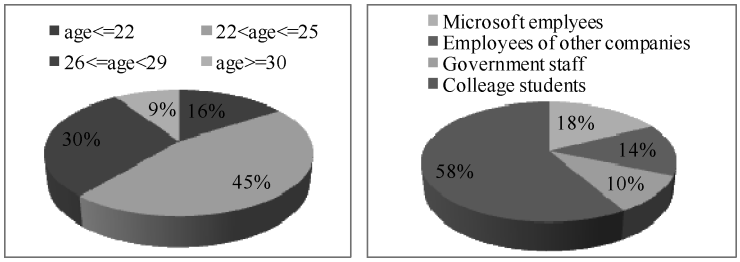
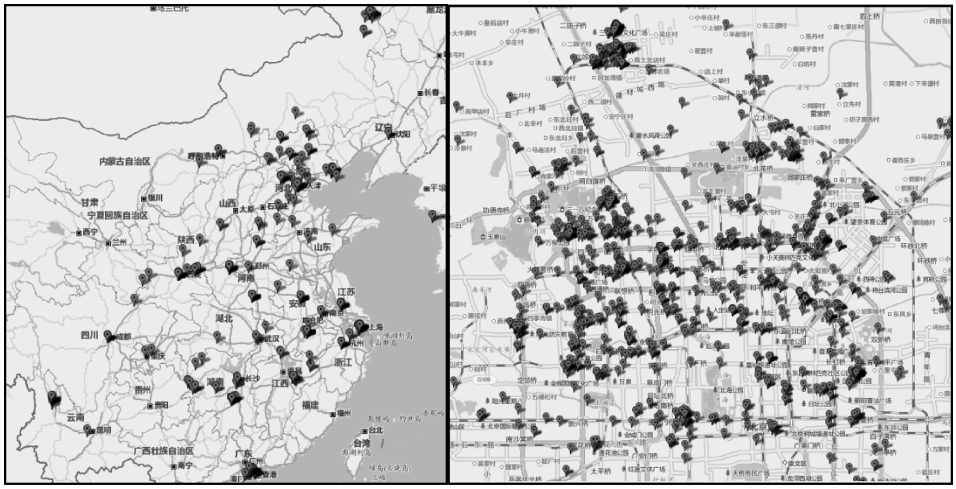


Fig. 20. Demographic statistics of our experiment.



(a) Data distribution in China (b) Data distribution in Beijing

Fig. 21. Distribution of GPS data used in the experiments.

configure their settings to record GPS points as densely as possible because they are not allowed to be configured for recording data by fixed time interval. When an individual changes his/her heading direction or speed to some extent, a GPS point is recorded with such devices.

5.1.2. Volunteers. Carrying a GPS-enabled device mentioned in the 75 users, preceding Section, 41 females and 34 males, recorded their outdoor movements with GPS trajectories over the past year. All of the users are based in China, and most of them live in Beijing. Figure 20 depicts the demographic information of these volunteers. More than half of the volunteers were college students, 18 percent of them were employees of Microsoft, and the rest of them came from a variety of corporations and organizations. Their ages ranged from 19 to 35 years old, and their education background ranged from undergraduate students to Ph. D. holders; the average age of the volunteers was 24. From these volunteers, we can conveniently discover human relationships, such as friends, roommates, lovers, married couples, classmates, colleagues, neighbors, acquaintances and strangers.

5.1.3. GPS Data. Figure 21 depicts the distributions of the stay points extracted from the GPS dataset that we used in the experiment. The dataset covers 36 cities in China and some cities in the USA, South Korea, and Japan. The volunteers are motivated to log their outdoor movements as much as possible by the incentive payments based

Fig. 22. GPS devices used in our experiments.

Table I. The Policies of Incentive Payment for the Data Collection

Policies		Payment	
Payment based on distance of a trajectory		Distance of a trajectory	Price (RMB/KM)
		0 < D < 30KM	0.2
		30 < D < 60KM	0.15
		60 < D < 200KM	0.1
		D > 200KM	0
Payment based on stay points		3RMB per stay point	
Awards	for the top 3 GPS-carrying-rate	300RMB per person	
	for the top 3 effective-time-span	300RMB per person	

Table II. GPS Dataset We Used in the Experiments

Carrying Period (Month)	Number of Users	Number of GPS Points (Million)	Number of Stay Points (K)	Distance (K-KM)
Period <3	20	1.1	1.1	13
3< Period <6	21	1.5	1.6	18
6< Period <12	24	2.8	2.1	36
12< Period	10	1.6	2.8	69
Total	75	7.0	7.6	135.9

on the policy shown in Table I (the exchange rate is 1 US Dollar = 6.88 Chinese RMB at this moment). This incentive policy is composed of three parts: the payment based on the distance of a collected GPS trajectory, the payment based on number of stay points, and the awards. The first two parts are tiny compensation for these volunteers' travel since we do not want to change their behaviors. Regarding the award, the GPS-carrying-rate means the ratio between the number of days in which a volunteer has at least one trajectory and the period of data collection program. The effective-time-span denotes the sum of the time duration of all the trajectories collected by a volunteer. In other words, the more frequently a volunteer has carried a GPS device, the higher these two values are. As a result, the total distance of the GPS trajectories has exceed 135 thousand kilometers, and the total number of GPS points reached almost 7 million (6,963,824).

Table II details the information related to the dataset. Almost half of the volunteers have carried a GPS device over 6 months, and three quarters of them have participated in the data collection activity for a period over 3 months. Using the detection algorithm presented in Figure 7, 7539 stay points have been extracted from the dataset. Considering the privacy issues, we use these datasets anonymously. In short, people cannot build connection between a particular individual and their GPS trajectories.

5.1.4. Parameter Selection

Stay point detection. Regarding the range threshold, θ_d, in most cases, 150 $\sim$ 300 meters could cover the geographical scales of most significant places, such as landmarks, plazas and beaches. On one hand, a too small distance constraint, like 50 meters, will cause many stays to remain undetected or be overdetected (refer to Figure 6). On the

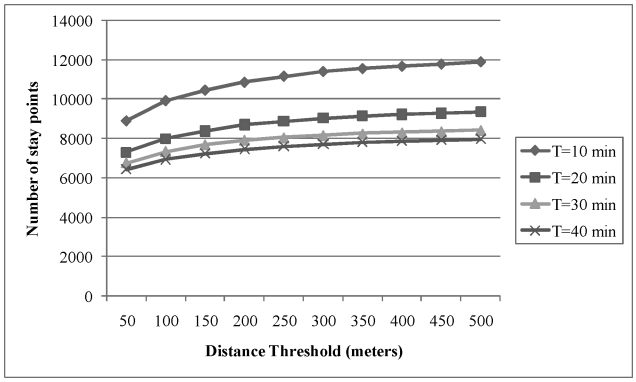


Fig. 23. The number of stay points detected by different time interval and distance thresholds.

other hand, a too large distance constraint, for example, 800 meters, would (1) make the representation of a stay point imprecise (a big region could cover many businesses) and (2) cause multiple stays of a user merged into one stay point. Regarding the time threshold, using a short time interval like 5 minutes, a stay point would be detected at some insignificant places, such as bus stops or cross roads where people wait for the red light. On the contrary, a long time interval, such as 2 hours, would cause some significant stays to remain undetected, e.g., generally, people would not spend too much time in a fast food store or at a plaza.

In our experiment, when detecting stay points from a given GPS trajectory, we set θ_t to 30 minutes and θ_d to 200 meters. In other words, if an individual stays over 30 minutes within a distance of 200 meters, a stay point is detected. Basically, these two parameters are derived from real-world commonsense knowledge. Meanwhile, we tested a set of parameter candidates and observe (from Figure 23) that the numbers of detected stay points are close to each other when $20 \leq \theta_t \leq 40$ minutes and $150 \leq \theta_d \leq 300$ meters. Especially when θ_t is set to 30 minutes, the numbers of stay points do not change too much even if θ_d is configured to different values, 200, 250 and 300 meters.

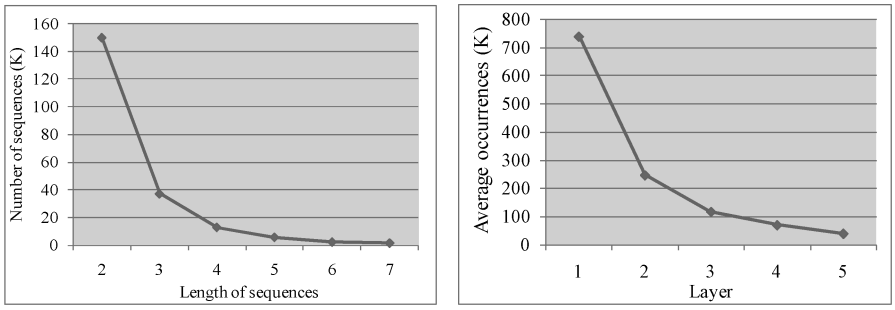
Therefore, (1) selecting $\theta_d = 200$ meters and $\theta_t = 30$ minutes would not affect the final recommendations too much even if they might not be the perfect candidates. (2) These two parameters enable us to find out some significant places, such as restaurants, landscapes and shopping malls, while ignoring the insignificant locations, like the cross road and bus stops.

Clustering. A density-based clustering algorithm called OPTICS [Ankerst et al. 1999] is employed to hierarchically cluster stay-points into geospatial regions in a divisive manner. As compared to an agglomerative method like K-Means, the density-based approach is capable of detecting clusters with irregular structures which may stand for a set of nearby restaurants or travel spots. The clustering operation is conducted iteratively until one of the following conditions hold. (1) The number of users pertaining to a cluster is less than two or (2) the diagonal of a cluster's minimal boundary rectangle (MBR) in geospace is smaller than 500 meters. Taking these parameters, we establish a 5-layer hierarchical structure which provides each user with a consistent framework to build individual hierarchical graphs. Table III details the information of the framework; the top layer is referred to as layer 1 (higher) and the bottom layer is called layer 5 (lower).

Sequence matching. Two parameters are involved in the process of sequence matching; one is the temporal constraint, p, the other is the sequence partition threshold,

Table III. Information of the Shared Framework

Layer	Number of Cluster	Average diagonal of MBR (KM)	Average number of user/cluster	Average number of stay points/cluster
1	1	11450.7	75	7,539
2	29	15.5	10.8	259
3	47	8.3	8.9	158
4	91	2.4	6.5	81
5	150	0.28	5.5	49



(a) Number of sequences changing over length (b) Average occurrences two users sharing one cluster

Fig. 24. Differentiate the significances of similar sequences with different length and on different layers.

t_p. p is used to check whether the transition time of two users is similar, while t_p is employed to partition a long sequence into a set of short subsequences for improving matching efficiency. In the experiments, we test a set of t_p (24 h, 36 h, 48 h) and p (from 0.01 to 0.65 with a step of 0.02), and show the sequence matching results changing over them.

Similarity measurement. To differentiate the significance of similar sequences with different lengths and on different layers, we set $\beta = 2^{len-1}$, and $\alpha = 2^{l-1}$. Here β increases exponentially with the lenght of sequence, *len*, since we observe that the occurrence of *len-length* similar sequences drops exponentially as the *len* increases (See Figure 24(a)). Thus, the significance of an occurrence of a *len-length* similar sequence increases exponentially with *len*. At the same time, as depicted in Figure 21(b), the average occurrence that two users share a cluster on the l-layer drops exponentially as the l increases. Therefore, the significance of similar sequences found on l-layer increases exponentially with l.

Location recommendation. In our experiments, the top 50 similar users are selected as a subject's potential friends considering the following three reasons. First, during the process of the experiment, we observe that more than 20 unvisited locations can be retrieved for a subject from the location histories of the top 50 users that are similar to them. Those are enough for the further location recommendation, as we only provide a subject with the top 10 recommendations in our system. Second, intuitively, the recommended locations would not change even if we involve more similar users' location histories. If a place is not included by the location histories of the top N similar users, the inferred interest of the subject in that place cannot be very high. Hence, the place will not be recommended. 3) From the implementation's perspective, using top N similar users will make the system more efficient and easy to control. Imagining we would have 1 million users in the future; the computation would be extremely high using the whole user set.

Table IV. Examples of Users' Profiles Shown in the Questionnaire

Name	Gender	Affiliation	Living place
Yukun Chen	*male*	*Microsoft Research Asia*	*Tsinghua university, Beijing*
Yechen Hao	*female*	*Deloitte CPA company*	*Xiushuiyuan community, Beijing*
Quannan Li	*male*	*Huazhong university*	*Wuhan, Hubei, China*
...	...	...	...

Table V. Detailed Relevance Settings

Relevance level		Relationships suggestion
4	Strongly similar	Family members, intimate lovers, roommates
3	Similar	Good friends, close colleagues, close classmates
2	Weakly similar	Ordinary friends, acquaintances, neighbors in a community
1	Different	Strangers in the same city
0	Quite different	Strangers in other cities

Based on these friends' location histories, we recommend for each subject the top ten geospatial regions. Meanwhile, if the number of users visiting a location is less than five, the content-based method is used to reduce the cold start problem. The top 5 similar regions to this location are employed to expand the ratings on the location.

5.2. Evaluation Approaches

5.2.1. Evaluation of Friend Recommendation

Ground truth. After the data collection, we present each user with a name list of all the volunteers with a simple user profile, as shown in Table IV. To protect a user's privacy, the information, e.g., living places, is not very detailed. Each user is required to rate other users based on individual understanding and the relevance suggestion shown in Table V. Then, a relation matrix of these volunteers is generated and is used as the ground truth to evaluate the recommendation results for each user. As these volunteers rate each user in a stand-alone manner, the relevance rating between two users might be asymmetric, that is, though user u_1 rates 2 on u_2, u_2 may not rate 2 on u_1.

Of course, we agree that the perfect evaluation should be conducted only using the data from strangers. But, at that moment, it would be even more difficult to get the ground truth of users' relationships. Although this ground truth might not be perfect, it is the best way we can conceive.

In the 75 users, we identified only 1 married couple and 2 pairs of intimate lovers. Typically, they traveled together four times per month (because they live in different parts of Beijing and only meet on weekends). They are the people who are most likely to travel together. Although other users could have friendships, the frequency of traveling together is very rare (because they might live in different parts of a city and have different living routines). In short, the percentage of colocated traces is very small ($<2\%$) which would not affect the validity of the evaluation results very seriously.

Evaluation Framework. As demonstrated in Figure 25, our approach is evaluated as an information retrieval problem, in which 75 people are, respectively, used as queries to search for each of them the top ten similar users. For instance, using u_k as a query, we retrieve the top ten similar users based on their similarity score to u_k. Then, a relevance vector, G, of the search results is formulated based on the relationship matrix. Given the retrieved G and the corresponding ground truth, we calculate *MAP* and *NDCG* for this retrieval. After all the volunteers have been tested, we calculate a mean value of *MAP* and *NDCG* based on each individual's results.

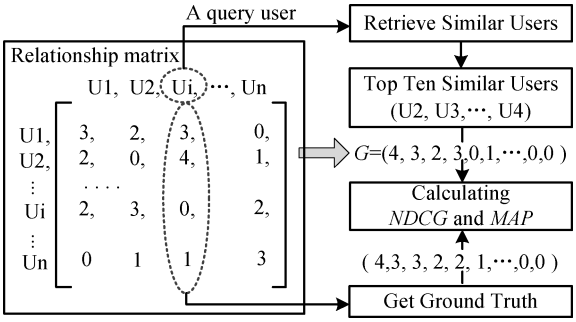


Fig. 25. The framework of evaluation on friend recommendation.

Evaluation Criteria. *MAP* and *NDCG* are employed to evaluate the performance of our approach. *MAP* is the most frequently used summary measure of a ranked retrieval run. In our experiment, it stands for the mean of the precision score after each relevant user is retrieved. In the search results, a user is deemed as a relevant user if his/her relevant level is greater than or equal to 3. For instance, the *MAP* of a relevance vector, $G = \langle 4, 0, 2, 3, 3, 1, 0, 2, 1, 1\rangle$ is computed as follows:

$$MAP = \frac{1 + 2/4 + 3/5}{3} = 0.7.$$

NDCG is used to compute the relative-to-the-ideal performance of information retrieval techniques [Jarvelin et al. 2002]. The discounted cumulative gain of G is computed as follows. (In our experiments, $b = 2$.)

$$DCG[i] = \begin{cases} G[1], & if\, i = 1 \\ DCG[i-1] + G[i], & if\, i < b \\ DCG[i-1] + \dfrac{G[i]}{log_b i}, & if\, i \geq b. \end{cases} \tag{8}$$

Given the ideal discounted cumulative gain, *DCG'*, then *NDCG* at i-th position can be computed as $NDCG[i] = DGC[i]/DCG'[i]$.

Baselines. We compare our HGSM with three baselines; similarity by count, the cosine similarity and Pearson similarity. The first method measures the similarity between two users by counting the regions shared by these two users. It is an intuitive method that most people might conceive. The rest are the cosine similarity and the Pearson similarity measures, which have been widely used in recommendation systems and have been claimed to outperform other existing similarity measures [Spertus et al. 2005]. Suppose $N = |C_i|$ clusters $\{c_{ij} \in C_i,\ 1 \leq j \leq N\}$ are generated on the i-th layer of the shared framework. If in the cluster c_{ij}, u_1 has m_j stay-points and u_z has m'_j stay-points, two vectors can be respectively constructed for u_1 and u_2 as follows,

$$U_1 = \langle m_1, m_2, \ldots m_j, \ldots, m_N\rangle \quad \text{and} \quad U_2 = \langle m'_1, m'_2, \ldots, m'_j, \ldots, m'_N\rangle.$$

The similarity by count is computed as Equation (9):

$$sim_{count}(u_1, u_2) = \sum_{j=0}^{N} \min(m_j, m'_j) \tag{9}$$

Table VI. Detailed Ratings

Ratings	Score	Notations
R1	2	Very interesting
R2	1	Interesting
R3	0	Neural
R4	−1	Boring

The cosine similarity and Pearson similarity are computed as Equation (10) and Equation (11), respectively:

$$sim_{cosine}(u_1, u_2) = \frac{\sum_j m_j m'_j}{\sqrt{\sum_j m_j^2}\sqrt{\sum_j (m'_j)^2}} \tag{10}$$

$$sim_{pearson}(u_1, u_2) = \frac{\sum_j (m_j - \overline{U_1})(m'_j - \overline{U_2})}{\sqrt{\sum_j (m_j - \overline{U_1})^2 \sum_j (m'_j - \overline{U_2})^2}} \tag{11}$$

5.2.2. Evaluation of Location Recommendation

Evaluation Framework. Regarding the evaluation of the location recommendation, we conduct a user study which uses Beijing as a test region, that is, only the locations located in Beijing will be recommended. 30 volunteers, 15 females and 15 males, were invited to participate in the study. They are a subset of the 75 users collecting data for us. Also, each of them has been in Beijing for at least 6 years and has accumulated at least two-months of GPS trajectories. In short, they have rich travel knowledge about Beijing and could be more likely to give a reasonable evaluation on the recommendations than others. In this study, using our prototype system, each volunteer was respectively recommended ten geospatial regions based on their location history (refer to Figure 3 for an example) on a desktop computer. These ten regions may contain different types of POIs, such as businesses, cinemas and restaurants, because no filtering has been performed. Later, the volunteer is requested to offer his/her feedback on each recommended location by giving one of the ratings shown in Table VI.

In order to test the effects of the various scales of GPS trajectories on the recommendation results, we repeat this study three times based on the GPS trajectories collected by users over different periods. In the first run, we study the effectiveness of the location recommendation using these subjects' GPS trajectories for two weeks. In the second run, the subjects' GPS trajectories for one month are used. Finally, their GPS trajectories for three months are employed in the third run of the study.

Evaluation Criteria. After performing a user study, we investigate the following two aspects of the criteria; one is the average score ($\bar{s}$), the other is the percentage of each rating ($P(R_i)$) in the subjects' feedback. These two criteria can be computed as Equations (12) and (13);

$$\bar{s} = \frac{\Sigma_k^N \Sigma_j^m r_j^k . score}{N}, \tag{12}$$

$$P(R_i) = \frac{\Sigma_k^N u_k . |R_i|}{N * m}, \tag{13}$$

where N stands for the number of the volunteers that participated in the study and m denotes the number of locations recommended to a user, here, $N = 30$, $m = 10$. Meanwhile, $r_j^k \cdot score$ represents u_k's rating on location c_j, and $u_k.|R_i|$ means the number of ratings pertaining to R_i in u_k's feedback. In other words, the higher $\bar{s}$ is, the better the location recommendation might be; the higher $P(R_1)$ and $P(R_2)$ are, the better the

recommendation might be. On the contrary, the higher $P(R_3)$ and $P(R_4)$, the worse the recommendation would be.

Baseline. We compare our approach with two baseline methods, pure-item-based recommendation and random recommendation. In the former method, a geospatial region is regarded as an item and the occurrences of an individual on this region are deemed as his/her ratings on the region. Then, by integrating all users' location histories, an item-based collaborative filtering, slope-one algorithm [Lemire et al. 2005] is employed to infer the individual's tastes on the regions he/she does not visit. In contrast to our approach, this method does not take into account the similarity between users to weight the ratings of different users. In the latter baseline, the system will randomly recommend to a particular user a set of geospatial regions the user does not visit.

5.3. Results

5.3.1. Results of Friend Recommendation

Notations. We define some notations shown in the following figures. *Seq* stands for the similarity measure considering only the sequence feature, and *Hier* denotes the measure considering the hierarchy property of geographical spaces. *IDF* means the similarity measure taking into account the factor of visited popularity (inverse document frequency). Thus, *Seq+IDF+Hier* (our HGSM) represents the measure of similarity simultaneously considering the sequence, *IDF* and hierarchy properties. Meanwhile, *Count* means *similarity-by-count* on the bottom layer, and *Hier+Count* represents *similarity-by-count* across multilayers. *Cosine* and *Pearson* respectively denote the cosine similarity and the Pearson similarity on the bottom layer. Likewise, *Hier+Cosine* and *Hier+Pearson*, respectively, represent the cosine similarity and Pearson similarity across multilayers.

Objectives. We focus on investigating the effectiveness of the proposed properties, sequence, hierarchy and IDF, in measuring the similarity between users. In addition, we test the performances of the combination of these features and study their contributions to the baseline measures, such as the cosine and Pearson similarities.

Comparison of Different Measures. Figure 26 and Figure 27, respectively, depict the *MAP*s and *NDCG* of different similarity measures. From the data shown in these figures, we can obtain the following results.

First, in contrast to the baselines, the sequence property is more powerful in modeling users' location histories and differentiating the people with various degrees of correlations.

Second, the IDF feature brings a significant improvement to the measure only based on sequence property. In addition, with this IDF feature, the performances of the baseline methods have also been enhanced.

Third, by taking into account the hierarchy property, we can further improve the performance of our approach. However, this feature is not that useful for the baseline methods, that is, the hierarchy property can achieve a good performance only when it is used together with the sequence property and IDF. Otherwise, it may cause the suspicion of an over-count of the similarity between two users on a lower layer.

Finally, our method, HGSM (*Seq+IDF+Hier*), employing these three features, outperforms other methods in measuring the user similarity.

Sequence Matching of HGSM. To further explore the property of HGSM, Figure 28 and Figure 29, respectively, show the *MAP* and *NDCG*@5 of HGSM changing over the temporal constraint p and the sequence partition threshold t_p. Three t_p candidates,

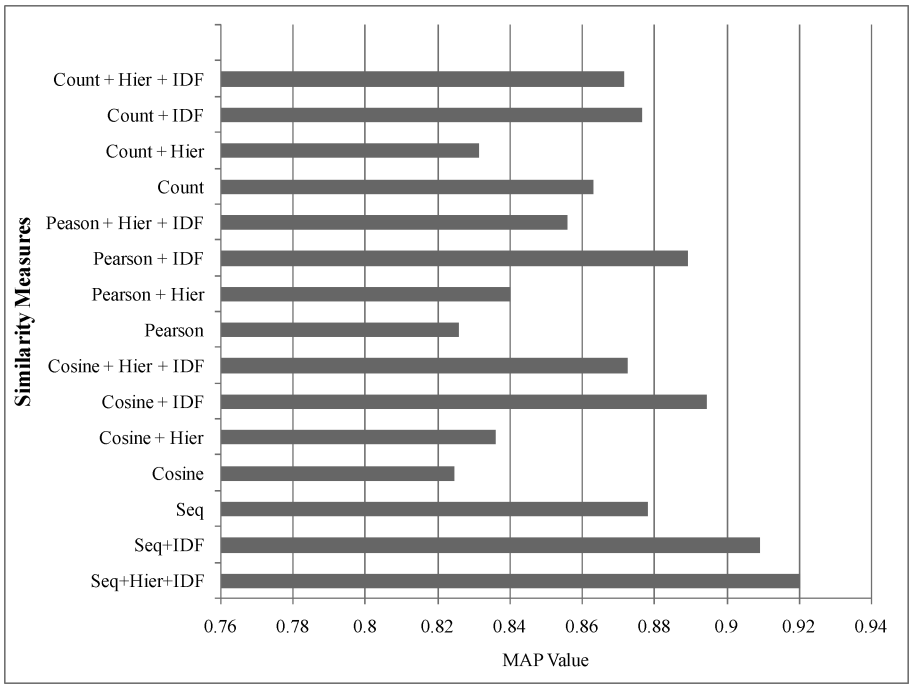


Fig. 26. Comparison of MAP among different similarity measures.

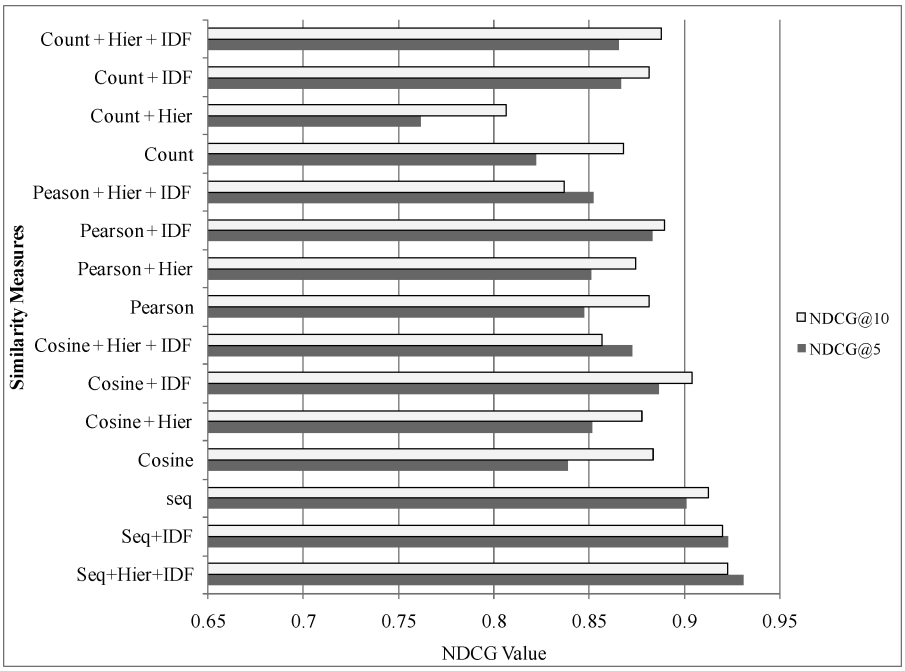


Fig. 27. *NDCG*@5 and *NDCG*@10 of different similarity measures.

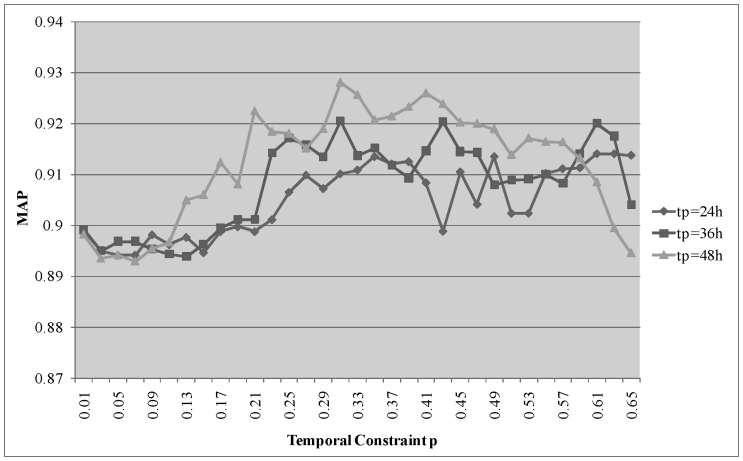


Fig. 28. *MAP* of HGSM changing over the temporal constraint p and partition threshold t_p.

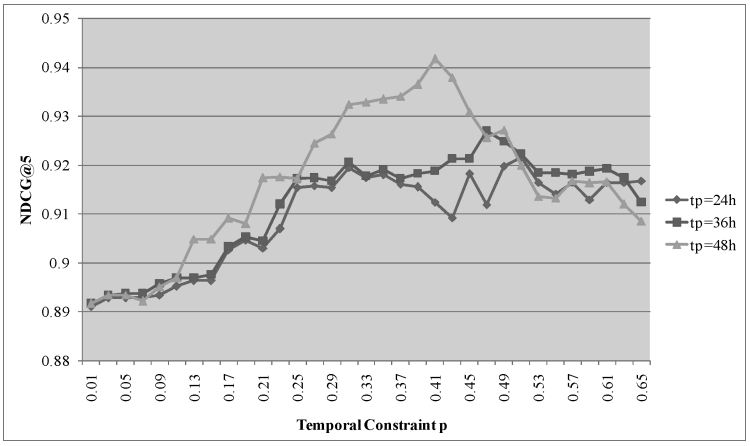


Fig. 29. *NDCG*@5 of HGSM changing over the temporal constraint p and partition threshold t_p.

24 hours, 36 hours, and 48 hours, are tested in the experiment. In other words, if the time interval between consecutive nodes in a sequence exceeds 24 (or 36, or 48) hours, the sequence will be divided into two subsequences for better efficiency of similar sequence matching. As we can see, when p is set to a small value like 0.05, HGSM cannot achieve a good performance. Intuitively, some similar sequences will not be detected from two users' location histories due to the over strict temporal constraint. This will reduce the capability of HGSM in differentiating similar users of various degrees. On the contrary, if the p is configured as a very large value, like 0.65, some sequences, which are not that similar will be regarded as similar ones; hence, the performance drops. Thus, we select $p = 0.41$, $t_p = 48$ hours in our experiments.

Hierarchy of HGSM. Table VII shows the *MAP* and *NDCG* of our approach using only one layer of the proposed framework. As we can see, both *MAP* and *NDCG* increase as the level of layer increases, that is, layer 5 is more capable of discriminating similar users than layer 4, while the approach considering the hierarchy property achieved the best performance.

Table VII. *MAP* and *NDCG* Changing on Different Layer

	Layer-2	Layer-3	Layer-4	Layer-5	Multi-layer
MAP	0.607	0.713	0.829	0.878	0.92
NDCG@5	0.647	0.743	0.839	0.901	0.931
NDCG@10	0.675	0.771	0.847	0.92	0.923

Table VIII. The Results of the User Study on Location Recommendation

User Study	Recommender	Percentage of Ratings				Mean Score ($\bar{s}$)
		P(R1)	P(R2)	P(R3)	P(R4)	
The 1st run (2-week)	HGSM-Based CF + Content	0.253	0.438	0.222	0.087	**0.857**
	HGSM-Based CF method	0.231	0.420	0.256	0.093	0.789
	Item-Based CF method	0.225	0.434	0.267	0.074	0.810
	Random Recommendation	0.167	0.324	0.328	0.181	0.477
The 2st run (1-month)	HGSM-Based CF + Content	0.354	0.418	0.183	0.045	**1.081**
	HGSM-Based CF method	0.299	0.404	0.239	0.058	0.944
	Item-Based CF method	0.285	0.424	0.227	0.064	0.930
	Random Recommendation	0.171	0.335	0.319	0.175	0.502
The 3rd run (2-month)	HGSM-Based CF + Content	0.365	0.422	0.172	0.041	**1.110**
	HGSM-Based CF method	0.321	0.411	0.203	0.065	0.988
	Item-Based CF method	0.315	0.406	0.219	0.06	0.976
	Random Recommendation	0.206	0.292	0.295	0.207	0.497

5.3.2. Results of Location Recommendation. Table VIII presents the results of the user study we performed to evaluate the location recommendation. Using the GPS trajectories collected by 30 subjects over different periods (2 weeks, one month, and two months), we compare our system with two baseline methods (item-based collaborative filtering (CF) and random recommendation). Meanwhile, the contributions brought by the content-based method have been studied. In addition, the effectiveness of our location recommendation affected by a user's data scale can also be revealed through the study.

First, using the two-week dataset of these users, the preliminary HGSM-based CF approach showed clear advantages over the random recommendation; however, this approach does not outperform the item-based baseline method. After we incorporate the content-based method into the HGSM-based CF, the performance of our method has been significantly improved. Using the ANOVA test, we can view the statistical results on the two methods with and without integrating the content-based method.

$$P(R1) : F(1, 30) = 9.73,\ p < 0.01; P(R2) : F(1, 30) = 7.86,$$
$$p < 0.01;\ \bar{s} : F(1, 30) = 10.61,\ p < 0.01.$$

In short, combined with the content-based method, the percentage of R1 and that of R2 (refer to Table VI) of our method have been significantly enhanced in the evaluation results, i.e., more places matching users' tastes have been recommended. Meanwhile, the mean score of the evaluations offered by users has been clearly improved.

In contrast to item-based CF method, however, our approach, incorporating content-based method, does not show significant advantages.

$$P(R1) : F(1, 30) = 6.32,\ p < 0.05; P(R2) : F(1, 30) = 3.57,$$
$$p < 0.08;\ \bar{s} : F(1, 30) = 9.61,\ p < 0.01,$$

On one hand, according to the mean score $\bar{s}$, our method is significantly better than that of item-based CF method as $p < 0.01$. On the other hand, although the average value of P(R1) and P(R2) of our method are slightly bigger than that of item-based CF method, the difference is not very significant (as both $p > 0.04$).

With a small scale of GPS trajectories, our method becomes less capable of measuring the similarity between users. Therefore, the inferred ratings of a location might not that accurate.

Second, using the 1-month dataset, the HGSM-based method starts to present its advantages beyond the baseline methods. When combined with the content-based method, the HGSM-based approach achieved a significant improvement and clearly outperformed other methods.

In contrast to item-based CF method:

$$P(R1) : F(1, 30) = 15.25, p < 0.01;\ \bar{s} : F(1, 30) = 11.61, p < 0.01,$$

that is, the percentage of R1 and the mean score of our approach (incorporating the content-based method) is significantly higher beyond the item-based CF method. This means that more geospatial regions which strongly interest an individual have been recommended to the individual using our method.

In addition, we observe a clear improvement on the performance of our method over that shown in the first run ($P(R1) : F(1, 30) = 23.65, p \ll 0.01; \bar{s} : F(1, 30) = 18.35, p \ll 0.01$).

Third, with the two-month dataset, our method continues enhancing its performance and shows its advantages over the baseline.

In contrast to the item-based CF method:

$$P(R1) : F(1, 30) = 14.53, p < 0.01; \bar{s} : F(1, 30) = 13.26, p < 0.01.$$

In short, our method significantly outperforms the item-based CF method.

Although the performance of our method using the 2-month dataset is beyond that using the 1-month dataset, the improvement is not as much as expected.

$$P(R1) : F(1, 30) = 4.73, p < 0.05;\ \bar{s} : F(1, 30) = 6.13, p < 0.08.$$

The reason behind this phenomenon lies in the recommendation policy; we do not recommend places a user visited previously. On one hand, with a relatively large dataset, we are more likely to accurately estimate the similarity between users and properly infer a particular user's interests on an unvisited region. Thus, more places matching the user's tastes might be recommended. On the other hand, fewer regions remain for recommendation since the individual's location history might have covered many places.

5.3.3. User Feedback. When performing the user studies, we recorded some users' feedback and comments on our system. Typically, users' will be faced with four situations when giving feedback based upon a recommended location.

(1) The recommended locations have been visited by a user although the user's GPS trajectories have not covered these locations (perhaps they had no GPS recording device during previous visits). Therefore, such users are very confident in giving an evaluation on the recommended places shown on Web maps. For example, Yi Du said, "This is Houhai. I have been to this place twice, since the bars located in this place are very nice. From my perspective, it is better than Sanlitun village."
(2) A user has not visited a recommended location, however, this location contains some POIs, which are branches of the businesses (e.g., a different branch of Starbucks) that a user previously accessed and is interested in. In these cases, the users can also show their confidence when giving an evaluation. For instance, Yechen Hao said, "There is a branch of JiaoYe (a famous Thai-food restaurant) in this region. I tried another branch restaurant of JaioYe in Zhongguanchun when celebrating the birthday of my friend last year. It is very impressive, as you can watch a

Thai-culture show when enjoying pretty Thai-food. So, I believe this branch of Jiaoye deserves to be tried again."

(3) A user has not been to the recommended location while the location contains some POIs whose categories interest a user. For example, Quannan Li said, "There is a lake located in this region. I would like to travel to this place as I like lake and fresh air." Another example is, "There is a fantastic mall where I can go shopping, watch a movie, and enjoy Taiwanese food. It looks very similar to ShuangAn shopping mall," said Tingting. Although the subject's confidence is not as strong as if his/her faced with the previous two situations, he/she still can offer his/her evaluations on the recommended location. The more categories that interest the user in the recommended region, the better the evaluation the user could give to the region.

(4) A user has not visited the recommended places which do not contain any POIs or categories the subject is familiar with. In this case, a subject typically spends a relatively long time generating his/her evaluation on the places, and his/her confidence in this evaluation drops below the previous three situations. For instance, Yukun said, "it is difficult to identify what would be interesting within this region from the maps. Maybe they might be nice, but I do not know how nice it would be to me."

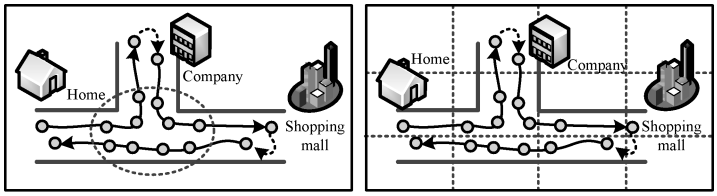
Because the users we selected to participate in the user study have been in Beijing for more than 6 years, they have rich knowledge and travel experiences about Beijing. In short, most evaluations offered on the recommended locations can be covered by the first three situations. Hence, the evaluation would be as accurate as if they really travel to the recommended locations.

5.4. Discussion

5.4.1. Location History Modeling

Stay Point Detection. The reasons we detect stay points using the algorithm shown in Figure 7, rather than directly clustering raw GPS points, lie in two aspects. (1) First, as depicted in Figure 30(a), most significant places, like shopping malls and restaurants, cannot be detected if we directly cluster raw GPS points. As GPS devices lose satellite signal indoors, few GPS points will be generated on such places. Thus, the density of points recorded there cannot satisfy the condition to formulate a cluster. On the contrary, some regions, like road crossings, that a user iteratively passes but does not carry semantic meanings, will be extracted. (2) To address this problem, an interpolation operation should be conducted for each user's GPS trajectories. For example, some GPS points should be interpolated between the last point recorded before a user entered a shopping mall and the first point recorded after the user came out from the mall. Then, the clustering algorithm could make sense. Now, we have GPS trajectories collected by 75 users over a period of 1 year. So, after interpolation we would have 75 (users) $\times$ 365 (day) $\times$ 24 (hours) $\times$ 3600 (Seconds) = 2,365,200,000 (as different GPS devices have different configurations, only 1 second/point can make the interpolated data consistent). The computation of clustering such a big dataset will be extremely heavy and will become a terrible disaster with increasing users. What are we supposed to do if there are thousands of users in the near future? Actually, it is very easy to reach this scale with thousands of users even if we do not perform the interpolation.

As demonstrated in Figure 30(b), the boundary problem of the grid-based partition method might also miss significant places. In other words, the geospatial region of a shopping mall might be split into several grids, in each of which the density of GPS points would not reach the condition to formulate a cluster.



(a) Clustering-based detection (b) Grid-based detection

Fig. 30. Other possible stay-point-detection algorithms.

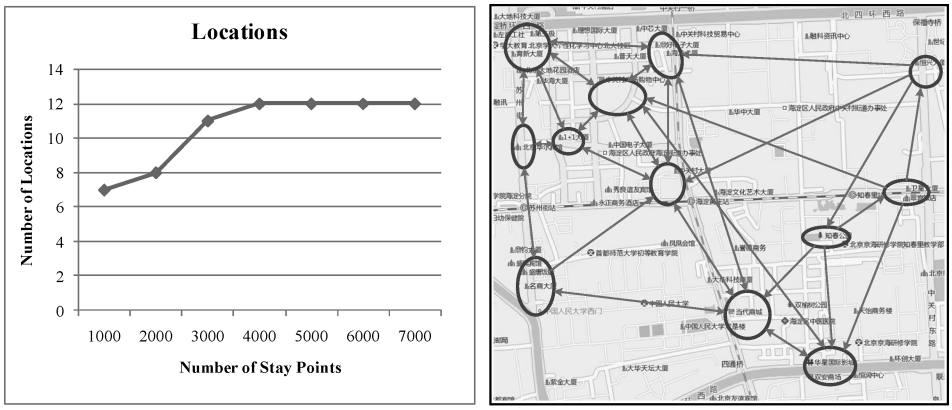
Clustering. Human trajectories show a high degree of temporal and spatial regularity [Gonzalez et al. 2008]. Each individual is characterized by a time-independent characteristic travel distance and a significant probability to return to a few highly frequented locations. Therefore, as compared to other methods using predefined grids or administrative regions to build hierarchy, clustering user-generated stay points is a data-driven approach which is more likely to feature the distribution of users' spatiotemporal data. Also, this method can discover the regions with semantic meanings and irregular structures, such as shopping streets and pretty beaches. Meanwhile, in contrast to clustering raw GPS points, grouping users' stay points can generate a more precise presentation of users' stays and save lots of computation.

5.4.2. User Similarity Measure.

Sequence Property. We are not surprised at the advantage of sequence property (over a set of separated locations) as shown in the experimental results, due to the following reasons. A sequence of geographic regions (1) can better model users' movements, and (2) reveals the correlation between a user's individual behaviors at different places. Therefore, beyond individual locations, the sequence property can present a more comprehensive view of a user's preferences and intention. The following cases can justify our claim.

One example can be described using two well-known types of people traveling to Xidan (a shopping street) and Houhai (a bar street along a pretty lake) in Beijing. One type of person typically travels to Xidan before visiting Houhai. He/she likes drinking in the bars of Houhai while these bars would not open until the evening. So, he/she can go shopping first in Xidan, and then travel to Houhai. On the contrary, the other type of person, who does not like the noisy bars, prefers to travel to Houhai first and then go to Xidan for shopping. In the daytime, Houhai is quiet and pretty; therefore, people can enjoy the natural scene of the lake nearby Houhai. Although these two types of people have visited Xidan and Houhai, their intention and preferences are quite different. One prefers the bars (entertainment); the other likes nature scenes (travel). Without the sequence property, we cannot differentiate these two types of people from their individual behaviors.

Another example can be introduced using the interns of Microsoft Research Asia (MSRA). Some students from Tsinghua University perform an internship in MSRA. Typically, they would generate a sequence of "Tsinghua → MSRA." Occasionally, some famous professors from the USA would visit MSRA and Tsinghua University at different times but they would not access these two places in a sequence. Maybe, they would travel to these two places in different trips (e.g., visit MSRA in January while traveling to Tsinghua in June) or start from the hotel they are staying at (e.g., MSRA → Hotel → Tsinghua). Obviously, the interns and professors are different types of people since they have different intentions and preferences when visiting these two places. However, the professors could share the same location history with the interns of MSRA



(a) The number of the locations changing over the stay points (b) Distribution and relation of the locations

Fig. 31. The discovered locations within a given geospatial range.

(Tsinghua and MSRA). In short, we cannot differentiate them if we do not take into account the sequence property.

In the cases mentioned, the measure of similarity-by-count, the cosine similarity and the Pearson similarity cannot successfully differentiate between these people as they do not take into account the sequence property.

Hierarchy Property. In general, Figure 26 and Figure 27, respectively, present the contribution of the hierarchy property of HGSM over the single-layer method. Further, Table VI illustrates how this contribution is generated by investigating the performance of each layer of the hierarchy. The layer with a finer granularity is more capable of differentiating similar users from each other beyond the layer with a coarse granularity. Imagine that a cluster might cover a whole city on a higher layer of the hierarchy. At this moment, users living in this city are indistinctive if we only explore user similarity on that layer. On the contrary, if we only consider users' location histories on the layer with a fine granularity, users' high-level movements would be neglected. Thus, some similar users would be missed. For instance, two individuals travel from Beijing to Seattle frequently while they share little location history within Beijing. In this case, the similarity between the two users cannot be well recognized if we only investigate their movements on the bottom layer. Overall, the layer with a relatively fine granularity improves HGSM's capability of precisely differing similar users, while the layer with a relatively coarse granularity enhances HGSM's capability of recalling similar users.

5.4.3. New Users Problem

The new user problem will affect two aspects of our system, scalability and the cold start.

Scalability. In the experiments, we observed that the number of locations discovered in a fixed geospatial range does not continue to increase when the number of users joining in our system exceeds a certain value. As depicted in Figure 31(a), we randomly add the stay points detected from 75 users' location histories step by step into the dataset, which will be hierarchically clustered into several regions. As a result, in a geospatial range of 3 kilometers by 3 kilometers, the number of discovered locations does not increase any more when the number of stay points exceeds 4000. Figure 31(b) illustrates the distribution of these locations within the given geospatial range. Also, the cooccurrence of two locations in users' GPS trajectories is demonstrated with some

directed lines. Here, an edge between two locations means that at least 5 users have traveled to these two locations in a sequence; the direction of which is specified by an arrow.

Motivated by the above observation, rather than rebuilding the framework with the arrival of new users, we can insert the stay point of a new user into the existing framework, F, and update the framework in a relatively low frequency, e.g., 1 update per week. Meanwhile, it is not necessary to recalculate the similarity between users once someone uploads some new GPS trajectories. Adding a few days GPS trajectories to an individual's dataset would not cause many changes in the user similarity matrix.

Cold Start. Typically, a new user could have little data when they join in this Web community. Thus, we are not able to infer their interests in locations no matter what kinds of recommendation techniques we use. In our system, we recommend the top_N locations with the most visited popularity to a user having little GPS trajectories. Later, when the time span of the user's dataset reaches a certain value, for example, two weeks, we use HGSM to measure the similarity between him/her and others.

5.4.4. New Locations Problem. When a new location appears in the system, it represents that at least a few people have visited this place. Otherwise, it is impossible to formulate a cluster on this region when we perform density-based clustering. In other words, we can obtain a few ratings on this region from several people once the region is discovered by the system. Hence, the cold start problem of our system is not as serious as other recommenders. However, a location with a few ratings might also face the challenges of being ignored during the process of the recommendation. Therefore, we combine the content-based method with collaborative filtering to offer a better recommendation result. By exploring the categories of POIs within geospatial regions, we are able to find some regions similar to a new location based on their profiles. Later, other users' ratings on these regions can be used as estimated ratings of these users on this new location.

6. RELATED WORK

6.1. Mining Location History

6.1.1. Mining Personal Location History. Motivated by the convenience of data collection, some research has been performed based on individual GPS data during the past years. These works include detecting significant locations of a user [Ashbrook et al. 2003; Hariharan and Toyama 2004], predicting the user's movement among these locations [Krumm and Horvitz 2007 and Liao et al. 2005], and recognizing user-specific activities at each location [Liao et al. 2004 and Patterson et al. 2003]. As opposed to these works, we aim to mine knowledge from multiple users' location histories rather than recognize user-customized activity. Hariharan and Toyama [2004] proposed the concepts of stay points and destinations which can be used to model a particular users' location history based on GPS trajectories. Although the stay-point-detection method is similar to our approach, in this article, we aim to not only model an individual's location history but also make multiple users' location histories comparable and understandable.

6.1.2. Mining Multiple Users' Location Histories. Gonotti et al. [2007] developed an extension of the sequential pattern mining paradigm that analyzes the trajectories of moving objects. The trajectory pattern they called represents a set of individual's trajectories that share the property of visiting the same sequence of places with similar travel times. MSMLS [Krumm et al. 2006] uses a history of a driver's destination along with data about driving behavior extracted from multiple users' GPS trajectories to predict where a driver may be going as a trip progresses. Zheng et al. [2008a, 2008b, 2010b] aim to infer users' transportation modes, such as walking and driving, based on the

GPS trajectories of 60 individuals. Meanwhile, respectively using location-acquisition techniques of 802.11 [Krumm and Horvitz 2004] and GSM networks [Timothy et al. 2006], some projects attempt to recognize user mobility, such as stationary and walking, from the location histories of a group of people. In contrast to the techniques mentioned previously, we extend the paradigm of mining multiple users' location histories from recognizing user behaviors to understanding the correlation between user behaviors.

6.2. Common Recommendation Systems

Typically, a recommender system compares a user's profile to some reference characteristics and seeks to predict the rating that the user would give to an item they had not yet considered. These characteristics may be from the information of items (the content-based approach) or the user's social environment (the collaborative filtering approach). Recommender systems are usually classified into the following categories based on how recommendations are made [Balabanovic and Shoham 1997].

(1) Content-based recommendations. The user will be recommended items similar to the ones the user preferred in the past.
(2) Collaborative recommendations. The user will be recommended items that people with similar tastes and preferences liked in the past.
(3) Hybrid approaches. These methods combine collaborative and content-based methods.

6.2.1. Collaborative Filtering. The general idea behind collaborative filtering [Goldberg et al. 1992 and Nakamura and Abe 1998] is that similar users vote in a similar manner on similar items. Thus, if similarity is determined between users and items, a potential prediction can be made for the vote of a user for some items. According to Breese et al. [1998], algorithms for collaborative recommendations can be grouped into two general classes, memory-based (or heuristic-based) and model-based.

Memory-Based. Memory-based algorithms essentially are heuristics that make rating predictions based on the entire collection of previously rated items by the users [Adomavicius and Tuzhhilin 2006]. That is, the value of the unknown rating for a user and an item is usually computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item. There are at least two classes of memory-based collaborative filtering; user-based [Shardanand and Mayes 1995 and Resnick et al. 1994] and item-based techniques [Lemire and Maclachlan 2005 and Badrul et al. 2001].

(1) User-based techniques are derived from similarity measures between users. The similarity between two users, (A and B), is essentially a distance measure and is used as a weight. In other words, when predicting the rating of user A on an item, the more similar the user A and B are, the more weight the rating of user B will carry on the item. Various approaches have been used to compute the similarity between users in collaborative recommender systems. In most of these approaches, the similarity between two users is based on their ratings of items that both users have rated. The two most popular approaches are correlation and cosine-based. In a social network, a particular user's neighborhood with similar tastes or interests can be found by calculating the Pearson correlation [Sarwar et al. 2000]. Further, by collecting the preference data of the top-N nearest neighbors of the particular user, the user's preferences can be predicted by calculating the data using certain techniques. Spertus et al. [2005] present an extensive empirical comparison of six distinct measures of similarity for recommending online communities to members of the Orkut social network. As a result, they found that the cosine similarity

measure showed the best empirical results, beyond that of other measures, such as log odds and point-wise mutual information.
(2) Item-based techniques predict the ratings on one item based on the ratings on another item. Examples of binary item-based collaborative filtering include Amazon's item-to-item patented algorithm [Linden et al. 2003] which computes the cosine similarity between binary vectors representing the purchases in a user-item matrix. Slope one [Lemire and Maclachlan 2005] is the simplest form of nontrivial item-based collaborative filtering based on ratings. Their simplicity makes it especially easy to implement them efficiently, while their accuracy is often on a par with more complicated and computationally expensive algorithms.

Model-Based. In contrast to memory-based methods, model-based algorithms [Getoor and Sahami 1999 and Hofmann 2003] use the collection of ratings to learn a model which is then used to make rating predictions. For example, Breese et al. [1998] proposed a probabilistic approach to collaborative filtering. It is assumed that rating values are integers between 0 and n, and the probability expression is the probability that a user will give a particular rating to an item given that user's ratings of the previously rated items. Hofmann [2003] proposed a collaborative filtering method in a machine learning framework where various machine learning techniques (such as artificial neural networks) coupled with feature extraction techniques can be used.

6.2.2. Hybrid Approaches. Several recommendation systems use a hybrid approach by combining collaborative and content-based methods which helps to avoid certain limitations of content-based and collaborative systems. Different ways to combine collaborative and content-based methods into a hybrid recommender system can be classified as follows [Adomavicius and Tuzhhilin 2006].

(1) Implementing collaborative and content-based methods separately and combining their predictions [Pazzani 1999 and Claypool et al. 1999];
(2) Incorporating some content-based characteristics into a collaborative approach [Melville et al. 2002 and Good et al. 1999];
(3) Incorporating some collaborative characteristics into a content-based approach [Soboroff and Nicholas 1999];
(4) Constructing a general unifying model that incorporates both content-based and collaborative characteristics [Basu et al. 2001].

In our work, we incorporated a content-based method into a user-based collaborative filtering algorithm to estimate the rating of a user on an item. The major difference between our work and the techniques mentioned previously lies in two aspects. One is that we extend the direction of user similarity exploration from people's online behaviors to the real-world location histories. The other is the novel measure, HGSM, we designed to estimate the similarity between users.

6.3. Location-Based Recommender System

6.3.1. Systems Based on Real-Time Location. Quite a few recommender systems take into account a particular user's current geographic location when recommending content to the user. Yang et al. [2008] proposed a location-aware recommender system that accommodates a customer's shopping needs with location-dependent vendor offers and promotions. Brunato et al. [2002] attempt to recommend Web sites to individuals depending on the locations where they access the Web. As compared to our recommender, these systems focus on employing a customer's real-time location as a constraint when rendering other information, like Web sites, to the customer. However, we mine

multiple users' location histories and explore the correlations between individuals and locations.

6.3.2. Systems Based on Location History. Zheng et al. [2009b] conducted a generic travel recommender that provides a user with the top interesting locations and travel sequences (in a geographic region) mined from a large number of users' GPS trajectories. Further, Zheng et al. [2010a and 2010e] perform a location-activity collaborative recommendation which (1) recommends to a user some proper activities that could be proper to perform in a given location or (2) offers a set of candidate locations where a specific activity, like shopping, can be conducted. In contrast to these technologies, we implement a personalized location recommendation in this work, instead of a generic one. Horozov et al. [2006] proposed an enhanced collaborative filtering solution that uses location as a key criterion to generate the recommendation of a restaurant. Takeuchi and Suginoto [2006] attempt to recommend shops to users based on their individual preferences and needs, estimated by analyzing their past location histories. Zheng et al. [2010c] first learn the correlation between locations and then use the correlation to enable a personalized location recommender. Although exploring the correlation among geographic locations, these systems still directly employ the technologies used in traditional recommender systems without considering the sequence property of users' movement and the hierarchy property of geographic spaces. Justified by the experimental results, such properties are vital to differentiate geographic information systems from other online communities, like Amazon, when measuring similarity between users.

7. CONCLUSION

In this article, we reported on a location-history-based recommender system which (1) uses a particular individual's visits on a geospatial location as their implicit ratings on the location and (2) tries to predict a particular user's interest in an unvisited location in terms of their location history and those of other users. In this system, each user will be recommended a group of potential friends who might share similar tastes of travel, sports, or entertainment, and a list of geospatial locations which might match the user's interests. Therefore, a user can organize some social activities in a community and expand their geographical knowledge with minimal effort. This is a step towards understanding the correlations between users and locations using user-generated geospatial data. Also, this is a step towards integrating recommender systems into GIS communities on the Web.

A similarity measure, HGSM, is proposed for this recommender to uniformly model various users' location histories and infer the similarity among users. Three features, the sequence property of user movement, hierarchy property of geographical spaces and visited popularity of a location, have been considered in this similarity measure. Then, we incorporated a content-based method into a user-based collaborative filtering algorithm which uses HGSM as the user similarity measure to estimate the rating of a user on an item. Hence, we are able to reduce, to some extent, the cold start problem of recommender systems and offer users a better location recommendation.

Using the GPS trajectories collected by 75 subjects in the past year, we evaluated our recommender system. Regarding the friend recommendation, HGSM outperformed the baseline methods of similarity-by-count, the cosine similarity and the Pearson similarity. Moreover, the three proposed features showed their advantages in measuring the similarity between users. In terms of *NDCG* and *MAP*, the performance of our approach has been improved step by step when those proposed features were taken into account one by one. Regarding the location recommendation, we performed a three-run user study with 30 users selected from the 75 subjects. According to these users' feedback, our system outperformed item-based collaborative filtering and random

recommendation by providing them with more attractive places and more personalized user experiences. When combined with the content-based method, the HGSM-based CF approach has achieved a significant improvement. In addition, this study revealed the effectiveness of our system depending on the scale of a user's dataset. With more GPS trajectories from an individual, our system is more likely to infer a user's interests accurately and, hence, provide him/her with a more personalized recommendation.

8. FUTURE WORK

In the future, we would like to extend our work in the following three areas.

Regarding similarity measure. We intend to take into account more features of users' movements, such as the distance of a similar sequence and the time a user stayed in a geospatial region. Meanwhile, improving the efficiency of measuring user similarity is also a potential work.

Regarding the evaluation of friend recommendation. We plan to conceive a new approach evaluating our similarity measure, study user behaviors after they are recommended with potential friends, check whether they become real friends and conduct this evaluation with more users from a variety of cities.

Regarding location recommendation. First, we aim to further understand the similarity between locations and propose a more sophisticated content-based method. Second, we try to incorporate the content-based method into the collaborative filtering method in a more sophisticated manner. Third, we would like to study some item-based CF methods to estimate a user's interest levels in unvisited places.

REFERENCES

ADOMAVICIUS, G. AND TUZHHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Engin. 17*, 6, 734–749.

ANKERST, M., BREUNIG, M. M., KRIEGEL, H., AND SANDER, J. 1999. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM Press, 49–60.

ASHBROOK, D. AND STARNER, T. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiq. Comput. 7*, 5, 275–286.

BADRUL M., SARWAR, G. K., JOSEPH, A., AND KONSTAN, J. 2001. Riedl: Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 3rd International Conference on World Wide Web*. 285–295.

BALABANOVIC, M. AND SHOHAM, Y. 1997, Fab: Content-based, collaborative recommendation. *Comm. ACM 40*, 3, 66–72.

BASU, C., HIRSH, H., AND COHEN W. 2001. Recommendation as classification: Using social and content-based information in recommendation. Recommender systems papers from 1998 workshop, Tech. rep. WS-98-08, AAAI Press.

BIKELY. http://www.bikely.com!

BOGERS, T. AND BOSCH, A. 2007. Comparing and evaluating information retrieval algorithms for news recommendation. In *Proceedings of the ACM Conference on Recommender Systems*. 141–144.

BREESE, J., S., HECKERMAN, D., AND KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the International 14th Conference on Uncertainty in Artificial Intelligence*.

BRUNATO, M., BATTITI, R., VILLANI, A., AND DELAL, A. 2002. A location-dependent recommender system for the Web. Tech. rep. DIT-02-093, University of Trento.

CHEN, Z., SHEN, H. T., ZHOU, X., ZHENG, Y., AND XIE, X 2010. Searching trajectories by locations: An efficiency study. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM Press.

CLAYPOOL, M., GOKHALE, A., MIRANDA, T., MURNIKOV, P., NETES, D., AND SARTIN, M. 1999. Combining content-based and collaborative fitters in an online newspaper. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*.

COUNTS, S. AND SMITH, M. 2007. Where were we: Communities for sharing space-time trails. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*.

DAS, A. S., DATAR, M., GARG A., AND RAJARAM, S. 2007. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*. 271–280.

GETOOR, L. AND SAHAMI, M. 1999. Using probabilistic relational models for collaborative filtering. In *Proceedings of the Workshop on Web Usage Analysis and User Profiling*.

GOLDBERG, D., DAVID, N., BRAIN, M. O., AND DOUGLAS, T. 1992. Using collaborative filtering to weave an information tapestry. *Comm. ACM 35*, 12, 61–70.

GONOTTI, F., NANNI, M., PEDRESCHI, D., AND PINELLI, F. 2007. Trajectory pattern mining. In *Proceedings of the 13rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM Press, 330–339.

GONZALEZ, M. C., HIDALGO, C., A., AND BARABASI, A.-L. 2008. Understanding individual human mobility patterns. *Nature 453*, 6, 779–780.

GOOD, N., SCHAFER, J. B., KONSTAN, J. A., BORCHERS, A, SARWAR, A. B., HERLOCKER, J. L., AND RIEDL, J. 1999. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the Conference on Artificial Intelligence*. AAAI Press, 439–446.

GPS SHARING. http://gpssharing.com.

GPS TRACK ROUTE EXCHANGE FORUM. http://www.gpsxchange.com.

HARIHARN, R. AND TOYAMA, K. 2004. Project Lachesis: Parsing and modeling location histories. In *Proceedings of the 3rd International Conference on Geographic Information Science.* 106–124.

HOFMANN, T. 2003. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press. 259–266.

HOROZOV, T., NARASIMHAN, N., AND VASUDEVAN, V. 2006. Using location for personalized POI recommendations in mobile environments. In *Proceedings of the International Symposium on Applicaiions on Internet*. SAINT Press, 124–129.

JARVELIN, K. AND KEKALAINEN, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inform. Syst. 22*, 1,422–446.

KRUMM, J. AND HORVITZ, E. 2004. LOCADIO: Inferring motion and location from wi-fi signal strengths. In *Proceedings of the 1st International Conference on Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*. IEEE Press, 4–13.

KRUMM, J. AND HORVITZ, E. 2006. Predestination: Inferring destinations from partial trajectories. In *Proceedings of the 8th International Conference on Ubiquitous Computing*. Springer-Verlag, 243–260.

KRUMM, J. AND HORVITZ, E. 2007. Predestination: Where do you want to go today? *IEEE Comput. Mag. 40*, 4, 105–107.

LEMIRE, D. AND MACLACHLAN, A. 2005. Slope One: Predictors for online rating-based collaborative filtering. In *Proceedings of the SIAM Data Mining Conference.* SIAM Press.

LI, Q., MYAENG, S., H. AND KIM, B. M. 2007. A probabilistic music recommender considering user opinions and audio features. *Int. J. Inform. Process. Manage. 43*, 2, 473–487.

LI, Q., ZHENG, Y., CHEN, Y., LIU, W., AND MA, W. 2008. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* ACM, 1–10.

LIAO, L., FOX, D., AND KAUTZ, H. 2004. Learning and inferring transportation routines. In *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press, 348–353.

LIAO, L., PATTERSON, D. J., FOX, D., AND KAUTZ, H. 2005. Building personal maps from GPS data. *Ann. N.Y. Acad. Sci. 1093*, 249–265.

LINDEN, G., SMITH, B., AND YORK, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet. Comput. 7*, 1, 76–80.

LR, Q., ZHENG Y., XIE, X., CHEN, Y., LIU, W., AND MA, W. 2008. Mining user similarity based on location history. In *Proceedings of the 16th International Conference on Advances in Geographic Information Systems*, ACM Press: 1–10.

MELVILLE, P., MOONEY, R. J., AND NAGARAJAN, R. 2002. Content-boosted collaborative filtering for hnproved reconnnendations. In *Proceedings of the 18th National Conference on Artificial Intelligence*. AAAI Press, 187–192.

NAKAMURA, A. AND ABE, N. 1998. Collaborative filtering using weighted majority prediction algorithms, In *Proceedings of the 15th International Conference on Machine Learning.* ACM Press, 395–403.

PATTERSON, D. J., LIAO, L., FOX, D., AND KAUTZ, H. 2003. Inferring high-level behavior from low-level sensors. In *Proceedings of the 8th International Conference on Ubiquitous Computing.* Springer, 73–89.

PAZZANI, M. 1999. A framework for collaborative, content-based, and demographic filtering. *Artif. Intel. Rev. 13*, 5, 393–408.

RESNICK, P., IAKOVOU, N., SUSHAK, M., BERGSTROM, P., AND RIEDL, J. 1994. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the International Conference on Computer Supported Cooperative Work*. ACM Press, 175–186.

SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Application of dimensionality reduction recommender system: A case study, In *Proceedings of the ACM WebKDD Workshop*.

SHARDANAND, U. AND MAES, P. 1995, Social information filtering: Algorithms for automating "word of mouth." In *Proceedings of the International Conference on Human Factors in Computing Systems*. ACM Press, 210–217.

SOBOROFF, I. AND NICHOLAS, C. 1999. Combining content and collaboration in text filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence Workshop: Machine Learning for Information Filtering*. ACM Press, 86–91.

SPERTUS, E., SAHAMI, M., AND BUYUKKOKTEN, O. 2005. Evaluating similarity measures: A large-scale study in the Orkut social network. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 678–684.

SPORTSDO. 2007. http://sportsdo.net/Activity,IActivityBlog.aspx.

TAKEUCHI, Y. AND SUGIMOTO, M. 2006. CityVoyager: An outdoor recommendation system based on user location history. In *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing*. Springer, 625–636.

TIEMANN, M. AND PAUWS, S. 2007. Towards ensemble learning for hybrid music recommendation. In *Proceedings of the ACM Conference on Recommender Systems*. ACM Press, 177–178.

TIMOTHY, S., VARSHAVSKY, A., LAMARCA, A., CHEN, M. Y., AND CHOUDHURY, T. 2006. Mobility detection using everyday GSM traces. In *Proceedings of the 11th International Conference on Ubiquitous Computing*. Springer, 212–224.

TOBLER, W. 1970. A computer movie simulating urban growth in the Detroit region. *Econ. Geog. 46*, 2, 234–240.

WANG, L., ZHENG, Y., XIE, X., AND MA, W. Y. 2008. A flexible spatio-temporal indexing scheme for large-scale GPS track retrieval, In *Proceedings of the 9th International Conference on Mobile Data Management*. IEEE Press, 1–8.

YANG, W. S., CHENG, H. C., AND DIA, J. B. 2008. A location-aware recommender system for mobile shopping environments. *Exp. Syst. Appl. Int. J.* 437–445.

ZHENG, W., CAO, B., ZHENG, Y., XIE, X., AND YANG, Q. 2010a. Collaborative filtering meets mobile. recommendation: A user-centered approach. In P*roceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAl Press.

ZHENG, W., ZHENG, Y., XIE, X., AND YANG, Q. 2010f. Collaborative location and activity recommendations with GPS history Data. In *Proceedings of the 19th International Coriference on World Wild Web*.

ZHENG, Y. AND XIE, X. 2010c. Learning Location Correlation from GPS trajectories. In *Proceedings of the International Conference on Mobile Data Management*. IEEE Press, 27–32.

ZHENG, Y. AND XIE, X. 2010e. Learning travel recommendations from user-generated GPS traces. *ACM Trans. Intel. Syst. Technol. 2*, 1.

ZHENG, Y., CHEN, Y., LI, Q., XIE, X., AND MA, W. Y. 2010b. Understanding transportation modes based on GPS data for Web applications. *ACM Trans. Web. 4*, 1, 1–36.

ZHENG, Y., CHEN, Y., XIE, X., AND MA, W. Y. 2009a. GeoLife2.0: A location-based social networking service. In *Proceedings of the International Conference on Mobile Data Management.* IEEE Press, 357–358.

ZHENG, Y., LI, Q., CHEN, Y., XIE, X., AND MA, W. Y. 2008a Understanding mobility based on GPS data, In *Proceedings of 10th International Conference on Ubiquitous Computing*. ACM Press, 312–321.

ZHENG, Y., LIU, L., WANG, L., AND XIE, X. 2008b. Learning transportation mode from raw GPS data for geographic applications on the Web. In *Proceedings of the 11th International Conference on World Wide Web*. ACM Press, 247–256.

ZHENG, Y., WANG, L., ZHANG, R., XIE, X., AND MA, W. Y. 2008c. GeoLife: Managing and understanding your past life over maps. In *Proceedings of the 9th International Conference on Mobile Data Management*. IEEE Press, 211–212.

ZHENG, Y., XIE, X., AND MA, W. Y. 2008d. Search your life over maps. In *Proceedings of the International Workshop on Mobile Information Retrieval*. ACM Press, 24–27.

ZHENG, Y., XIE, X., AND MA, W. Y. 2010d. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engin. Bull. 33*, 2, 32–40.

ZHENG, Y., ZHANG, L., XIE, X., AND MA, W. Y. 2009b. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wild Web.* ACM Press, 791–800.

Received September 2008; revised May 2009, November 2009; accepted April 2010