Mining User Similarity Based on Location History

Quannan Li^{1,2}, *Yu Zheng*^{2}, *Xing Xie*^{2},*Yukun Chen*^{2}, *Wenyu Liu*^{1}, *Wei-Ying Ma*^{2}

^{1}Dept. Electronics and Information Engineering, Huazhong University of Science and Technology,
Wuhan, 430074, P.R. China
^{2}Microsoft Research Asia
4F, Sigma Building, No.49 Zhichun Road, Haidian District, Beijing 100190, P. R. China
^{1}liuwy@mail.hust.edu.cn
^{2}{v-quali, yuzheng, xingx, v-yukche, wyma}@microsoft.com

ABSTRACT

The pervasiveness of location-acquisition technologies (GPS, GSM networks, etc.) enable people to conveniently log the location histories they visited with spatio-temporal data. The increasing availability of large amounts of spatio-temporal data pertaining to an individual's trajectories has given rise to a variety of geographic information systems, and also brings us opportunities and challenges to automatically discover valuable knowledge from these trajectories. In this paper, we move towards this direction and aim to geographically mine the similarity between users based on their location histories. Such user similarity is significant to individuals, communities and businesses by helping them effectively retrieve the information with high relevance. A framework, referred to as hierarchical-graph-based similarity measurement (HGSM), is proposed for geographic information systems to consistently model each individual's location history and effectively measure the similarity among users. In this framework, we take into account both the sequence property of people's movement behaviors and the hierarchy property of geographic spaces. We evaluate this framework using the GPS data collected by 65 volunteers over a period of 6 months in the real world. As a result, HGSM outperforms related similarity measures, such as the cosine similarity and Pearson similarity measures.

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*.

General Terms

Algorithms, Measurement, Experimentation.

Keywords

Spatio-temporal data mining, User similarity, GPS logs, Similar sequence Matching.

1. INTRODUCTION

The increasing pervasiveness of location-acquisition technologies such as GPS, GSM network, etc., is leading to the collection of large spatio-temporal datasets and the opportunity of discovering valuable knowledge about movement behavior. Recently, a branch of geographic applications based on user-generated GPS data has appeared on the Web and received considerable attention. In such applications [1][2][3], using GPS-enabled devices, people can record travel/sports experience with GPS logs, and then upload, visualize and browse their GPS data on a Web map. Further, users are enabled to exchange life experiences by sharing GPS logs in the Web community. GPS-log-sharing provides a more explicit and fancy approach than text to help people express their life experience. However, so far, these applications still use raw GPS data directly without much understanding; this is not optimal to the development of such applications. Actually, besides the GPS data itself, people intend to know more information about user intention and user interests behind the given data.

To address this issue, quite a few projects [9][12][13][15] aiming to understand user-specific activity from individual GPS data have emerged. However, in these projects more attention has been paid on detecting significant locations of a user, predicting the user's movement among these locations and recognizing individual activities on each location. Meanwhile, a few techniques [6][10][11][16][21] have been proposed to mine knowledge from multiple users' GPS logs. Unfortunately, so far, the correlation between users are not explored in these works even though such correlation has significance to both individuals and merchants in helping them effectively retrieve information with high relevance.

Being that it is important information to customers and commercial enterprises, user similarity has been used in many services and application systems over the past twenty years. Most of the research and products have been performed based on users' transaction records in supermarkets or bookstores and online user behavior in Web communities. On one hand, by employing user similarity, an individual would become capable of discovering potential friends who share similar interests in books, music and movies, etc., with him/her. Further, the person could leverage similar users' experiences to extend individual knowledge and retrieve the information matching his/her tastes with minimal efforts. On the other hand, based on user similarity, merchants would become more capable of improving their sales and marketing by reasonably recommending products to customers.

User similarity is also important to geographic information systems. It can not only explore the relationship between users but also reveal the correlation among geographic locations. For instance, people who like hiking might find some potential friends

ACM GIS '08, November 5-7, 2008. Irvine, CA, USA
(c) 2008 ACM ISBN 978-1-60558-323-5/08/11...$5.00"

in the community when attempting to sponsor a hiking activity. Moreover, from similar users' past experiences people are more likely to retrieve, with minimal efforts, some travel routes or places which might match their taste and preference.

The work reported in this paper is motivated by the importance of user similarity, increasing the availability of large amounts of data pertaining to individual trajectories and the first law of geography. According to the first law of geography, everything is related to everything else, but near things are more related than distant things. Given the close relationship between our daily lives and geographic location, user-generated GPS logs imply to some extent user behavior and user preference. Hence, we claim that people who have similar location histories would share similar interests and preferences. The more location histories they shared, the more related these two users would be.

In this paper, we aim to mine user similarity based on user-generated GPS trajectories in the real-world. It offers a novel approach to measure user similarity geographically. Meanwhile, it is a step towards mining knowledge from multiple users' spatio-temporal data. A framework, referred to as hierarchical-graph-based similarity measurement (HGSM), is proposed to model people's location histories and to explore the similarity between users in geographic spaces *sequentially* and *hierarchically*. The contributions of our work lie in following aspects.

- HGSM provides an approach to effectively and consistently model each individual's location history. We put all users' data together and hierarchically cluster it into geographic regions (clusters) in a divisive manner. Such hierarchical clusters offer a unified framework for each user to build an individual hierarchical-graph based on his/her own data. In such graphs, a node stands for a region a user accessed and an edge between two nodes denotes the order of the two regions being visited by the user.
- ***Sequentially***: When measuring the similarity between users, we take into account not only the geographic regions they accessed, but also the sequence of these regions being visited. The longer the sequence matched between two users' location histories, the more related these two users might be.
- ***Hierarchically***: We mine user similarity by exploring people's movement behavior on different scales of geographic spaces. From the top to the bottom of the proposed framework, the granularity of geographic regions increases from being coarse to being fine, while the geospatial scale of each graph node decreases from being large to being small. Consequently, users who share similar location history on a lower layer (with fine granularity) might be more similar than others who share location history on a higher layer (with coarse granularity).

The rest of the paper is organized as follows. In Section 2, we survey related work, and point out the difference between ours and others. In Section 3, after clarifying some definitions, we outline the architecture of HGSM, which includes three processes: location history extraction, user similarity exploration, and recommendation. Further, user similarity exploration is described in detail in Section 4. In Section 5, we evaluate the performance of HGSM using GPS data collected by 65 volunteers over a period of 6 months. Some major experimental results and related discussions are also provided in this section. Finally, we draw our conclusion and offer an outlook on the future work in Section 6.

2. RELATED WORK

2.1 Mining Location History

Mining personal location history: Motivated by the convenience of data collection, quite a few researches have been performed based on individual GPS data during the past years. The work [9][12][13][15] includes detecting significant locations of a user, predicting the user's movement among these locations and recognizing user-specific activities on each location. As opposed to these works, we aim to mine knowledge from multiple users' location histories rather than recognize user-customized activity.

Mining multiple users' location histories: Fosca et al. [6] 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 [10] 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. [21] aim to infer users' transportation mode, such as walking and driving, etc., based on the GPS trajectories of 60 individuals. Meanwhile, respectively using location-acquisition techniques of 802.11 [11] and GSM network [16], some projects attempt to recognize user mobility, such as stationary and walking, etc., from the location histories of a group of people. In contrast to the techniques mentioned above, we extend the paradigm of mining multiple users' location histories from recognizing user behavior to understanding the correlation between user behaviors.

2.2 Recommendation Systems

2.2.1 Common recommendation systems

Recommendation systems use the opinions of a community to help individuals in that community more effectively identify content of interest from a potentially overwhelming set of choices [4]. Companies like Amazon [14] have shown that a retail experience can be substantially enhanced by statistically correlating macro patterns in buying and browsing behavior. A well-known technique used in such systems is called collaborative filtering when trying to predict the rating of a product to a particular user. The general idea behind collaborative filtering is that similar users vote similarly 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.

User similarity has also been explored in social networks to facilitate people to identify potential friends and content of interest on the Web. One of the most commonly used algorithms is Nearest Neighborhood approach [18]. In a social network, a particular user's neighborhood with similar taste or interest can be found by calculating the Pearson Correlation. Further, by collecting the preference data of top-N nearest neighbors of the particular user, the user's preference can be predicted by calculating the data using certain techniques. Spertus et al. [17] 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 despite other measures, such as point-wise mutual information.

The major difference between our work and the techniques mentioned above lies in two aspects. One is we extend the direction of user similarity exploration from people's online behavior to the real-world location histories. The other is the novel measure of similarity, HGSM, we designed for geographic information systems.

2.2.2 Location-based recommender system

***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. [20] proposed a location-aware recommender system that accommodates a customer's shopping needs with location-dependent vendor offers and promotions. Brunato et al. [5] attempt to recommend websites to individuals depending on the locations where they access the Web. As compared to our HGSM, these systems focus on employing a customer's real-time location as a constraint when rendering information to the customer, while we aim to mine multiple users' location histories and explore the similarity between individuals and locations.

***Systems based on location history*:** Recently, using people's real-world location history, some recommender systems such as *Geowhiz* [7] and *CityVoyager* [19] have been designed to recommend geographic locations like shops or restaurants to users. Horozov et al. [7] proposed an enhanced collaborative filtering solution that uses location as a key criterion to generate a recommendation of a restaurant. In paper [19], the authors attempt to recommend shops to users based on their individual preferences and needs estimated by analyzing their past location histories. Although aiming to explore 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 behavior 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 in measuring similarity between users and locations.

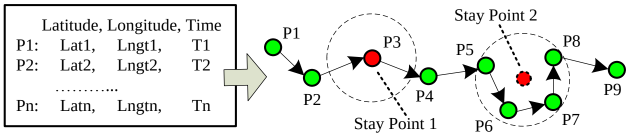
3. ARCHITECTURE

In this section, we first clarify some terms used in this paper and briefly describe the architecture of our work.

3.1 Preliminary

In this subsection, we will clarify some terms, including GPS logs, GPS trajectory, stay point, location history and hierarchical graph.

GPS log and GPS trajectory: Basically, as depicted in the left part of Figure 1, a GPS log is a sequence of GPS points $P=\{p_1, p_2, \dots, p_n\}$. Each GPS point $p_i \in P$ contains latitude ($p_i.Lat$), longitude ($p_i.Lngt$) and timestamp ($p_i.T$). As depicted on the right part of Figure 1, on a two dimensional plane, we can connect these GPS points into a GPS trajectory (*Traj*) according to their time serials.



	Latitude,	Longitude,	Time
P1:	Lat1,	Lngt1,	T1
P2:	Lat2,	Lngt2,	T2
	………..		
Pn:	Latn,	Lngtn,	Tn

Figure 1. GPS logs and stay points

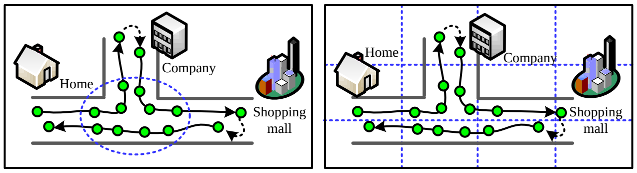
Stay point: A stay point *S* stands for a geographic region where the user stayed for a while. As compared to a raw GPS point, each stay point carries a particular semantic meaning, such as the place we work/live, restaurant we visit and the spots we travel to, etc. Figure 1 demonstrates two categories of stay points. In one situation, like *stay point 1*, a stay point occurs at *P*3 where an individual remains stationary for a time period exceeding a threshold. In most cases, this status happens when people enter a building and lose satellite signal for a time interval until coming back outdoors. In the other situation, like *stay point 2*, a user wanders around within a certain spatial region for a period. At this moment, several GPS points (*P*5, *P*6, *P*7 and *P*8) were involved in the spatial region. Consequently, we need to calculate the mean coordinates of the region based on these GPS points. In most cases, this situation occurs when people travel outdoors and are attracted by the surrounding environment.

Using the algorithm shown in Figure 2, these stay points can be detected automatically from a user's GPS trajectory by seeking the spatial region where the user spent a period exceeding a certain threshold. For instance, in our experiment, if an individual spent more than 30 minutes within a distance of 200 meters, the region is detected as a stay point. Each stay point we extract contains information about mean coordinates, arrival time (*S.arvT*) and leaving time (*S.levT*).

```
Algorithm StayPoint_Detection(P, distThreh, timeThreh)
Input:  A GPS log P, a distance threshold distThreh
        and time span threshold timeThreh
Output: A set of stay points SP={S}
i=0, pointNum = |P|; //the number of GPS points
while i < pointNum do,
    j:=i+1; Token:=0;
    while j < pointNum do,
        dist:=Distance($p_i$, $p_j$);  //calculate the distance between points
        if dist > distThreh then
            ΔT:=$p_j$.T-$p_i$.T;   //calculate the time span between two points
            if ΔT>timeThreh then
                S.coord:=ComputMeanCoord({$p_k$ | i<=k<=j})
                S.arvT:= $p_i$.T;  S.levT=$p_j$.T ;
                SP.insert(S);
                i:=j; Token:=1;
            break;
        j:=j+1;
    if Token!=1  then i:=i+1;
return SP.
```

Figure 2. Algorithm for stay point detection

The reasons why we detect stay points in such ways lie in two aspects. On one hand, if we directly perform clustering on each individual's GPS logs, as depicted in Figure 3-A), we will miss some significant places like home and shopping malls. As GPS devices lose satellite signal indoors, few GPS points will be generated on such places (like stay point 1 shown in Figure1). 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 do not carry semantic meanings, will be extracted. Moreover, the computation of clustering will be extremely heavy as the number of GPS points is quite large compared to that of stay points. On the other hand, as demonstrated in Figure 3-B), the boundary problem of the grid-based partition method might also cause the non-detection of significant places.



A) Clustering-based detection B) Grid-based detection

Figure 3. Other possible stay-point-detection algorithms

Location history: Location history is a record of locations that an entity visited in geographical spaces over an interval of time. Given GPS trajectories and the stay points detected from the trajectories, an individual's location history can be represented as a sequence of places they visited with corresponding arrival and leaving times. However, the location history of various people is 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 distance between them. Moreover, user similarity is not a binary value, i.e., 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.

Hierarchical graph: To address this issue, we propose a framework, referred to as a hierarchical graph. As illustrated in Figure 4, we put all users' stay points into a dataset and hierarchically cluster this dataset into several spatial regions (clusters) in a divisive manner. Thus, the similar stay points from various users will be assigned to the same clusters on different layers. On each layer of the hierarchical framework, with individual stay points and trajectories, each user can build a 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 trajectories that a user created between two places (clusters). Consequently, a user's hierarchical graph (HG) can be formulated as a set of graphs HG={G} built on different geo-spatial scales. Each graph $G_i \in HG$ includes a set of vertexes and edges, G_i=(V, E), whereas V={C} is a set of clusters which contains the user's stay points.

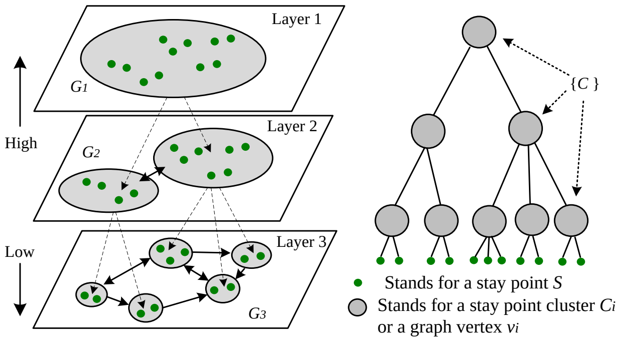

Layer 1
G_1
High
Layer 2
G_2
Low
Layer 3
G_3
{C }
Stands for a stay point S
Stands for a stay point cluster C_i or a graph vertex v_i

Figure 4. Hierarchical graph modeling user location history

Human trajectories show a high degree of temporal and spatial regularity. 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 pre-defined grids or administrative regions to build hierarchy, clustering user-generated stay points is a data-driven approach, which can feature the distribution of users' spatio-temporal data and discover the regions with semantic meanings and irregular structure.

Using this hierarchical graph, we are enabled to model each individual's location history consistently and measure user similarity on different geo-spatial scales. From the top to the bottom of the hierarchy, the spatial 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 a different extent of similarity. The users who share the same location histories on a lower layer would be more similar than those who share location histories on a higher layer.

3.2 Architecture of HGSM

Figure 5 gives an overview of the architecture of HGSM, which consists of three processes: location history presentation, user similarity exploration and friend & location recommendation. In this paper we pay more attention on user similarity exploration, which will be detailed in Section 4.

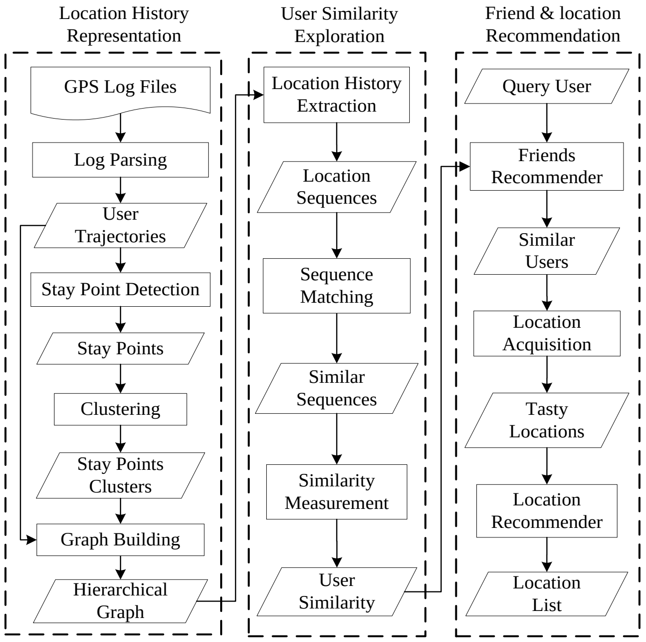


Figure 5. Architecture of HGSM

3.2.1 Location history representation

As depicted in the left most box, given GPS logs from a group of people, we first parse the spatio-temporal data and formulate trajectories for every user. Second, we extract stay points from each individual's trajectories using the algorithm described in Figure 2, and then put these stay points together into a dataset, which will be hierarchically clustered into several spatial regions in a divisive manner. The hierarchical clusters derived from all users' stay points provide an approach to consistently measure the similarity between different individual's stay points. Third, based on the shared stay point clusters and individual trajectories, a hierarchical graph is built to model each user's location history. In other words, each user holds an individual hierarchical graph based on a shared framework.

3.2.2 User similarity exploration

The middle box of Figure 5 shows the procedure of user similarity exploration which can be performed offline. First, given two users' location histories, represented by two hierarchical graphs, we search for the same graph nodes these two users shared on each layer of the framework. Later, a sequence containing such graph nodes will be respectively retrieved from each user's directed graphs on each layer. Second, knowing the information, including arrival time and leaving time of each stay point, we are capable of calculating the time interval between two nodes from the extracted sequence. Thus, we can find the similar sub-sequence from the given sequence pairs. Here a similar sequence stands for two individuals sharing the property of visiting the same sequence of places with a similar time interval. Third, based on the retrieved similar sequences, we calculate for the pair of users a similarity score considering the following factors.

- The longer the similar sequence of places shared by two users, the more similar the two users might be.
- The finer the granularity of geographic regions shared by two individuals, the more similar the individuals might be.

Consequently, we endow the location sequence 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 4 for more details.)

3.2.3 *Friend and location recommendation*

Given a user as a query, we can rank the people in a community according to the similarity of their score to the user. Then a set of people with relatively high scores can be retrieved as potential friends for the person. Further, using the location histories of these friends, the individual becomes more capable of discovering some geographic regions, such as shopping malls, restaurants and parks, etc., matching his/her taste. Later, any existing memory-based algorithms for collaborative recommendation can be employed here to measure the user's interests to these locations. Hence, in this paper, we pay more attention on measuring user similarity rather than describing the details of recommendation algorithms.

4. User Similarity Exploration

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

4.1 Location History Extraction

The hierarchical graph offers an effective representation of a user's location history, which implies sequence property of user movement behavior on geographic spaces of different scales. To better measure the similarity between two users, on each layer of their hierarchical graphs we first find the same graph nodes the users shared, and then formulate a sequence based on these graph nodes. Later, measuring the similarity between two users can be transformed into a problem of sequences matching.

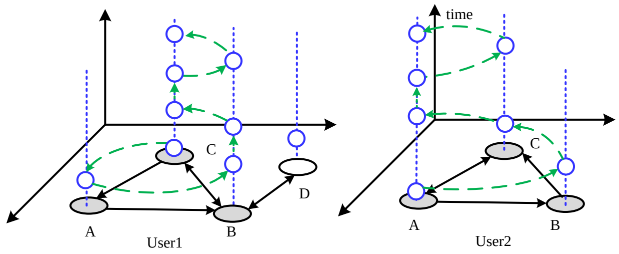


Figure 6. Sequence representation

Using a layer of two users' hierarchical graphs as a case, Figure 6 demonstrates how a sequence of places is extracted from each individual's location history. Over each graph node, a list of blue nodes linked by a dash line denotes the instances of the users' visiting at each cluster at a different time. This list can be obtained by ranking the user's stay points in each cluster according to timestamps. As we can see, user 1 and user 2 share the same graph nodes *A*, *B* and *C*. Using a green curve, we can sequentially connect the blue nodes over these graph nodes in terms of time serials. Therefore, a sequence < *C*, *A*, *B*, *B*, *C*, *C*, *B*, *C* > is generated for user 1 and a sequence <*A*, *B*, *C*, *A*, *A*, *C*, *A*> is created for user 2. For simplicity we represent these sequences as follows <*C*(1), *A*(1), *B*(2), *C*(2), *B*(1), *C*(1)> and <*A*(1), *B*(1), *C*(1), *A*(2), *C*(1), *A*(1)>; whereas, the number following a graph node represents how many times the user successively traveled in the corresponding cluster. Given each user's arrival time (*S.arvT*) and leaving time (*S.levT*) on each cluster, we are able to calculate the time interval Δt_i between two items of these sequences. Thus, the two sequences can be represented as follows:

$$\text{User 1: } C(1) \xrightarrow{\Delta t_1} A(1) \xrightarrow{\Delta t_2} B(2) \xrightarrow{\Delta t_3} C(2) \xrightarrow{\Delta t_4} B\ (1) \xrightarrow{\Delta t_5} C\ (1)$$

$$\text{User 2: } A(1) \xrightarrow{\Delta t_1{}'} B(1) \xrightarrow{\Delta t_2{}'} C(1) \xrightarrow{\Delta t_3{}'} A(2) \xrightarrow{\Delta t_4{}'} C(1) \xrightarrow{\Delta t_5{}'} A(1)$$

Consequently, we can formally define a sequence of geographic regions extracted from a user's location history as follows.

$$seq\text{=}< a_1(k_1) \xrightarrow{\Delta t_1} a_2(k_2) \xrightarrow{\Delta t_2} \ldots \xrightarrow{\Delta t_{i-1}} a_i(k_i) \xrightarrow{\Delta t_i} a_{i+1}(k_{i+1}) \xrightarrow{\Delta t_{i+1}} \ldots>,$$

where $\mathrm{a_i} \in V$ is a cluster ID, and $\mathrm{k_i}$ is the times the user successively visits cluster $\mathrm{a_i}$. Being the transition time the user traveled from cluster $\mathrm{a_i}$ to $\mathrm{a_{i+1}}$, $\Delta \mathrm{t_i}$ is calculated as equation (1).

$$\Delta t_i = a_{i+1}(0).arvT - \ a_i(k_i - 1).levT; \qquad (1)$$

4.2 Sequence Matching

4.2.1 *Definitions Related to Similar Sequences*

Similar sequences: A pair of sequences $seq_{\ 1}$ and $seq_{\ 2}$ are similar, if and only if they satisfy the following conditions:

$$seq_{\ 1} = < a_1(k_1) \xrightarrow{\Delta t_1} a_2(k_2) \xrightarrow{\Delta t_2} \ldots\ a_m(k_m)>,$$

$$seq_2 \ = < \ b_1(\mathrm{k}_1{}') \xrightarrow{\Delta \mathrm{t}_1{}'} b_2(\mathrm{k}_2{}') \xrightarrow{\Delta \mathrm{t}_2{}'} \ldots\ b_m(k_m{}') >,$$

1. $\forall\, 1 \le \mathrm{i} \le \mathrm{m}, \mathrm{a_i} = \mathrm{b_i}$, i.e., the nodes at the same position of the two sequences share the same cluster ID;
2. $\forall\, 1 \le i < m, |\Delta t_i - \Delta t_i{}'| \le t_{th}$ where t_{th} is a pre-defined time 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 $sseq$ contained in $seq_{\ 1}$ and $seq_{\ 2}$ is retrieved as below.

$$<b_1(\min(\mathrm{k}_1, \mathrm{k}_1')) \to b_2(\min(\mathrm{k}_2, \mathrm{k}_2')) \to \cdots\ b_m(\min(\mathrm{k_m}, \mathrm{k_m'}))>,$$

where $\min(\mathrm{k}_1, \mathrm{k}_1')$ denotes the minimal value between k_1 and k_1'.

m-length similar sequence: If the number of nodes in a similar sequence is *m*, we call this sequence *m-length* similar sequence. The *maximum-length* similar sequence stands for the sequence that is not contained in any other similar sequences.

Using three sequences from three users, Figure 7 illustrates the definitions presented above. At first, the three sequences shown below can be extracted from these users' location histories by employing the approach we demonstrated in Figure 6.

$$< (A(1) \xrightarrow{1.5h} B(2) \xrightarrow{2h} C(3) >, < A(2) \xrightarrow{2h} B(3) \xrightarrow{3.2h} C(2)> ,$$

$$< A(2) \xrightarrow{1h} D(1) \xrightarrow{1h} B(2) \xrightarrow{2.4h} C(2) >.$$

Clearly, User1 and User2 share the same nodes and a similar visiting order. Meanwhile, the interval between the two users' transition times from *A* to *B* is 0.5 hours (2-1.5=0.5) and from *B* to *C* is 1.2 hours (3.2-2=1.2). If the *temporal constraint* mentioned above is configured as 3 hours, a *3-length* similar sequence $< A(1) \to B(2) \to C(2) >$ is detected from their location histories. In User3's sequence, though node D does not occur in others' location histories, the transition time from node A to node B still satisfies the *temporal constraint* (2.5 hours in this case if User3 stays at cluster D for half an hour). Therefore, the second node of this sequence can be skipped, and a *3-length* similar sequence $< A(2) \rightarrow B(2) \rightarrow C(2) >$ is retrieved from User1 and User3's sequence pair. Here, these two *3-length* similar sequences are *maximum-length* similar sequences as they are not contained in others. However, a sequence like $< A(2) \rightarrow B(2) >$ is not a maximum one as it is a subset of $< A(2) \rightarrow B(2) \rightarrow C(2) >$.

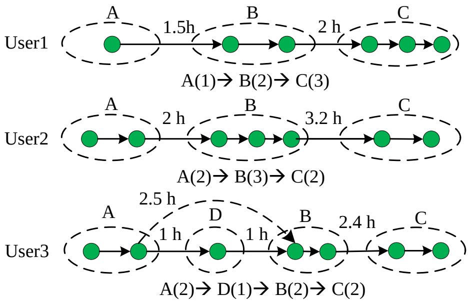


Figure 7. An example of similar sequences

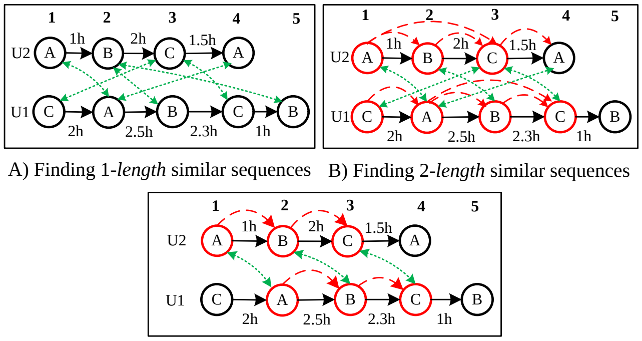
4.2.2 *Similar Sequence Matching*

Figure 8 shows the algorithm we implemented to detect similar sequences from given sequence pairs. Two operations, including sequence extension and sequence pruning, are involved in this process. In the extension operation, we aim to extend each *m-length* similar sequence to a (*m+1*)-*length* one. This operation starts from 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. However, it will be quite time-consuming to search similar sequences with very long lengths. We observe that the longer a similar sequence is the lower probability of occurrence it would be. Therefore, to improve the efficiency of sequence matching, we set a parameter *maxLength*, which is used to stop the extension operation when the length of a similar sequence increases to a certain value.

```
Algorithm Sequence_Matching(Seq1, Seq2, maxLength, $t_{th}$)
Input:    A sequence pair Seq1 and Seq2,  a length threshold of similar
          sequence maxLength, a time threshold $t_{th}$
Output: A set of maximum-length similar sequences SequenceSet
Local Variable: step
Add 1-length sequence into SequenceSet, step := 1
while step <= maxLength do
    for each step-length sequence seq in SequenceSet
        // Extend a step-length sequence to a step+1-length one
        G = ExtendSequence(seq, $t_{th}$);
        Add G into SequenceSet
    PruneSequence(SequenceSet); //Prune non-maximum sequences
    step := step +1;
return SequenceSet
```

Figure 8. Algorithm of sequence matching

Using part of the sequence pair we extracted in Section 4.1, Figure 9 demonstrates the algorithm presented above. Here, the figures on the top of each box stand for the position of each node in a sequence. First, as shown in Figure 9 A), we detect the *1-length* similar sequences as follows. <A_{12}>, <B_{23}>, <B_{25}>, <C_{31}>, <C_{34}> and <A_{42}>, where the subscript of each character represents the position of the matched node in each sequence. For instance, <A_{12}> denotes the the first node of sequence 1 sharing the same node A with the seconde node of sequence 2. The position can help us differentiate nodes of the same cluster ID being visited by users at different times. Second, Figure 9 B) depicts the process of the extension operation based on the results of the first step. If we set the *temporal constraint* t_{th} to 2 hours, four *2-length* similar sequences including <A_{12}, B_{23}>, <A_{12}, C_{34}>, <B_{23}, C_{34}> and <C_{31}, A_{42}> can be retrieved. Then, in the pruning operation, all the *1-length* sequences will be removed from the similar sequence set (*SeqenceSet*) because they are contained in the *2-length* sequences. Third, based on the *2-length* sequences, one *3-length* similar sequence <A_{12}, B_{23}, C_{34}> can be detected. Subsequently, in the pruning operation, except for <C_{31}, A_{42}>, the rest of *2-length* similar sequences will be removed from *SeqenceSet* as they are subsets of the retrieved *3-length* similar sequence.



A) Finding 1-*length* similar sequences B) Finding 2-*length* similar sequences

C) Finding 3-*length* similar sequences

Figure 9. Demonstration of sequence matching

4.3 Similarity Measurement

The retrieved similar sequences are used to compute an overall similarity score for each user-pair. When calculating the score, we take into account two factors: the length of a similar sequence and the layer in which the sequence was found. First, we calculate the score that two users get on a certain layer by adding up the score of each similar sequence found on this layer. Then, the score of each layer will be weighted and summed up to a final score.

Similarity measure of an m-length sequence: the score that an *m-length* similar sequence obtains can be formulated as equation (2):

$$s_{(m)} = \alpha_{(m)} \sum_{i=1}^{m} \min(\mathrm{k}_i, \mathrm{k}_i') \qquad (2),$$

where $\alpha_{(m)}$ is an m-dependent coefficient which will be enlarged with the increasing length m. For instance, in the experiment, we found that when $\alpha_{(m)} = 2^{m-1}$, the measure we proposed achieved a high performance.

Similarity at single layer: As shown in equation (3), the similarity between two users on a certain layer of their hierarchical graphs is measured based on all the *maximum-length* similar sequences retrieved on this layer. Here n is the number of similar sequences the two users matched on the given layer. s_i is the score of the i-th similar sequence, which can be calculated according to equation (2). N_1 and N_2 respectively denote the number of stay-points of the two users.

$$S_l = \frac{1}{N_1 * N_2} \sum_{i=1}^{n} s_i \qquad (3)$$

Dividing the similarity by the factor $N_1 * N_2$ is motivated by the problem of unbalanced data of users. Intuitively, users joining in a Web community on different periods will generate different amount of GPS logs. Thus, if we do not consider the scale of data, the individuals owning a large amount of data are more likely to be similar to others than users having less data.

Similarity across multi-layer: As shown in equation (4), the user similarity across multi-layer is computed as the weighted sum of the score of each layer:

$$S_{overal} = \sum_{l=1}^{H} \beta_l S_l \qquad (4)$$

Here *H* stands for the total layers of the hierarchical graph. β_l is a layer-dependent coefficient which represents the support of similarity of sequences on the *l-th* layer. The lower the layer a sequence was detected, the higher score it obtains. In our experiment, $\beta_l=2^{l-1}$.

5. Experiments

In this section, we first present the experimental settings which consist of the introduction about GPS devices we used, volunteers we summoned, data we collected and some parameters we selected in the experiment. Then, our approach is evaluated as an information retrieval problem. With a user-labeled ground truth, a relationship matrix among these volunteers, we are able to evaluate the search results using mean average precision (*MAP*) and normalized discounted cumulative gain (*nDCG*).

5.1 Settings

5.1.1 GPS Devices and GPS Logs

Figure 10 shows the GPS devices we chose to collect data from. They are comprised of stand-alone GPS receivers (Magellan Explorist 210/300, G-Rays 2 and QSTARZ) and GPS phones. All of them are set to receive GPS coordinates every two seconds.

Figure 10. GPS Devices Used in the Experiments

Using these devices, 65 volunteers respectively logged their life experiences with GPS traces over the past 6 months. All of them were suggested to switch on their devices as long as they traveled outdoors. Among these volunteers, we can identify the relationships, such as family member, girlfriend, boyfriend, roommates, workmates, classmate, strangers, etc. As depicted in Figure 11, the data they collected covers 28 big cities in China and some cities in the USA, South Korea, and Japan. The total distance of these GPS logs exceeds 50,000 KM.

5.1.2 Parameter Selection

Stay point detection: In our experiment, when detecting stay points from a given GPS trajectory, we set *timeThreh* to 30 minutes and *distThreh* to 200 meters. In other words, if an individual stays over 30 minutes within a distance of 200 meters, a stay point is detected. These two parameters enable us to find out each individual's significant places, such as restaurant, home, and shopping mall, etc., while ignoring the geographic regions without semantic meaning, like the places where people wait for traffic lights or meet congestion.

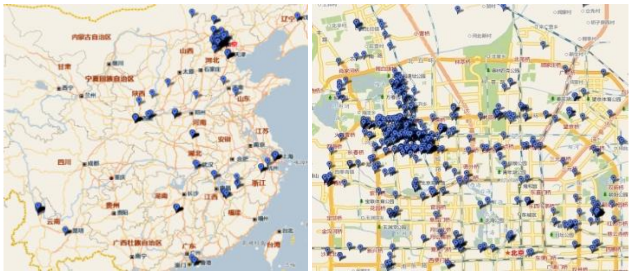


Figure 11. Distribution of the data used in the experiments

Clustering: A density-based clustering algorithm called OPTICS is employed to hierarchically cluster stay-points into geographic 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, like a lake. The clustering operation continues working 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 in geo-space is smaller than 500 meters. Taking these parameters, we establish 4-layer hierarchical clusters, which provide each individual with a consistent framework to build individual hierarchical graphs. The top layer is referred to as layer 1 (higher layer) and the bottom layer is called layer 4 (lower layer).

Sequence matching: With regard to the *temporal constraint*, we endow t_{th} with variable values on different layers considering the property of user behavior on diverse scales of geographic spaces. Intuitively, the transition time we spend between two places, like two cities, far away from each other would take us a longer time than it takes to travel between close regions like restaurants near our home. Given the relatively long transition time, a bigger time threshold needs to be selected for the sequence matching operation on a higher layer. Therefore, we set $\mathrm{t_{th}}$of layer *l* to (*H-l+1*)·*T*, where *H* is the depth of the hierarchy (here *H*=4 is based on the clustering result mentioned above). In other words, the t_{th} of layer 4 is configured as T and t_{th} of layer 1 is selected as 4*T*. After trying a set of *T*, we observe that the performance of HGSM does not change when *T* increases to a certain value.

Similarity measurement: To differentiate the significance of similar sequences with different length and on different layers, we set $\alpha_{(m)} = 2^{m-1}$, and $\beta_l = 2^{l-1}$. Here $\alpha_{(m)}$ increases exponentially with the length of sequence (*m*), since we observe that the occurrence of *m-length* similar sequences drops exponentially as the *m* increases. Thus, the significance of an occurrence of an *m-length* similar sequence increases exponentially with *m*. At the same time, the number of similar sequences found on the *l*-layer drops exponentially as the *l* increases. Therefore, the significance of similar sequences found on *l*-layer increase exponentially with *l*.

5.2 Evaluation Approach

Ground truth: After data collection, each volunteer is required to rate other users based on individual understanding and the relevance suggestion shown in Table 1. Then, a relation matrix of these volunteers is generated and is used as the ground truth to evaluate the search results of each user. The relevance rating between two users is asymmetric, i.e., though user A rates 2 on user B, user B may not rate 2 to A.

Table 1. Detailed relevance settings

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

Evaluation Framework**:** As depicted in Figure 12, our approach is evaluated as an information retrieval problem, in which 65 people are respectively used as queries to search for each of them the top ten similar users. For instance, using user U_i as a query, we retrieve the top ten similar users based on their similarity score to U_i. Then, a relevance vector *G* of the search results is formulated based on the relationship matrix. Given the retrieved *G* and 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.

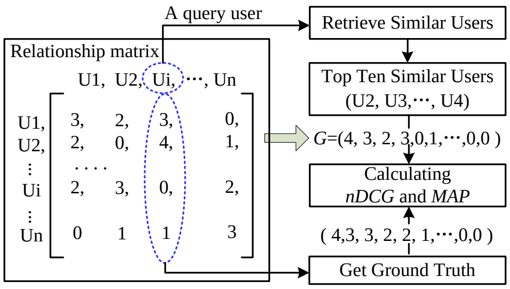


Figure 12. The framework of evaluation

Evaluation Criterions: *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 $G =< 4, 0, 2, 3, 3, 1,0,2,1,1 >$ 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 [8]. The discounted cumulative gain of *G* is computed as follows: (In our experiments, *b* = 2.)

$$DCG[i] = \begin{cases} G[1], & if\ i = 1 \\ DCG[i-1] + G[i], & if\ i < b \\ DCG[i-1] + \dfrac{G[i]}{log_b i}, & if\ i \geq b \end{cases} \qquad (5)$$

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

Baselines**:** we compare our approach with three baselines. First is similarity by counting the regions two users shared. It is an intuitive method that most people might conceive of. The rest are cosine similarity and Pearson similarity measures which have been widely used in recommendation systems, and have been claimed in paper [18] to outperform other existing similarity measures. Suppose *N* clusters $\{c_i, 1 \leq i \leq N\}$ are generated on a certain layer of the shared framework. If in the cluster c_i User1 has k_i stay-points and User2 has l_i stay-points, the location histories of User1 and User2 can be represented as follows.

$u_1 =< k_1, k_2, \ldots k_i, \ldots, k_N >$ and $u_2 =< l_1, l_2, \ldots, l_i, \ldots, l_N >$. The similarity of two users by *count* is computed as equation (6):

$$sim_{count}(u_1, u_2) = \sum_{i=0}^{N} \min(k_i, l_i) \qquad (6)$$

Cosine similarity and Pearson similarity are computed as equation (7) and equation (8) respectively:

$$sim_{cosine}(u_1, u_2) = \frac{\sum_i k_i l_i}{\sqrt{\sum_i l_i^2}\sqrt{\sum_i k_i^2}} \qquad (7)$$

$$sim_{pearson}(u_1, u_2) = \frac{\sum_i (k_i - \bar{u}_1)(l_i - \bar{u}_2)}{\sqrt{\sum_i (k_i - \bar{u}_1)^2 \sum_i (l_i - \bar{u}_2)^2}} \qquad (8)$$

5.3 Experimental Results

First we clarify some notations shown in the following figures: *Seq* stands for the similarity measure only considering sequence feature, and *Hier* denotes the measure considering the hierarchical property of geographic spaces. Thus, *Hier+Seq* represents the measure (HGSM) of similarity considering both the sequence and hierarchy properties. *Count* means *similarity-by-count* on the bottom layer, and *Hier+Count* means *similarity-by-count* across multi-layer. *Cosine* and *Pearson* respectively denotes the cosine similarity and Pearson similarity on the bottom layer. *Hier+Cosine* and *Hier+Pearson* respectively represent the cosine similarity and Pearson similarity across multi-layers.

Figure 13 shows the comparative study of *MAP* between our approach and baselines. HGSM shows clear advantages over cosine similarity, Pearson similarity and *similarity-by-count*. Moreover, by considering the similarity across multi-layer, *Hier+Seq* outperformed *Seq* which only calculates similarity based on the similar sequences on the bottom layer.

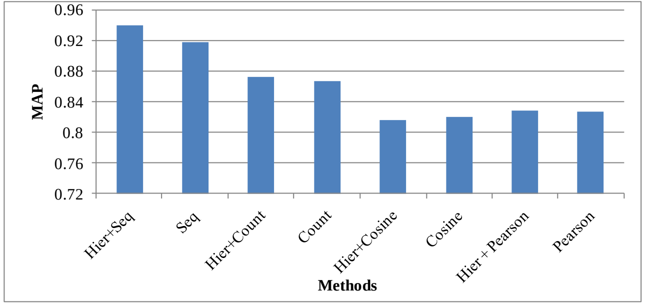


Figure 13. Comparison of MAP among different methods

Using *nDCG*, Figure 14 further differentiates our approach from baselines. Obviously, HGSM (*Hier+Seq*) leads the performance in both *nDCG@*5 and *nDCG*@10 among these methods. Moreover, the hierarchical property of geo-space better improves the performance of *Seq*.

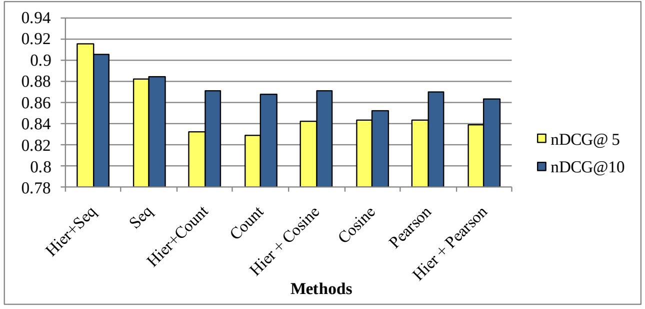


Figure 14. Comparison of *nDCG* among different methods

As we mentioned previously, sequence matching would be quite time-consuming if we attempt to find very long similar sequences. To improve the matching efficiency, a length threshold *maxLength* is defined in the sequence matching algorithm. Figure 15 shows the *MAP* and *nDCG*@5 of our approach changing over the *maxLength*. We observe that when the *maxLength* exceeds 5, the performance of the ranking does not vary any more. It suggests that instead of searching for all the similar sequences, finding sequences under a certain length can achieve comparable performance.

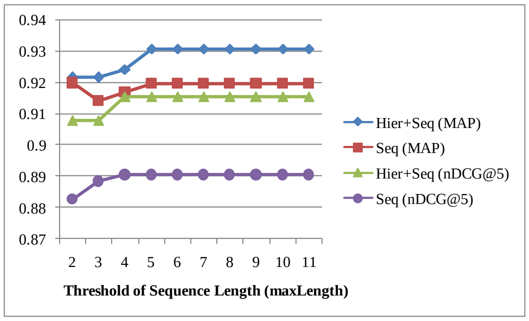


Figure 15. *MAP* and *nDCG*@5 changing over *maxLength*

Figure 16 and Figure 17 respectively present the *MAP* and *nDCG*@5 of our approach changing over the time threshold t_{th} defined in the sequence matching algorithm. We observe that at the beginning the performance of our approach is improved as the t_{th} increases. Then, when the time threshold increases to a certain value, the performances reach their summit and do not vary any more. The data shown in these curves also justified the advantage of the hierarchical property of geo-space over the single-layer method in measuring user similarity.

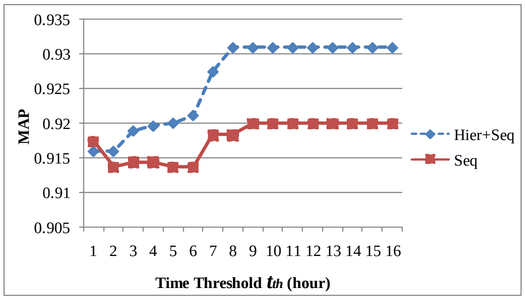


Figure 16. MAP changing over time threshold

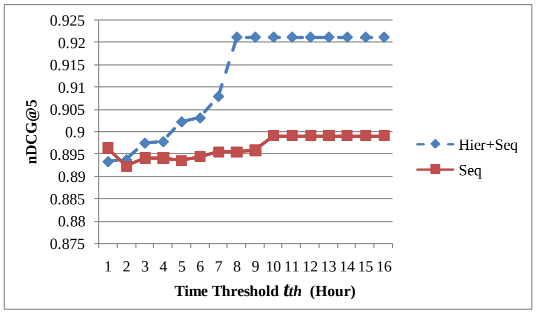


Figure 17. *nDCG*@5 changing over time threshold

Table 2 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, i.e., layer 4 is more capable of discriminating similar users than layer 3, while the approach considering the hierarchy property achieved the best performance.

Table 2. *MAP* and *nDCG* changing on different layer

	Lay-1	**Lay-2**	**Lay-3**	**Lay-4**	**multi-lay**
MAP	0.607	0.713	0.829	0.917	**0.939**
nDCG@5	0.647	0.743	0.839	0.881	**0.915**
nDCG@10	0.675	0.771	0.847	0.884	**0.905**

5.4 Discussions

Based on the experimental results, we can easily draw the conclusion that the HGSM is quite effective against existing measures in mining user similarity based on geographic data.

Sequence property: We are not surprised to observe the advantage of sequence property of HGSM in the experimental results, since a sequence of geographic regions will capture more information of users' movement behavior as compared to stand-alone locations. It is not difficult to understand this claim using the following case. Suppose user A and user B are a couple who share the location sequences of $< A \rightarrow \mathrm{B} \rightarrow \mathrm{C} >$. Meanwhile, another user C also visit these three places separately or in another order like $< B \rightarrow A \rightarrow C >$. In this case, the measure of *similarity-by-count*, cosine similarity and Pearson similarity cannot distinguish user C from user B when we attempt explore their similarity to user A.

Hierarchy property: In general, Figure 13 to Figure 17 respectively presents the contribution of the hierarchy property of HGSM over the single-layer method. Further, Table 2 illustrates how this contribution generated by investigating the performance of each layer of the hierarchy. On one hand, the layer with finer granularity is more capable of distinguishing similar users from each other as compared to the layer with coarse granularity. Imagine that a cluster would 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 other hand, however, if we only consider users' location histories on the layer of fine granularity, users' high-level movement behavior 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 recognized if we only investigate their movement behavior on the bottom layer. Overall, the layer with relatively fine granularity improves HGSM's capability of precisely discriminating similar users, while the layer with relatively coarse granularity enhances HGSM's capability of recalling similar users.

Constraint on sequence length: The data shown in Figure 15 justifies the feasibility of our approach in constraining the length of a similar sequence we attempt to find. We can clearly observe that the performance of HGSM would not increase any more when the length of a similar sequence exceeds 5. Basically, with the increasing length of sequences, the occurrence of such sequences in users' location histories decreases rapidly.

Temporal constraint: The performance of HGSM increases as the threshold of *temporal constraint* increases at the beginning, while remaining unchanged after the threshold exceeds a certain value. The reason behind this phenomenon lies in two parts. On one hand, a restrictive threshold for *temporal constraint* is not optimal to retrieve similar sequences. Therefore, when the time threshold increases, more similar sequences will be found, and more evidence is provided to support users' similarity. On the other hand, when the time threshold increases to certain value, almost all the similar sequences have already been retrieved. Hence, the performance of HGSM will not vary anymore.

6. Conclusion

People's location histories imply their interests and preferences. In this paper, we mine similarity between users based on their geographic location histories. A framework, referred to as HGSM, is proposed to enable us to consistently model each individual's location history, and effectively measure the similarity among users. It is a step towards mining knowledge from multiple users' spatio-temporal data. It can explore not only the relationship between users but also the correlation among geographic regions. Many applications, such as friend recommendation and location recommendation, etc. can be enabled by this framework.

The advantages of HGSM in measuring user similarity based on spatio-temporal data lie in two parts, the sequence property of people's movement behavior and the hierarchy property of geographic spaces. On one hand, a sequence of geographic regions captures more information of users' movement behavior as compared to stand-alone locations. Thus, we become more capable of differentiating people of different levels of similarity. On the other hand, we explore users' location histories on different scales of geographic spaces. The layer with relatively fine granularity enhances our capability of precisely discriminating similar users, while the layer with relatively coarse granularity enables us to recognize high-level user behavior and further recall unobvious similar users.

We evaluated the performance of HGSM using the GPS data collected by 65 volunteers over a period of 6 months. As a result, HGSM considering sequence property (*Seq*) clearly outperforms the *similarity-by-count*, cosine similarity and Pearson similarity in both *MAP* and *nDCG*. Further, the combination of hierarchy property and sequence property (*Hier+Seq*) offers a significant improvement on the performance of HGSM (*Seq*).

In the future, we intend to extend our work in the following three directions. First, we attempt to further improve the performance of HGSM by indentifying useful features, such as distance between geographic locations and the popularity of a location, etc, for similarity measurements. Second, we would like to improve the efficiency of the algorithms used in HGSM. Third, developing novel applications, such as personalized location recommendation based on HGSM, is also a task we aim to fulfill.

7. REFERENCES

[1] Bikely: http://www.bikely.com/

[2] GPS Track route exchange forum: http://www.gpsxchange.com/

[3] GPS sharing: http://gpssharing.com/

[4] Adomavicius, G., Tuzhhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transaction on Knowledge and Data Engineering. 17, 6 (2006), 734-749.

[5] Brunato, M., Battiti, R., Villani, A., Delai, A. A Location-Dependent Recommender System for the Web. In Proc. of the MobEA Workshop, 2002.

[6] Gonotti, F., Nanni, M., Pedreschi, D., Pinelli, F. Trajectory pattern mining. In Proc. of KDD'07, ACM Press (2007), 330-339

[7] Horozov,T., Narasimhan, N., Vasudevan, V. Using Location for Personalized POI Recommendations in Mobile Environments. In Proc. of the International Symposium on Applications on Internet 2006, 124-129

[8] Jarvelin, K., Kekalainen, J. Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems ,ACM Press(2002), 422-446

[9] Krumm, J., Horvitz, E. Predestination: Where Do You Want to Go Today?, *IEEE Computer Magazine*, vol. 40, no. 4, April 2007, pp. 105-107

[10] Krumm, J., Horvitz, Eric. Predestination: Inferring Destinations from Partial Trajectories. In Proc. of UBICOMP'06, Springer-Verlag Press(2006), 243-260

[11] Krumm, J., Horvitz, Eric. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In *Proc.* of *Mobiquitous 2004*, IEEE Press (2004), 4-13.

[12] Liao, Ll, Patterson, D.J., Fox, D., Kautz, H. Building Personal Maps from GPS Data. In proc. of *IJCAI MOO05*, Springer Press(2005), 249-265

[13] Liao, L., Fox, D., Kautz, H. Learning and Inferring Transportation Routines. In Proc. of the National Conference on Artificial Intelligence. ACM Press (2004), 348-353.

[14] Linden, G., Smith, B., York, J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 7, 1(2003), 76-80

[15] Patterson, D.J., Liao, L., Fox, D., Kautz, H. Inferring High-Level Behavior from Low-Level Sensors. In Proc. of UBICOMP'03, Springer Press (2003), 73-89

[16] Sohn, T., Varshavsky, A., LaMarca, A., Chen, Y.M. Mobility Detection Using Everyday GSM Traces. . In Proc. of UBICOMP'06, Springer-Verlag Press(2006), 212–224

[17] Spertus, E., Sahami, M., Buyukkokten, Orkut., Evaluating similarity measures: a large-scale study in the orkut social network, In Proc. of KDD'05, ACM Press (2005), 678 – 684.

[18] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Application of Dimensionality Reduction Recommender System – A Case Study, In ACM WebKDD Workshop, 2000.

[19] Takeuchi, Y. Sugimoto, M. CityVoyager: An Outdoor Recommendation System Based on User Location History. In Proc. UIC'2006, Springer Berlin (2006), 625-636

[20] Yang W.S., Cheng, H.C, Dia, J.B. A location-aware recommender system for mobile shopping environments. An international Journal: Expert Systems with Applications. Pergamon Press (2008), 437-445.

[21] Zheng, Y., Liu, L., Wang, L.H., Xie, X. Learning transportation mode from raw GPS data for geographic applications on the Web. In Proc. *WWW 2008*, ACM Press (2008), 247-256